Jean-Claude Moisdon
Michel Nakhla

# Recherche opérationnelle

## Méthodes d'optimisation en gestion

Presses des Mines

# Recherche opérationnelle
## Méthodes d'optimisation en gestion

Michel Nakhla, Jean-Claude Moisdon

**Collection Les cours**

Dans la même collection

*Systèmes énergétiques vol 1, vol 2, vol 3*
Renaud Gicquel

*Stratégie d'entreprise*
Thierry Weil

*Mécanique des fluides*
François Cauneau

*Aide-mémoire de géostatistique linéaire*
Pierre Chauvet

*Introduction aux transferts thermiques*
Dominique Marchio, Paul Reboux

*Introduction au génie atomique,*
Jacques Bouchard, Jean-Paul Deffain, Alain Gouchet

*Matériaux pour l'ingénieur*
Anne-Françoise Gourgues-Lorenzon, Jean-Marc Haudin,
Jacques Besson, Noëlle Billon, Sabine Cantournet, Yvan Chastel,
Bernard Monasse, Loeiz Nazé

*Abrégé de thermodynamique,*
Daniel Fargue

*Introduction au traitement de l'énergie électrique*
Georges Pierron

*Introduction à la physique quantique*
Bernard Degrange

*Cours d'automatique*
Brigitte d'Andréa-Novel, Michel Cohen de Lara

*Les imperfections des marchés*
Daniel Fixari

*Introduction à la métallurgie générale*
Jacques Lévy

*Comment maîtriser sa productivité industrielle,*
Hugues Molet

*Géostatistique linéaire – applications,*
Margaret Armstrong, Jacques Carignan

# Recherche opérationnelle
## Méthodes d'optimisation en gestion

Michel Nakhla, Jean-Claude Moisdon

60 boulevard Saint-Michel, 75272 Paris cedex 06, France

e-mail : presses@mines-paristech.fr

http://www.mines-paristech.fr/Presses

ISBN : 978-2-911256-15-8

Dépôt légal : 2010

Crédit photos : M. Nakhla

## INTRODUCTION

La Recherche Opérationnelle constitue selon les cas, une simple branche des mathématiques ; à l'autre bout du spectre, elle consiste en l'application des méthodes scientifiques au contrôle et au pilotage de l'action organisée. Dans le cadre de cet ouvrage, nous nous limiterons volontairement à la définition suivante :

> « La Recherche Opérationnelle est une collection de techniques, issues du champ des mathématiques appliquées, destinées à représenter des situations où un ou plusieurs acteurs ont un certain nombre de choix à effectuer, et à guider ces acteurs dans leur décision de façon à ce qu'ils satisfassent au mieux un ou plusieurs critères tout en respectant un ensemble de contraintes prédéfinies ».

En quelque sorte, on s'intéresse bien à un secteur particulier des mathématiques, celui de l'optimisation sous contraintes, mais en en restreignant le champ : en effet la définition proposée suggère que l'on ne traitera que de problèmes de décision économique, ou en tout cas qui y ressemblent : l'ensemble des variables de décision, des critères, des contraintes auront une signification économique et gestionnaire. Il s'agira donc de formaliser des problèmes d'allocation de ressources, d'organisation de la logistique, d'ordonnancement de production, de planification de production…

Cette définition explique tout d'abord la nature des problèmes abordés par la R.O (Recherche Opérationnelle) : ils ne sont pas guidés par le développement autonome d'une branche des mathématiques, mais par l'ensemble des problèmes de décision économique que peuvent se poser les collectivités organisées, pour l'essentiel les entreprises ou les administrations. Ces problèmes sont ordonnés dans des classes larges (problèmes d'ordonnancement d'atelier par exemple) où non seulement les questions posées sont communes à tous les problèmes de la classe, mais également un certain nombre de variables et de relations entre ces dernières, si bien que l'on peut élaborer des modèles de base, quitte ensuite, au cas par cas, à examiner comment on peut les adapter aux situations concrètes envisagées.

Il va de soi que le présent ouvrage, compte tenu de son volume, n'a pour ambition que de présenter les principaux modèles de référence.

Une autre conséquence de cette définition est que, comme le lecteur s'en apercevra facilement, une hétérogénéité certaine caractérise fatalement un exposé des principales techniques de la R.O : cela provient de la grande variété des problèmes décisionnels que l'on peut rencontrer dans les entreprises, problèmes qui n'ont aucune raison de relever d'un formalisme voisin. Il y a en effet peu de rapport entre l'optimisation d'un plan de transport de marchandises et la fixation d'un nombre de guichets dans un phénomène d'attente : en effet les modélisations s'attaquant à l'un et à l'autre de ces deux problèmes n'ont clairement pas grand chose de commun.

La R.O s'attaque à des problèmes de gestion et de décision des organisations économiques et à la prise en compte du combinatoire et de l'incertitude.

Prenons deux exemples pour illustrer ce propos.

Dans le problème du voyageur de commerce, ce dernier doit passer dans *n* villes, sachant qu'il souhaite visiter chacune d'entre elles une fois et une seule. Il connaît les temps de transport entre chaque ville et les autres, temps que l'on suppose fixes. Dans quel ordre doit-il effectuer ses visites de façon à ce que son temps de transport total soit minimal ? La réponse est a priori facile. La ville de départ importe peu, on s'en rend compte rapidement. Il se peut de toute façon qu'elle soit fixée. A partir de ce point de départ, notre voyageur de commerce a *n*-1 possibilités pour la ville suivante, puis $n-2$ pour la deuxième, puis $n-3$ pour troisième etc. Au total, il y a $(n-1)\times(n-2)\times ...\times 3\times 2\times 1$ solutions à ce problème, soit $(n-1)$ ! On peut alors avoir comme idée d'explorer systématiquement ces solutions, en calculant à chaque fois la somme des temps de parcours, donc le temps de parcours total, et retenir, parmi ces $(n-1)$ ! temps totaux le plus petit. Or le nombre $(n-1)$ ! devient très vite important lorsque n augmente. Pour six villes, le calcul reste possible, puisque l'on a 120 permutations à explorer et à chiffrer. Pour $n=20$, ce nombre est à peu près égal à $1{,}2164\times10^{17}$. Si par exemple on écrit un programme informatique permettant d'évaluer toutes les solutions (et un programme est facile à écrire) et si l'ordinateur dont on dispose nécessite un nanoseconde pour chaque solution on explorera l'ensemble des possibilités en 3,8 ans environ ! C'est dire que les performances des ordinateurs actuels ne libèrent pas de la nécessité de trouver des méthodes permettant de nous orienter dans la prolifération des solutions possibles liées à la présence de nombreux problèmes de gestion combinatoire. Cet exemple est intéressant car le problème posé est tout à fait réaliste (dans les cas concrets il se complique souvent considérablement ce qui ajoute encore à la complexité de sa résolution) et renvoie à des questions que se posent de nombreuses entreprises (tournées de postiers, de camions de coopératives laitières etc.). Des méthodes existent en effet pour traiter ce problème de façon efficace (même si elles ne sont pas entièrement satisfaisantes comme on le verra dans cet ouvrage à propos de l'exposé de l'une d'entre elles).

Le deuxième exemple est issu lui aussi d'un problème tout à fait pratique : un marchand de journaux se pose la question de la quantité optimale d'exemplaires d'un quotidien qu'il doit acheter à un distributeur. La demande est variable ; il ne sait pas à l'avance combien de clients lui achèteront le journal chaque jour ; il connaît évidemment le prix d'achat au distributeur et le prix de vente, si bien que pour un niveau de vente donné il peut calculer sans difficultés le bénéfice (ou la perte s'il achète trop de journaux) obtenu. Le problème est qu'il ne sait pas intégrer dans un calcul unique les différents bénéfices correspondant aux événements non prévisibles constitués par les niveaux de vente. Le calcul vient alors à son secours par deux modalités consécutives : il est tout d'abord invité à analyser sur un certain nombre de jours la chronique des ventes ; si cette analyse est suffisamment poussée, il peut éventuellement raccorder par un test statistique cette chronique de ventes à une distribution de probabilités. Ce faisant il a déjà « domestiqué » considérablement l'incertitude : certes il ne sait pas à l'avance quelle quantité de journaux il vendra tel ou tel jour, mais il a une information précieuse, à savoir la probabilité de vendre une quantité donnée (et toutes les probabilités associées aux diverses quantités possibles). La deuxième étape est alors constituée par une démonstration, qui aboutit à un résultat qui est loin d'être intuitif : si le phénomène est

répétitif (prix fixes et demande constante en probabilités), il peut alors agréger les bénéfices correspondant aux différentes ventes grâce à un opérateur simple que connaissent bien les spécialistes des probabilités, à savoir l'espérance mathématique (moyenne des bénéfices pondérés par leurs probabilités). Un théorème célèbre, le Théorème Central Limite, justifie en effet que l'on prenne ce critère comme critère de choix, dans un contexte de répétition (stabilité des données et décision répétée un grand nombre de fois). Il ne reste plus alors à notre marchand de journaux, convaincu par cet édifice théorique, qu'à calculer les espérances mathématiques de bénéfice, pour chaque niveau d'achats du quotidien, et à choisir le niveau qui maximise cette espérance. Il est intéressant de noter d'ailleurs qu'à ce stade le problème reste assez compliqué : en effet, il est facile de s'apercevoir que cette modélisation aboutit à une formule relativement complexe liant l'espérance du bénéfice au niveau d'achat, pour laquelle les techniques d'optimisation fonctionnelle classiques (dérivation par exemple) sont inopérantes ; le marchand de journaux sera donc obligé de procéder pas à pas, en tâtonnant sur le niveau d'achat, et en calculant à chaque fois la formule, pour trouver le niveau optimal. Et pourtant, les hypothèses sont frustes ! On sent bien qu'un problème réel ressemblant à ce problème « épuré » va supposer des hypothèses plus fines (variation de la loi de probabilités de la demande, possibilités de ventes le lendemain de la parution etc.) risquant de compliquer quelque peu le modèle.

Quoiqu'il en soit, on voit bien quel est l'apport de la R.O dans ce cadre : elle permet ce que l'esprit ordinaire ne peut pas faire, c'est-à-dire l'intégration dans un calcul des conséquences des multiples événements qui peuvent peser sur une décision économique.

***Quelques éléments historiques***

On a l'habitude de situer la naissance de la R.O lors de la seconde guerre mondiale, au sein de l'armée britannique, qui a commencé à formaliser un certain nombre de problèmes stratégiques, tel l'optimisation du parcours ou de la composition des convois maritimes ou encore celle du positionnement des radars etc.

Certains auteurs, cela dit, ne sont pas sans souligner que les perspectives d'application des modèles à l'action organisée se sont manifestées bien avant, qu'il s'agisse de la construction des pyramides, de la mise en place du siège de Syracuse, ou encore des problèmes de déblai et de remblai formalisés par Monge.

Pour ce qui nous intéresse, c'est à dire la diffusion de la discipline dans les entreprises, on peut néanmoins considérer que cette diffusion s'est opérée, et à un rythme accéléré, dans les années cinquante.

On ne peut s'empêcher de relier ce développement à celui de l'informatique. La plupart des outils de la R.O nécessitent en effet un support informatique pour résoudre les problèmes auxquels ils s'attaquent.

Mais, bien qu'il existe peu d'études sur ce type de sujet, on peut aussi penser que la R.O est fondée sur une représentation implicite de l'entreprise et de son fonctionnement qui s'est constituée récemment, en tout cas au vingtième siècle. Faire de l'organisation une combinatoire d'activités plus ou moins incertaines signifie en effet un préalable qui est la codification de ces activités, une explicitation de ce qui les relie, une quantification des

consommations de ressources qu'elles impliquent. C'est dire que la R.O est inséparable de l'approche scientifique du travail de Taylor.

Au cours des années cinquante le développement de la R.O dans les organisations s'est transformé en véritable engouement; une sorte de rêve semble avoir été partagé par nombre de cadres de l'industrie, à savoir la possibilité de traiter tous les problèmes de gestion, d'organisation et de décision économique par des modèles et de conduire la destinée des entreprises grâce aux mathématiques.

La R.O s'est constituée en discipline à part entière, au début des années soixante, avec la mise sur pied des principaux attributs d'une profession : experts spécialisés, création de revues, d'associations savantes, de chaires dans les universités, de congrès etc. Du côté des entreprises on note le développement de bureaux d'études, de services spécialisés dans les grandes entreprises, souvent rattachés directement à la direction générale. Le secteur public est « touché » un peu plus tard, au milieu des années soixante. Les progrès au niveau des méthodes, de leur rapidité de calcul, sont spectaculaires. Les spécialistes s'attaquent à des problèmes de plus en plus difficiles. Le ton est à l'enthousiasme; beaucoup pensent que l'on dispose là d'un dispositif essentiel de rationalisation de l'action humaine dans les systèmes organisés.

À partir des années soixante-dix, on s'interroge sur la modestie des applications concrètes dans les entreprises, ou encore du sort réservé aux modèles, parfois complètement contradictoire avec les attendus des promoteurs. Bon nombre des services de R.O des grandes entreprises sont fusionnés avec les services « d'études économiques » ou de « stratégie », comme si un discrédit pesait sur la dénomination de la discipline elle-même. Dans la foulée, le sigle des enseignements se transforme lui aussi : on trouve les modèles de R.O dans les cours dits d'« aide à la décision », ou de « méthodes scientifique de gestion », ou encore de « modélisation des faits économiques ». En 1978 paraît dans la prestigieuse revue américaine « The journal of the O.R Society » une épitaphe joliment intitulée « The future of operational research is past » , d'autant plus perturbante qu'elle est l'œuvre d'un des pionniers de la discipline, Russel L. Ackoff. Cette contribution en entraîne beaucoup d'autres et ouvre un débat parfois passionné autour de ce que l'on a pu appeler ``la crise'' de la Recherche Opérationnelle.

***Un point de vue raisonné***

Il convient tout d'abord de souligner que les applications de la R.O. existent bel et bien : c'est ainsi que la programmation linéaire est devenue une sorte de routine pour des industries du continu, notamment le raffinage pétrolier (Partie 1 de cet ouvrage), qu'il n'est guère envisageable de démarrer un projet de quelque envergure sans avoir recours à une méthode d'ordonnancement des tâches, PERT ou Potentiel (Partie 2), que les analyses du risque se sont multipliées dans de multiples secteurs (Partie 3). On peut même parfois s'inquiéter lorsque l'on constate que certains logiciels de gestion, livrés « clef en main » et tendant à automatiser la conduite des entreprises, appartenant par exemple à la catégorie des GPAO (Gestion de Production Assistée par Ordinateur), intègrent parfois, sans que l'utilisateur en ait conscience, des modèles (gestion de stocks par exemple) sur la généralité desquels il est légitime d'avoir quelques doutes.

***Objectifs et organisation générale de l'ouvrage***

Les développements qui suivent couvrent un enseignement que les auteurs donnent à l'Ecole des Mines de Paris. Il s'agit essentiellement d'une initiation et que nous ne prétendons nullement être exhaustifs sur les modèles conçus dans le cadre de la R.O.

Dans ce cadre, nous n'évoquerons pas les extensions de la programmation linéaire vers les programmes en nombres entiers ( c'est à dire où l'on contraint les solutions du problème à prendre des valeurs entières, ce qui correspond à des exigences fréquentes dans la réalité - productions non divisibles par exemple) ; de même, nous serons rapide sur les différents types d'heuristiques qui sont actuellement perfectionnées et qui sont destinées à s'attaquer à des problèmes hautement combinatoires (algorithmes génétiques, méthode Tabou, recuit simulé etc.).

Au niveau de l'aléatoire, nous ne dirons rien de deux chapitres aujourd'hui très « vivants », qui étendent les calculs probabilistes à deux situations d'incertitude qui à priori ne relèvent pas de l'aléatoire (au sens de la prise en compte de données non connues à l'avance mais « domestiquées » parce que probabilisables, cf. l'exemple ci-dessus du marchand de journaux) : il s'agit en premier lieu des situations concurrentielles (le décideur ne sait pas ce qui va arriver dans la mesure où cela dépend du comportement d'autres, les autres en question étant dans la même expectative vis à vis du décideur) ; ces situations relèvent de la théorie des jeux, discipline d'ailleurs à part entière qui a vécu ces dernières années un renouveau considérable, notamment dans le cadre des travaux des économistes, s'intéressant aux déséquilibres de marché, aux asymétries d'information entre acteurs et aux externalités. En second lieu nous pensons aux situations à proprement parler incertaines, où aucune règle ni aucune expérience ne permet d'affecter des distributions de probabilité aux données. On a affaire alors à l'édifice considérable qui porte le nom de Théorie de la décision, que l'on peut considérer comme l'aboutissement de la rationalisation des choix individuels. On pourrait penser en effet que la modélisation est impuissante dans ce cas à aider le décideur ; il n'en est rien et il est tout à fait stimulant d'examiner comment l'ingéniosité des mathématiciens appliqués les a amenés à proposer malgré tout des formalisations dans ce contexte difficile.

En dehors du fait que dans une perspective gestionnaire ces deux types de modélisation constituent davantage des références de raisonnement que des outils opérationnels, ils nécessitent un ouvrage à eux seuls, et nous avons préféré les laisser de côté ici (bien que nous les enseignions).

Si la liste des outils que ce livre décrit est donc volontairement limitée, nous avons en revanche mis un certain soin à justifier chacun d'entre eux, et à démontrer par exemple la convergence des algorithmes ou la validité des formules. Une autre optique, en effet, choisie parfois par les ouvrages de R.O., consiste à parcourir une liste étendue de modèles, sans fournir de démonstration. Ce choix n'est pas sans entraîner une certaine frustration chez les lecteurs rigoureux ; il ne fournit pas non plus de vue sur la fréquente originalité du raisonnement des chercheurs opérationnels (qui, comme on l'a dit, ont résolu à leur façon un certain nombre de problèmes sur lesquels avaient échoué de grands mathématiciens). Enfin, aucun modèle existant dans la littérature ne s'applique tel quel à des situations décisionnelles concrètes ; ils doivent être adaptés. Mais pour savoir comment les modifier, il faut savoir comment et pourquoi ils « marchent ».

Une première partie sera consacrée à la programmation linéaire. D'un point de vue pédagogique, ce choix n'est pas optimal ; en effet les programmes linéaires, parmi les outils que nous exposons ici, sont ceux qui font le plus appel aux mathématiques, alors que les graphes, par exemple, sont très économes en concepts et connaissances mathématiques, et sont donc plus à même de séduire des lecteurs intéressés par la discipline mais moins par les équations. D'un autre côté, beaucoup de problèmes, notamment résolus par des méthodes appuyées sur les graphes, pourraient l'être aussi par un programme linéaire, l'inverse n'étant pas vrai. C'est cet aspect fédérateur qui jusitifie que l'on débute ainsi l'exposé.

Après avoir relié la programmation linéaire à une problématique générale de planification de production industrielle, on développera de façon complète la méthode de résolution « phare », à savoir l'algorithme du simplexe. On soulignera qu'il ne s'agit pas de la méthode de résolution *a priori* la plus rapide et on donnera quelques indications sur d'autres méthodes logiquement plus performantes (sans que cela se vérifie *empiriquement*).

On introduira dans cette partie la notion de dualité dans les programmes linéaires, notion fondamentale d'abord d'un point de vue économique (fixation des prix d'acquisition de ressources supplémentaires), ensuite d'un point de vue pratique (analyse de sensibilité).

Dans une seconde partie, nous passerons à la théorie des graphes, et nous montrerons que cet objet, très simple dans son principe (des points et des flèches) permet de résoudre un grand nombre de problèmes combinatoires concrets et ayant des implications évidentes au niveau de l'organisation de la production et de la logistique. C'est ainsi que l'on traitera de quelques algorithmes destinés à résoudre le problème du chemin de valeur minimale dans un graphe. Nous montrerons que ces méthodes peuvent s'appliquer sans difficultés à la question très importante de l'ordonnancement des tâches dans un projet. Comme on l'a signalé plus haut, peu de projets industriels se privent à l'heure actuelle d'une planification à base d'un PERT ou d'une méthode potentiels, qui constituent les deux outils concurrents en la matière. Là aussi, l'exposé restera succinct, dans la mesure où l'on ne fera qu'évoquer les complications introduites dans ces techniques par la prise en compte de contraintes autres que celles portant sur la succession des tâches (les contraintes disjonctives notamment, qui ouvrent l'immense et difficile chapitre de l'ordonnancement d'atelier).

On traitera ensuite des problèmes liés à des graphes particuliers, nommés « arbres » ; on montrera que cette notion permet de traiter simplement des questions de tracé de réseaux (comment relier $n$ points de la façon la plus économique ?). Le concept voisin d'arborescence conduit moins, de son côté, à des outils d'optimisation de réseau qu'à une classe particulière de méthodes visant à approcher des questions hautement combinatoires et particulièrement difficiles. C'est dans ce cadre que nous aborderons le fameux problème du voyageur de commerce (cf. supra).

Le dernier chapitre de cette partie abordera les problèmes de flots : il s'agit cette fois-ci de faire circuler des flux dans des réseaux (flux divers : pièces, matières premières, mais aussi individus ou voitures sur un réseau autoroutier) de façon soit à maximiser les entrées totales sur le réseau (problème du flot maximal), soit à minimiser un coût global de transport, à entrée totale donnée (programme de transport). On montrera également

que certains problèmes *a priori* éloignés de ces préoccupations (problème de l'affectation de tâches à des effecteurs potentiels notamment) peuvent être traités avec élégance par les formalisations issues de la notion de flot.

Enfin, dans une dernière partie, nous traiterons des phénomènes aléatoires. Pour ce faire, on fera appel au calcul des probabilités, en restreignant au minimum l'arsenal mathématique correspondant (on sait qu'il peut être considérable). Il suffira donc au lecteur d'avoir des connaissances limitées aux lois probabilistes les plus courantes, aux concepts les plus répandus (espérance et variance) et aux opérations classiques sur les variables aléatoires pour consulter les trois chapitres de cette partie, consacrés d'une part à la gestion des files d'attente, d'autre part à la fiabilité des équipements, enfin à la gestion des stocks.

# Chapitre 1 • Généralités sur la Programmation Linéaire

Nous commencerons le cours de Recherche Opérationnelle par l'examen d'un outil auquel il est très souvent fait référence dans nombre de problèmes économiques, qu'ils concernent l'entreprise ou l'Etat : la programmation linéaire. Nous n'analyserons pas celle-ci sous l'angle de la signification économique des différents concepts qu'elle met en œuvre, ce qui peut être fait par exemple dans un cours de calcul économique; nous nous préoccuperons plutôt des techniques de résolution usuelles des programmes linéaires. Parfois, nous serons amenés à introduire des notions économiques comme le « profit », le « coût », les « ressources », mais ce sera davantage par simple souci d'illustration que pour indiquer le champ d'application des techniques exposées.

Soulignons enfin que dans les chapitres suivants nous ferons fréquemment allusion à la programmation linéaire, en particulier pour les problèmes de graphes, ce qui justifie que ce premier fascicule soit consacré à cet outil.

## 1.1. UN EXEMPLE SIMPLE

Avant d'aborder l'exposé le plus général sur la programmation linéaire, il convient de donner un petit exemple n'ayant aucune prétention à représenter un problème réel, mais permettant d'illustrer la plupart des résultats que nous dégagerons par la suite.

Soit une entreprise qui fabrique deux types de camions (types A et B). Cette entreprise est divisée en trois ateliers, l'atelier I fabriquant les moteurs, l'atelier II fabriquant les carrosseries, l'atelier III étant chargé de l'assemblage.

Les temps unitaires pour chacune des trois opérations et pour chaque type de camions sont consignés dans le tableau suivant :

| Ateliers | Camions de type A | Camions de type B |
|---|---|---|
| I Moteurs | 1h | 3h |
| II Carrosseries | 2h | 1h |
| III Assemblage | 1h | 1h |

Par ailleurs, l'étude des capacités de production des 3 ateliers a dégagé qu'en un mois, 450 heures de travail pouvaient être utilisées dans l'atelier I, 350 heures dans l'atelier II, 200 dans l'atelier III.

Enfin, on sait que le bénéfice unitaire réalisé par l'entreprise sur les camions de type A s'élève à 4000 et que celui réalisé sur les camions de type B est de 8000.

La question que l'on se pose est la suivante : quelle doit être la production mensuelle en camions de chaque type pour rendre le bénéfice de l'entreprise le plus grand possible ?

Nous pouvons formaliser ce problème de la façon suivante :

Soit $x_1$ la production mensuelle en camions de type A et $x_2$ celle en camions de type B.

Les contraintes de disponibilité d'heures de travail dans chacun des ateliers peuvent s'écrire:

Atelier I : $x_1 + 3x_2 \leq 450$

Atelier II : $2x_1 + x_2 \leq 350$ (a)

Atelier III : $x_1 + x_2 \leq 200$

Par ailleurs, on a évidemment :

$x_1$ et $x_2 \geq 0$ (b)

Enfin, il s'agit de trouver $x_1$ et $x_2$ répondant aux contraintes (a) et (b) et rendant la fonction.

$f = 4000\,x_1 + 8000\,x_2$ maximale.

Ce problème constitue un programme linéaire.

D'une façon générale, un programme linéaire consiste à maximiser (ou minimiser) une fonction linéaire de *n* variables, ces variables étant assujetties à respecter un ensemble de *m* contraintes également linéaires.

Dans le cas simple qui nous préoccupe, ($n$ = 2, $m$ = 3), nous allons résoudre le problème en utilisant une représentation géométrique.

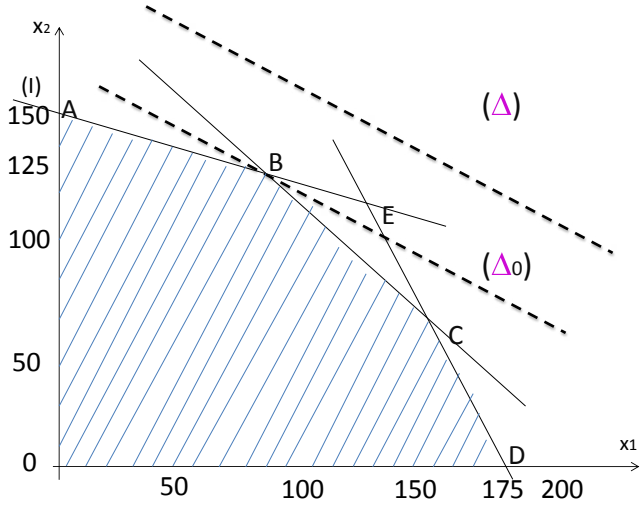



*Figure 1*

Un point M quelconque du plan représente une production qui n'est peut être pas réalisable. Pour qu'elle le soit, il faut d'abord que $x_1 \geq 0$ et $x_2 \geq 0$ donc que l'on se limite au quadrant nord-est du plan. Ensuite, il s'agit de représenter les autres contraintes du système d'inéquation (a).

Pour cela traçons les trois droites

$x_1 + 3x_2 = 450$ (I)

$2x_1 + x_2 = 350$ (II)

$x_1 + x_2 = 200$ (III)

Il est clair que ces trois droites délimitent un domaine D, hachuré sur la figure 1, qui représente l'ensemble des solutions possibles compte tenu des contraintes (a) et (b). Il s'agit du polygone convexe OABCD dont chaque sommet est donné par l'intersection de deux droites parmi les cinq qui déterminent les contraintes (A par exemple, est l'intersection de $x_1 = 0$ et de $x_1+3x_2 = 450$).

Il s'agit donc de trouver le point (ou les points) de D tel que la fonction $4000\, x_1 + 8000x_2$ soit maximale en ce point, ou, ce qui est la même chose, tel que la fonction $x_1 + 2x_2$ soit maximale.

Pour ce faire, il suffit de se souvenir que la distance de 0 à la droite $ax_1 + bx_2 = c$ est donnée par la quantité :

$$\frac{c}{\sqrt{a^2+b^2}} = \frac{ax_1^0 + bx_2^0}{\sqrt{a^2+b^2}}$$

si le point $(x_1^0, x_2^0)$ est un point quelconque de la droite.

En conséquence, le problème revient à trouver un point M de D tel que la distance de 0 à la droite passant par M et parallèle aux droites $x_1 + 2x_2 =$ constante soit la plus grande possible. Sur la figure, on a tracé une droite (Δ) de la forme $x_1 + 2x_2 =$ constante.

Pour avoir le point (ou les points) cherché, il suffit de déplacer la droite (Δ) parallèlement à elle-même en la rapprochant de 0, jusqu'à ce qu'elle « touche » le domaine D.

Il est aisé de voir que la position de (Δ) est alors ($\Delta^0$) et que le point cherché est le point B de coordonnées $(75 ; 125)$.

La résolution de ce programme linéaire a été facilitée par le fait que le nombre de variables (2) permettait une représentation géométrique. Ce n'est pas le cas en général et nous allons avoir recours à présent à des méthodes plus élaborées.

Néanmoins, de ce petit exemple, il convient de retenir les résultats suivants, qui ne sont que la transcription des résultats que nous allons trouver dans le cas général :

1) le domaine des solutions réalisables est convexe, pour $n = 2$, c'est un polygone.
2) le point où la fonction économique est maximale est un sommet du domaine convexe des solutions réalisables.
3) parmi les contraintes du programme, il en est qui sont à l'optimum « strictement » respectées (ici $2x_1 + x_2 \leq 350$) ; ce sont les contraintes « non saturées ». D'autres sont « juste » respectées (le point B est tel que $x_1 + 3x_2 = 450$ et $x_1 + x_2 = 200$) ; ce sont les contraintes saturées.

## 1.2. ETUDE DU CAS GENERAL

### 1.2.1. Forme générale d'un programme linéaire

Comme il a été dit ci-dessus, résoudre un programme linéaire consiste à maximiser (ou minimiser) une fonctionnelle linéaire d'un nombre fini quelconque de variables positives ou nulles, ces variables devant respecter un nombre fini quelconque de contraintes linéaires. Un programme linéaire peut toujours s'écrire de la façon suivante :

Soit *n* variables $x_1, x_2 \ldots x_n$ obéissant aux contraintes :

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ \ldots\ldots + a_{1n}\ x_n \leq\ b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ \ldots\ldots + a_{2n}\ x_n \leq\ b_2 \\
&\qquad\vdots \\
&a_{m1}x_1 + a_{m2}x_2 + \ \ldots\ldots + a_{mn}x_n \leq\ b_m \\
&\quad x_1 \geq 0 \\
&\quad x_2 \geq 0 \\
&\vdots
\end{aligned}
$$

$$x_n \geq 0$$

Il s'agit de trouver $x_1, x_2 \ldots x_n$ satisfaisant à ces contraintes et maximisant la fonctionnelle, appelée fonction économique :

$$z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

Les $a_{ij}$, $b_i$ et $c_j$ appartenant à R

On peut en effet toujours écrire un programme linéaire sous cette forme, après éventuellement les corrections suivantes :

1) s'il s'agit de minimiser :
$z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$

On maximisera :

$$z' = -c_1 x_1 - c_2 x_2 \ldots - c_n x_n$$

2) Si on a des contraintes du type :
$a_{i1} x_1 + a_{i2} x_2 + \ \ldots\ . + a_{in}\ x_n \geq b_i$

on les transformera en :

$$-a_{i1} x_1 - a_{i2} x_2 \ldots - a_{in} x_n \leq -b_i$$

3) Si on a des contraintes du type :
$a_{i1} x_1 + a_{i2} x_2 + \ \ldots\ . + a_{in}\ x_n = b_i$

elles équivalent à :
$a_{i1} x_1 + a_{i2} x_2 + \ \ldots\ + a_{in}\ x_n \leq b_i$
$a_{i1} x_1 + a_{i2} x_2 + \ \ldots\ . + a_{in}\ x_n \geq b_i$

soit encore à :
$a_{i1} x_1 + a_{i2} x_2 + \ \ldots\ + a_{in}\ x_n \leq b_i$

$$-a_{i1} x_1 - a_{i2} x_2 \ldots - a_{in} x_n \leq -b_i$$

On peut donc toujours se ramener à la forme énoncée ci-dessus, dite forme canonique que l'on peut synthétiser en utilisant des notations matricielles.

Posons en effet :

A, matrice (m,n) = $\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$

x, vecteur colonne $(n, 1) = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, b, vecteur colonne $(m, 1) = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$,

c, vecteur ligne $(1, n) = (c_1 \ldots c_n)$

le problème posé ci-dessus peut alors s'écrire :

$$Ax \leq b$$
$$x \geq 0$$
$$Max\ z = cx$$

Nous utiliserons souvent une autre forme que la forme canonique, appelée forme standard et obtenue de la façon suivante :

On transforme les inégalités $Ax \leq b$ en égalités en introduisant $m$ variables supplémentaires dites variables d'écart. En effet, le programme linéaire :

$$\text{(I)} \quad \begin{aligned} a_{11}x_1 + a_{12}x_2 + \ \cdots\cdots\ + a_{1n}\ x &\leq b_1 \\ a_{m1}x_1 + a_{m2}x_2 + \ \cdots\cdots\ + a_{mn}\ x_n &\leq b_m \\ x_1 x_2 \ \cdots\ x_n &\geq 0 \end{aligned}$$

Max $Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$ est équivalent au suivant :

$$\text{(II)} \quad \begin{aligned} a_{11}x_1 + a_{12}x_2 + \ \cdots\cdots\ + a_{1n}\ x_n + x_{n+1} &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ \cdots\cdots\ + a_{2n}\ x_n + x_{n+2} &= b_2 \\ a_{m1}x_1 + a_{m2}x_2 + \ \cdots\cdots\ + a_{mn}\ x_n + x_{n+m} &= b_m \\ x_1, x_2 \cdots x_n, x_{n+1} \ \ldots\ x_{n+m} &\geq 0 \end{aligned}$$

Max $Z' = c_1x_1 + c_2x_2 + \ldots + c_nx_n + 0x_{n+1} + \ldots + 0x_{n+m}$

(Les contraintes $x_{n+1} \ldots x_{n+m} \geq 0$ proviennent des inégalités $\leq$ du programme (I))

Les variables $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ sont appelées variables d'écart, par opposition aux variables $x_1, x_2 \ldots x_n$ appelées variables principales.

La forme (II), où toutes les inégalités sont transformées en égalités est appelée forme standard.

Si l'on utilise les notations matricielles pour un programme ramené à la forme standard (après adjonction de $m$ variables d'écart), on obtient

$$Ax = b$$
$$x \geq 0$$
$$Max\ z = cx$$

où

$A$ est une matrice $(m, m+n)$

$x$ un vecteur $(m+n, 1)$

$b$ un vecteur $(m, 1)$

$c$ un vecteur $(1, m+n)$

Pour un programme, mis sous forme canonique ou standard, on appellera :

- solution réalisable tout vecteur x satisfaisant toutes les contraintes (y compris celles du type $x_j \geq 0$)
- solution optimale toute solution réalisable maximisant la fonction économique.

### 1.2.2. Illustration économique des programmes linéaires

Comme nous l'avons dit, il n'est pas dans notre propos d'analyser ici la programmation linéaire sous l'angle de ses applications. Cependant il est intéressant de dégager les grands types de problèmes économiques auxquels cette technique peut se rattacher, car cette opération permet d'affecter une signification à bon nombre de raisonnements que nous ferons par la suite et donc de mieux comprendre leur mécanisme.

Le champ d'application des programmes linéaires est très diversifié. Cela dit, on peut signaler deux grands types de problèmes qui relèvent de la programmation linéaire.

1) Considérons une unité de production qui doit produire *n* biens différents en quantités $x_1, x_2, \ldots x_n$ et qui dispose de *m* facteurs de production en quantités limitées $b_1, b_2 \ldots b_m$. $a_{ij}$ représente la quantité du facteur de production *i* nécessaire pour produire une unité du bien $j$. Enfin $c_j$ est le bénéfice unitaire apporté par la production du bien $j$.

   L'objectif visé est d'établir le programme de production $x_1, x_2, \ldots x_n$ de façon à maximiser le profit total, compte tenu des contraintes de limitation des facteurs de production.

   Ce problème s'écrit bien sous la forme du programme linéaire :

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1,2 \ldots n$$

$$x_j \geq 0 \qquad j = 1, 2 \ldots n$$

$$\text{Max } Z = \sum_{j=1}^{n} c_j x_j$$

2) À présent, considérons que l'unité de production doit produire *m* biens dans des quantités au moins égales à $b_1, b_2 \ldots b_m$. ($b_i$ représentant la demande du bien $i$). $a_{ij}$ est la quantité de bien $i$ produite à partir d'une unité du facteur de production $j$. Enfin, le coût imputable à une unité du facteur de production $j$ est $c_j$.

   Le problème est de satisfaire la demande sur chaque produit en minimisant le coût de production, soit :

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad i = 1,2 \ldots n$$

$$x_j \geq 0 \qquad j = 1,2 \ldots n$$

$$Min\ Z = \sum_{j=1}^{n} c_j x_j$$

Un problème du premier type est par exemple le cas de l'entreprise de camions que nous avons étudié au premier paragraphe.

Un exemple célèbre de problèmes du second type est le problème du régime alimentaire: un éleveur doit fournir à ses animaux certaines quantités journalières d'éléments de base (lipides, protides, ou glucides) ; il peut acheter un certain nombre d'aliments, qui possèdent ces éléments de base dans des proportions variables. Quelles quantités des différents aliments doit-il acheter pour assurer aux animaux une nourriture satisfaisante et minimiser le coût total d'achat des aliments ?

On remarquera que l'écriture d'un problème sous la forme d'un programme linéaire nécessite un grand nombre d'hypothèses (en particulier les hypothèses de linéarité et d'indépendance des productions). L'emploi de tels outils est inséparable de l'examen attentif de la signification et de la validité de ces hypothèses.

## 1.3. THEOREMES GENERAUX SUR LES PROGRAMMES LINEAIRES

Nous allons donner quelques résultats généraux sur la programmation linéaire permettant de comprendre la technique de résolution exposée par la suite, à savoir la méthode du simplexe.

Nous commencerons par rappeler quelques éléments de vocabulaire relatifs aux ensembles convexes dans $R^P$.

### 1.3.1. Rappels sur les ensembles convexes

Un sous-ensemble $D$ de $R^P$ est dit convexe si pour tout $a$ et tout $b$ appartenant à $D$, le point $c$ est défini par

$c = \alpha a + \beta b$ appartient également à $D$ pour tout couple

$(\alpha, \beta) \in R^P$ *avec* $\alpha$ *et* $\beta \geq 0$ *et* $\alpha + \beta = 1$.

(autrement dit, tout point du segment $ab$ appartient à $D$).

Un point extrême $c$ d'un ensemble convexe $D$ est un point appartenant à $D$ tel qu'on ne puisse pas trouver deux points $a$ et $b$ tels que

$$c = \alpha a + \beta b$$

Avec $\alpha + \beta = 1$ *et* $0 \leq \alpha \leq 1$

$0 \leq \beta \leq 1$

(plus trivialement, $c$ n'appartient à aucun segment contenu dans $D$, sauf s'il en est une des extrémités).

Si on reprend le petit exemple analysé précédemment (l'entreprise de camions), le domaine $R^2$ ensemble des productions réalisables est convexe : c'est le polygone $OABCD$ (figure 1). Les points extrêmes sont ici les points $OABCD$, sommets du polygone : ils sont en nombre fini.

Par contre, si l'on dessine un cercle dans $R^2$, on obtient également un domaine convexe, mais tous les points de la circonférence sont des points extrêmes ; ils sont une infinité.

Rappelons également que si l'on a *n* points $a_1, a_2, \ldots a_n$ de $R^P$, une combinaison linéaire convexe de ces *n* points est un point $a$ qui s'écrit

$$a = \lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_n a_n$$

avec

$$\lambda_i \in R^+ \; et \; \sum_{i=1}^{n} \lambda_i = 1$$

A présent, nous allons démontrer un certain nombre de théorèmes sur la programmation linéaire en utilisant la forme standard, c'est-a-dire la forme obtenue après addition des variables d'écart transformant les inéquations en équations.

Nous avons donc le programme

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \qquad (1)$$

Max $z = cx$

avec $A$ matrice $(m, m+n)$

$x$ vecteur $(m+n, 1)$

$b$ vecteur $(m, 1)$

$c$ vecteur $(1, m+n)$

Nous travaillerons donc dans l'espace $R^{m+n}$

Toute solution réalisable du programme est un vecteur $x \in R^{m+n}$ satisfaisant aux contraintes (1).

### 1.3.2. Convexité du domaine des solutions réalisables

**Théorème I :** L'ensemble des solutions réalisables du P.L (programme linéaire) est convexe.
En effet soient deux solutions réalisables $x^1 \; et \; x^2 \in R^{m+n}$, on a :
$x^1 \geq 0$ et $x^2 \geq 0$
et $Ax^1 = b \; et \; Ax^2 = b$
prenons alors un point $x^3 = \alpha x^1 + \beta x^2$
avec $\alpha \geq 0, \beta \geq 0 \; et \; \alpha + \beta = 1$
On a :

1) $x^3 \geq 0$
2) $Ax^3 = \alpha Ax^1 + \beta Ax^2 = \alpha b + \beta b = b$

Donc $x^3$ est solution réalisable, ce qui assure la convexité du domaine des solutions réalisables.

***Remarque importante*** : de même, si l'on se place dans $R^n$ et si l'on se limite aux variables principales, (ensemble des x tels que Ax $\leq$ b) est également convexe.
Nous allons maintenant essayer de caractériser les points extrêmes du domaine des solutions réalisables.

### 1.3.3. Points extrême du domaine convexe des solutions réalisables

Pour démontrer le théorème suivant, nous allons écrire les équations du programme linéaire dans le langage vectoriel.

Soit les vecteurs $P_1, P_2, P_{n+m}$ de $R^m$ formés par les colonnes de la matrice $A$.

$P_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ . \\ . \\ . \\ a_{m1} \end{pmatrix}$, $P_2 = \begin{pmatrix} a_{1i} \\ a_{2i} \\ . \\ . \\ . \\ a_{mi} \end{pmatrix}$, etc..

Le programme II peut s'écrire :

$$\sum_{j=1}^{n+m} x_j \, P_j = b$$

$x_j \geq 0$

et la maximisation de la fonction suivante :

$$z = \sum_{j=1}^{n+m} c_j \; x_j$$

**Théorème II :** Si l'on trouve $k$ vecteurs P$i$ linéairement indépendants $(k \leq m)$, que l'on désignera, après renumérotation, par $P_1, P_2 \ldots P_K \; et \; k$ valeurs positives $x_1, x_2, \ldots x_k$ avec $x_1P_1 + x_2P_2 + \cdots + x_kP_k = b$ alors le point :

$x = \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_k \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$ est un point extrême du domaine des solutions réalisables.

Réciproquement, un point extrême de ce domaine, de coordonnées $x_1, x_2, \ldots x_{n+m}$ $(x_i \geq 0)$ est tel que les vecteurs $P_i$ associés aux $x_i$ strictement positifs sont indépendants.

Démontrons la première partie du théorème : Soit en effet $x_1, x_2, \ldots x_k \geq 0$ avec k ≤ m, $P_1, P_2 \ldots P_k$ indépendants et : $x_1P_1 + x_2P_2 + \cdots + x_kP_k = b$

Le point $\overline{x} = \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_k \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$ est solution réalisable du P.L. Supposons que $\overline{x}$ ne soit pas un point extrême du domaine des solutions réalisables, que nous appellerons D. C'est que l'on peut trouver deux points $x^1$ et $x^2$ (solutions réalisables) tels que :

$$x = \alpha x^1 + (1-\alpha) x^2$$

avec

$$0 < \alpha < 1$$

Ce qui impose d'abord que les $n+m-k$ dernières composantes de $x^1$ et $x^2$ soient nulles.

Donc :

$$x^1 = \begin{pmatrix} x^1{}_1 \\ x^1{}_2 \\ . \\ . \\ x^1{}_k \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \text{ et } x^2 = \begin{pmatrix} x^2{}_1 \\ x^2{}_2 \\ . \\ . \\ x^2{}_k \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$$

On a, puisque de $x^1$ et $x^2$ sont solutions réalisables :

$$x_1^1 P_1 + x_2^1 P_2 + \cdots + x_k^1 P_k = b$$

et

$$x_1^2 P_1 + x_2^2 P_2 + \cdots + x_k^2 P_k = b$$

mais, puisque $P_1$, $P_2 \ldots P_K$ sont linéairement indépendants, $b$ s'exprime d'une façon unique en fonction de ceux-ci; en conséquence :

$$x_1^1 = x_1^2 = x_1; x_2^1 = x_2^2 = x_2 \cdots; \quad x_k^1 = x_k^2 = x_k$$

d'où : $\alpha = 0$ ou 1

Donc, $\overline{x}$ est point extrême de D;

A présent, démontrons la réciproque :

Soit $\overline{x}$ un point extrême du domaine D, $\overline{x} \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ a_{n+m} \end{pmatrix}$

on a :

$$\sum_{i=1}^{n+m} x_i \, P_i = b$$

Parmi les $x_i$, un certain nombre sont positifs, les autres nuls. Supposons qu'il y en ait $r$ positifs. Après renumérotation, on peut écrire :

$$\sum_{i=1}^{r} x_i \; P_i = b$$

Supposons que les $P_i$ $(i = 1, 2, \ldots r)$ soient linéairement dépendants, alors, il existe des $\lambda_i (i = 1, 2, \ldots r)$non tous nuls tels que :

$$\sum_{i=1}^{r} \lambda_i \; P_i = 0$$

soit alors un nombre μ arbitraire; on a :

$$\sum_{i=1}^{r} x_i \; P_i + \mu \sum_{i=1}^{r} \lambda_i \; P_i = b$$

et

$$\sum_{i=1}^{r} x_i \; P_i - \mu \sum_{i=1}^{r} \lambda_i \; P_i = b$$

Considérons les deux points :

$$x^1 = \begin{pmatrix} x_1 + \mu\lambda_1 \\ x_2 + \mu\lambda_2 \\ . \\ . \\ x_r + \mu\lambda_r \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \text{ et } x^2 = \begin{pmatrix} x_1 - \mu\lambda_1 \\ x_2 - \mu\lambda_2 \\ . \\ . \\ x_r - \mu\lambda_r \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$$

Les $x_1, x_2 \ldots x_r$ étant toujours positifs, on peut toujours choisir μ de telle façon que $x_i + \mu\lambda_i$ et $x_i - \mu\lambda_i$ restent positifs pour tout *i*, donc que $x^1$ *et* $x^2$ soient des solutions réalisables. Mais, on a bien évidemment :

$$\bar{x} = \frac{1}{2} x^1 + \frac{1}{2} x^2$$

ce qui contredit l'hypothèse que $\overline{x}$ est point extrême.

En conséquence, les $P_i$ $(i = 1,2, \ldots r)$ sont linéairement indépendants. Le théorème II est alors complètement démontré.

Les implications de ce théorème sont les suivantes :

comme nous sommes dans $R^m$ en ce qui concerne les $P_i$ et comme un point extrême est tel que les $P_i$ associés aux $x_i$ strictement positifs sont indépendants, il en résulte que pour un point extrême, le nombre de $x_i$ strictement positifs est au plus égal à $m$.

On en déduit que pour obtenir un point extrême, il suffirait *a priori* d'extraire des $m + n$ vecteurs $P_i$ vecteurs indépendants avec $r \leq m$ et de résoudre le système d'équations:

$$\sum_{i=1}^{r} x_i \, P_i = b$$

en espérant que les $x_i$ trouvés soient positifs ou nuls.

Cependant, dans les cas où $r < m$ ce système est en général impossible. On essaiera toujours de se ramener au cas où $r = m$ grâce au théorème suivant :

**Théorème III :** À un point extrême du domaine des solutions réalisables peut être associé un ensemble de $m$ vecteurs $P_i$ linéairement indépendants.

Nous remarquerons tout d'abord que, grâce à l'addition des variables d'écart[1], le rang des vecteurs $P_1, P_2, P_{n+m}$ est égal à $m$. En effet :

$$P_{n+1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \; P_{n+2} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \; P_{n+m} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$ sont linéairement indépendants

Soit alors un point extrême $\overline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \\ 0 \\ 0 \\ 0 \end{pmatrix}$

avec $r < m$. Les vecteurs $P_1, P_2, P_r$ sont linéairement indépendants, en vertu du théorème II.

Si nous ne pouvons compléter ces vecteurs indépendants que par $r' - r$ autres vecteurs indépendants $P_{r+1}, P_{r+2}, \ldots P_{r'}$ avec $r' < m$ c'est que les $n + m - r'$ autres vecteurs s'expriment en fonction des r' vecteurs $P_1, P_2, P_r$, donc que le rang du système $P_1, P_2, P_{n+m}$ est $r' < m$ et non $m$ ce qui est contraire à la remarque faite plus haut. D'où le théorème III.

---

[1] Dans le cas où la forme standard $Ax = b$ ne résulte pas de l'addition systématique de variables d'écart (par exemple lorsque certaines contraintes sont des égalités) on supposera que le rang de $A$ est $m$. Les résultats dégagés seront alors valables pour ce cas.

En conclusion, un point extrême du domaine D est obtenu en choisissant $m$ vecteurs $P_1, P_2, P_m$ indépendants parmi les $n+m$ vecteurs $P_1, P_2, P_{n+m}$ et en résolvant le système de Cramer :

$$x_1P_1 + x_2P_2 + \cdots + x_mP_m = b$$

Le point $(x_1, x_2, \ldots x_m, 0 \ldots 0)$ obtenu est un point extrême si les $x_1, x_2, \ldots x_m$ sont positifs ou nuls.

Si un certain nombre de $x_i$ parmi les $x_1, x_2, \ldots x_m$ sont nuls, la solution est dite dégénérée.

La solution $(x_1, x_2, \ldots x_m, 0 \ldots 0)$ correspondant à un point extrême est dite solution de base du programme linéaire.

$x_1, x_2, \ldots x_m$ sont appelées les variables de base pour la solution de base considérée, $x_1, x_{m+1}, \ldots x_{m+n}$ les variables hors base.

Une autre conclusion très importante de ces deux théorèmes est la suivante : le nombre de points extrêmes du domaine des solutions réalisables, donc le nombre de solutions de base est fini. En effet ce nombre est borné par $C_{n+m}^{m} = \frac{(n+m)!}{n!m!}$ nombre de système de $m$ vecteurs que l'on peut extraire des vecteurs $P_1, P_2 \ldots P_{n+m}$

### 1.3.4. Définition géométrique des points extrêmes du domaine des solutions réalisables.

D'après ce que l'on vient de voir, un point extrême est obtenu en annulant $n$ variables $x_i$ parmi les $n+m$ variables du programme mis sous forme standard, et en résolvant le système d'équations par rapport aux $m$ variables restantes (variables de base).

Sur ces $n$ variables nulles (variables hors base) un certain nombre $p$ sont des variables principales et $n-p$ sont des variables d'écart.

Soit une variable principale hors base $x_i = 0$ est l'équation d'un plan de coordonnées dans $R^n$ (espace des variables principales).

Si maintenant on a une variable d'écart $x_{n+j}$ hors base, soit $x_{n+j} = 0$, c'est que la contrainte correspondant à cette variable d'écart est saturée. On a donc l'équation :

$$a_{j1}\, x_1 + a_{j2}\, x_2 + \cdots + a_{jn}\, x_n = b_j$$

qui est l'équation d'un hyperplan dans l'espace des variables principales ($R^n$).

Donc, un point extrême du domaine D correspond à l'intersection de n hyperplans dans l'espace $R^n$ des variables principales pris soit dans les hyperplans de coordonnées $(xi = 0)$ soit dans les hyperplans correspondants aux contraintes du programme linéaire

$$\left( \sum_{i=1}^{n} a_{ij}\ x_j = b \right).$$

On trouvera une illustration de ces points dans l'exemple des carmions exposé précédemment, où $n = 2\ et\ m = 3$. On a en effet déjà remarqué qu'un sommet du polygone convexe, donc un point extrême du domaine des solutions réalisables, est donné par l'intersection de deux droites (soit du type $x_1\ ou\ x_2 = 0$), soit du type $ax_1 + bx_2 = 0$

Par ailleurs, si on ajoute les variables d'écart $x_3, x_4, x_5$ avec

$$\begin{array}{ccccccc} x_1 & + & 3x_2 & + & x_3 & = & 450 \\ 2x_1 & + & x_2 \cdots & + & x_4 & = & 350 \\ x_1 & + & x_2 \cdots\cdot & + & x_5 & = & 200 \end{array}$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

On voit que chaque point extrême du domaine des solutions réalisables est caractérisé par 3 variables positives sur les 5 et les deux autres nulles, ce qui correspond bien aux solutions de base définies plus haut. Par exemple :

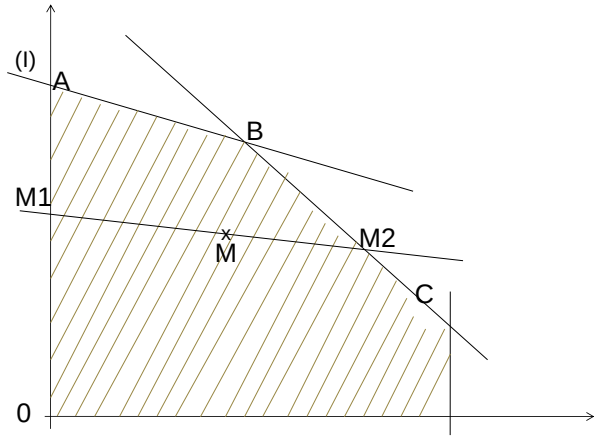
$$0 = \begin{pmatrix} 0 \\ 0 \\ 450 \\ 350 \\ 200 \end{pmatrix} A = \begin{pmatrix} 0 \\ 150 \\ 0 \\ 100 \\ 50 \end{pmatrix} etc., etc.$$

Revenons maintenant à l'exposé général pour souligner l'importance des points extrêmes du domaine *D*.

### 1.3.5. Solutions réalisables et solutions de base

**Théorème IV:** Si le domaine *D* des solutions réalisables est borné, toute solution réalisable *x* peut se mettre sous la forme d'une combinaison linéaire convexe des solutions de base $x^1, x^2, \ldots x^p$.

La démonstration de ce théorème étant assez longue, nous renvoyons le lecteur intéressé à une annexe placée à la fin de ce chapitre, où les principes du raisonnement lui seront exposés. On peut par ailleurs proposer une justification intuitive du résultat en prenant un polygone convexe de $R^2$ :




Prenons en effet un point $M$ à l'intérieur du polygone et traçons une droite quelconque passant par $M$. Cette droite coupe les côtés du polygone en deux points, ici $M_1$ et $M_2$. On peut écrire (synthétiquement) :

$$M = \lambda\, M_1 + (1-\lambda)\, M_2$$

$$avec\ \ 0 < \lambda < 1$$

$M_1$ lui-même peut s'écrire :

$$M_1 = \mu_1\, 0 + (1-\mu_1)\, A$$

$$0 < \mu_1 < 1$$

$$M_2 = \mu_2\, B + (1-\mu_2)\, C$$

$$0 < \mu_2 < 1$$

d'où

$$M = \lambda\, \mu_1 0 + \lambda(1-\mu_1)\, A + (1-\lambda)\, \mu_2\, B + (1-\lambda)(1-\mu_2)\, C$$

Dans cette expression, tous les coefficients numériques sont compris entre 0 et 1.

Par ailleurs,

$$\lambda\mu_1 + \lambda(1-\mu_1) + (1-\lambda)\, \mu_2 + (1-\lambda)(1-\mu_2) = 1$$

Donc, $M$ apparaît bien sur ce petit exemple comme combinaison linéaire convexe des sommets du polygone convexe.

Le théorème proposé est une généralisation à $R^n$ de ce phénomène. Sa démonstration est d'ailleurs fondée sur l'illustration que nous venons de proposer.

La restriction portant sur le caractère borné du domaine des solutions réalisables est fondamentale pour le théorème IV, comme on pourrait le voir en lisant l'annexe

Pour la suite de l'exposé, nous considèrerons toujours des domaines *D* bornés : cette hypothèse n'est pas gênante en recherche opérationnelle, où l'on est censé manier des grandeurs économiques; elle le serait sans doute davantage pour un exposé mathématique.

Le domaine *D* des solutions réalisables d'un P.L. porte, lorsqu'il est borné, le nom particulier de polyèdre convexe. Nous utiliserons ce terme dorénavant. Les points extrêmes, en nombre fini, seront les sommets du polyèdre.

### 1.3.6. Fonction économique et sommets du polyèdre convexe

L'analyse que l'on vient de faire sur les points extrêmes (ou sommets) du domaine des solutions réalisables et la relation existant entre n'importe quelle solution réalisable et ses points extrêmes trouve sa pleine justification dans le théorème suivant :

**Théorème V** : La fonction économique est maximale en au moins un des sommets du polyèdre convexe; si elle est maximale en plus d'un sommet, elle est maximale en toute combinaison linéaire de ces sommets.

Soit $x^1, x^2, \dots x^p$. les sommets du polyèdre convexe et soit un point $\overline{x}$ quelconque du polyèdre convexe ($\overline{x}$ solution réalisable); supposons que $\overline{x}$ ne soit pas un sommet.

D'après le théorème IV, $\overline{x}$ peut s'écrire sous la forme d'une combinaison linéaire convexe de $x^1, x^2, \dots x^p$.

$$\overline{x} = \sum_{i=1}^{p} \lambda_i \; x^i$$

avec $0 \leq \lambda_i \leq 1$

$$et \sum_{i=1}^{p} \lambda_i = 1$$

Appelons $Z(\overline{x})$ la valeur de la fonction économique pour une solution réalisable $\overline{x}$.

$Z(\overline{x}) = c\overline{x} = \sum_{i=1}^{p} \lambda_i \qquad cx^i = \sum_{i=1}^{p} \lambda_i \, Z(x^i)$

Soit un sommet $x^k$ tel que $Z(x^k) \geq Z(x^j)$

pour tout $j \neq k$, alors

$$Z(\overline{x}) \leq \sum_{i=1}^{p} \lambda_i \; Z(x^k)$$

$$Z(\overline{x}) \leq Z(x^k)$$

Donc la fonction $Z(x)$ est maximale en un ou plusieurs sommets du polyèdre convexe.

S'il existe plusieurs sommets $x^1, x^2, \ldots x^q$ tels que

$$Z(x^1) = Z(x^2) = \cdots Z(x^q) > Z(x^i)$$

avec $i \notin 1,2 \ldots q$ toute combinaison linéaire convexe des $x^1, x^2, \ldots x^q$

$$\bar{x} = \sum_{i=1}^{q} \lambda_i x^i \text{ avec } \sum_{i=1}^{q} \lambda_i = 1$$

est telle que

$$Z(\bar{x}) = \sum_{i=1}^{q} \lambda_i \, Z(x^i)$$

$$Z(\bar{x}) = Z(x^1) = Z(x^2) = \cdots Z(x^q)$$

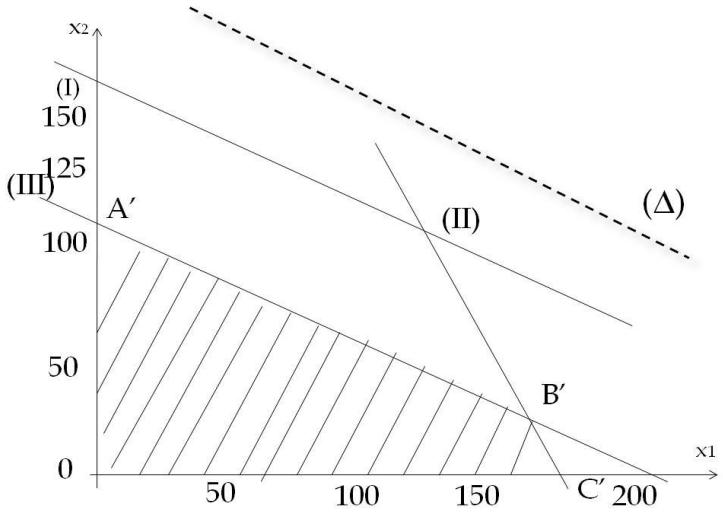
Donc la fonction Z est également maximale en $\bar{x}$.

Sur l'exemple numérique traité en début du cours (voir figure 1), on avait effectivement constaté que le maximum de la fonction économique était réalisé en B, sommet du polygone convexe des solutions réalisables, et on en avait vu l'interprétation géométrique.

Par ailleurs, supposons que pour l'atelier III, les temps d'assemblage ne soient pas d'une heure pour les camions de type A et d'une heure pour les camions de type B, mais respectivement d' une heure et de deux heures. La contrainte (III) devient :

$$x_1 + 2x_2 \leq 200$$

et le domaine des solutions réalisables :




On constate que la contrainte (I) est superflue.

Par ailleurs, le segment $A'B'$ est parallèle à la direction ( $\Delta$) de la fonction économique. Le maximum de celle-ci est trouvé en $A'B'$, et en tout point du segment $A'B'$, donc en tout point s'exprimant sous la forme d'une combinaison linéaire convexe $A'et\ B'$

**Remarque** : la circonstance suivant laquelle il existe une infinité de solutions optimales, comme précédemment, est appelée dégénérescence de second type.

Le terme de dégénérescence de premier type est réservé au cas, déjà examiné, où une solution de base est constituée de moins de $m$ variables non nulles.

Dans ce cas, il y a plus de $n$ variables nulles, ce qui veut dire que le sommet dans $R^n$ correspondant à la solution de base est l'intersection de plus en plus de $n$ hyperplans pris soit dans les hyperplans $x_j = 0$, soit dans les hyperplans

$$\sum_{j=1}^{n} a_{ij} x_j = b$$

## 1.4. INTRODUCTION A L'ALGORITHME DU SIMPLEXE

Nous savons donc que le maximum de la fonction économique est trouvé en un sommet (éventuellement plusieurs) du polyèdre convexe des solutions réalisables.

Par ailleurs, nous savons trouver théoriquement ces sommets : il suffit de choisir $m$ vecteurs $P_i$ , indépendants parmi $P_1, P_2, P_{n+m}$ et de résoudre le système,

$$\sum_{i=1}^{m} x_i P_i = b$$

Ce système est de Cramer pour les variables $x_i$ ; si ces dernières sont toutes positives ou nulles, on a bien obtenu ainsi un sommet du polyèdre (sinon, on recommence en choisissant $m$ autres vecteurs $P_i$). Pour la solution de base ainsi trouvée, on peut calculer la valeur de la fonction économique. On pourrait penser alors calculer cette valeur pour tous les sommets du polyèdre, et comparer entre elles les grandeurs trouvées pour obtenir le maximum. Cette méthode est très lourde.

Supposons en effet, pour fixer les idées, que *m =10* et *n = 20* (ce qui constitue un petit programme linéaire, si on le compare à ceux qui sont utilisés couramment dans la pratique).

Il faut essayer $C_{10}^{30}$ systèmes de 10 vecteurs indépendants, ce qui constitue plus de 27000 000 systèmes de Cramer à résoudre.

On est donc confronté à un problème hautement combinatoire et, pour le résoudre, il est nécessaire de disposer d'un algorithme, c'est-à-dire d'une procédure répétitive permettant, de progresser rapidement vers la solution optimale.

Cet algorithme dit du simplexe sera exposé dans le chapitre suivant. Nous nous contenterons ici d'en donner les principes. Nous commencerons par profiter des renseignements dégagés par les théorèmes I à V pour proposer une autre écriture d'un programme linéaire, à savoir l'écriture matricielle (alors que nous avons surtout utilisé jusqu'ici une écriture vectorielle).

### 1.4.1. Utilisation de l'écriture matricielle dans la programmation linéaire

Soit un programme linéaire mis sous forme standard

$$Ax = b$$

$$x \geq 0 \ (1)$$

$$Max \ Z = cx$$

$$Avec \ A \ la \ matrice \ (m, m+n)$$

$$b \ (m, 1)$$

$$x \ (m+n, 1)$$

$$c \ (1, n+m)$$

Supposons que nous disposions d'une solution de base du P.L., c'est-à-dire que nous ayons trouvé $m$ vecteurs indépendants de la matrice $A$. Rappelons que la solution de base est alors constituée par $m$ variables $x_j$ de base non nulles[2] (celles qui correspondent aux $m$ vecteurs indépendants) les $n$ autres variables étant nulles (variables hors base).

Nous appellerons $I$ l'ensemble des indices des variables de base, $\bar{I}$ l'ensemble des indices des variables hors base. On peut écrire

$$x = \begin{pmatrix} x_I \\ x_{\bar{I}} \end{pmatrix}$$

(après réordonnancement des variables)

$x_I$ sera le vecteur colonne des variables de base

$x_{\bar{I}}$ le vecteur colonne des variables hors base

($x_I$ vecteur colonne $(m, 1), x_{\bar{I}}$ valeur colonne $(n, 1)$ )

De même, on peut décomposer $A$ en $A = (A_I, A_{\bar{I}})$

avec $A_I$ matrice $(m, m)$ et $A_{\bar{I}}$ matrice $(m, n)$.

$A_I$ est la matrice extraite de $A$ en prenant les $m$ vecteurs colonnes correspondant aux variables de base. On sait que $A_I$ est une matrice inversible, puisque les vecteurs colonnes la composant sont indépendants.

$A_I$ sera appelée matrice de base; $A_{\bar{I}}$ qui correspond aux variables hors base, sera appelée matrice hors base.

---

[2] Sauf dégénérescences.

De la même façon on peut écrire :

$C = (C_I, C_{\bar{I}})$

avec $c_I$ vecteur ligne $(1, m)$

et $c_{\bar{I}}$ vecteur ligne $(1, n)$

Dans ces conditions, le programme linéaire (1) s'écrit :

$A_I x_I + A_{\bar{I}} x_{\bar{I}}$

$$x_I, x_{\bar{I}} \geq 0 \qquad (2)$$

$Max\ z = c_I x_I + c_{\bar{I}} x_{\bar{I}}$

Comme $A$ est irréversible, la première équation donne : $x_I = [A_I]^{-1}[b - A_{\bar{I}} x_{\bar{I}}]$

ou $x_I = [A_I]^{-1} b - [A_I]^{-1} A_{\bar{I}} x_{\bar{I}}]$ (3)

Cette écriture est absolument générale : elle ne préjuge en rien des valeurs des composantes de $x_I$ et $x_{\bar{I}}$: c'est une simple transformation des équations du programme (1).

Pour la solution de base correspondant à l'ensemble d'indices $I$, les valeurs des variables du PL sont données par :

$x_{\bar{I}} = 0$ et $x_I = [A_I]^{-1} b$

Sur la formule (3), on constate le fait suivant, très important :

**lorsque l'on dispose d'une base, les variables de base s'expriment linéairement en fonction des variables hors base.**

Posons : $[A_I]^{-1} b = t = (t_i)$ et : $[A_I]^{-1} A_{\bar{I}} = T = (t_{ij})$

$t$ est un vecteur colonne $(m, 1)$ et $T$ une matrice $(m, n)$

Pour tout $i \in I$ la formule (3) donne :

$$x_i = t_i - \sum_{j \in \bar{I}} t_{ij}\, x_j \qquad (4)$$

pour $i \in I$

**Remarque essentielle** : $t_i$ est positif, puisque valeur d'une variable de base d'une solution de base.

Par ailleurs, on a, d'après (2) :

$z = c_I x_I + c_{\bar{I}} x_{\bar{I}}$ combinons cette formule avec (3).

Il vient $Z = c_I [A_I]^{-1} b - c_I [A_I]^{-1} A_{\bar{I}} x_{\bar{I}} + c_{\bar{I}} x_{\bar{I}}$

soit

$Z = c_I[A_I]^{-1}b + (c_{\bar{I}} - c_I[A_I]^{-1}A_{\bar{I}})x_{\bar{I}}$ (5)

Cette écriture, là aussi, est absolument générale.

Pour le sommet correspondant à la base donnée par l'ensemble d'indices $I$

$Z_0 = C_I[A_I]^{-1}b$

donne la valeur de la fonction économique en ce sommet.

**Par ailleurs, on voit sur (5) que, lorsqu'on dispose d'une base, la fonction économique peut toujours s'exprimer linéairement en fonction uniquement des variables hors base.**

La formule (5) donne, si l'on pose

$$z_j = \sum_{i \in I} t_{ij}\, c_i$$

pour $j \in \bar{I}$

$$Z - Z_0 = \sum_{j \in \bar{I}} (c_j - z_j)\, x_j \qquad (6)$$

Avec

$$x_j = \sum_{i \in I} t_{ij}\, c_i$$

pour $j \in \bar{I}$

Les quantités : $\delta_j = c_j - z_j$ pour $j \in \bar{I}$ sont appelées gains marginaux.

### 1.4.2. Passage d'un sommet à un autre

Supposons donc que nous disposions d'un sommet, ou, ce qui est la même chose, d'une solution de base relative à un ensemble d'indices $I$.

Pour ce sommet et pour $i \in I$ on a, d'après ce qu'on a vu :

$x_i = t_i$ avec $t = [A_I]^{-1}b$

$x_j = 0$ pour $j \in I$

$Z = Z_0$ avec $Z_0 = c_I[A_I]^{-1}b$

Par ailleurs, d'après les formules (4) et (6), $x_i$ et $Z$ s'expriment en fonction des variables hors base, qui sont nulles pour le sommet correspondant :

$$x_i = t_i - \sum_{j \in \bar{I}} t_{ij}\, x_j \qquad (7)$$

$$Z - Z_0 = \sum_{j \in \bar{I}} (c_j - z_j)\, x_j \qquad (8)$$

avec

$$z_j = \sum_{i \in I} t_{ij}\, c_i$$

À partir du moment où l'on a $I$, on peut obtenir $t_i, t_{ij}, z_j, Z_0$ grâce à de simples calculs matriciels.

Supposons que pour un $j \in I$ on ait $c_j - z_j > 0$

Rendons la variable correspondante $c_j$ positive, en laissant les autres variables hors base nulles, le but de cette opération étant d'augmenter la fonction économique : en effet si $\Delta x_j$ est l'accroissement de la variable $x_j$, les autres variables hors base restant nulles, l'accroissement $\Delta Z$ de la fonction économique est :

$\Delta Z = (c_j - z_j)\Delta x_j > 0$ puisque $(c_j - z_j) > 0$

D'après l'équation (7), pour que l'on ait dans ces conditions une solution réalisable, il faut que :

$x_i > 0 \; \forall \; i \in I$ et $x_i = t_i - t_{ij} x_j \; \forall i \in I$

on en déduit :

1) si $t_{ij} \leq 0$ pour tout $x_i = t_i - t_{ij} x_j$ est positif quelle que soit la valeur (positive) de $x_j$ ; $x_j$ peut donc croître indéfiniment ainsi que les variables $x_1, x_2, \ldots x_m$ et on obtient une solution réalisable à $m + 1$ variables infinies : le domaine n'est pas borné, ce qui est contraire à l'hypothèse faite en II-3-5.
2) Il existe donc au moins un $t_{ij} > 0$ ; on doit alors avoir, pour respecter $x_i > 0 \; \forall i \in I$
   $x_j \leq \frac{t_j}{t_{ij}}$ pour tout $i \in I$ tel que $t_{ij} > 0$ ou encore $x_j \leq min \frac{t_j}{t_{ij}}$ i/$t_{ij} > 0$
   si l'on fait croître $x_j$ jusqu'à sa limite :
   $x_j^0 \leq min \frac{t_j}{t_{ij}}$ i/$t_{ij} > 0$ et si l'on appelle $i_0$ l'indice tel que $\frac{t_{i0}}{t_{i0j}} \leq \frac{t_i}{t_{ij}}$ pour tout $i \in I$
   on voit que la variable $x_{i0}$ s'annule, si $x_j = x_j^0$.

Cette opération a donc consisté à rendre positive une variable hors base pour annuler une variable de base.

À la fin, on a donc toujours $m$ variables non nulles et $n$ variables nulles, donc une solution de base : on est passé d'un sommet du polyèdre à un autre.

**Remarque :** il faut vérifier que les vecteurs colonnes de la matrice des contraintes associées aux $m$ variables nouvelles de base sont bien indépendants.

Cela devient parfaitement évident si l'on prend soin de prendre pour contraintes du programme linéaire les équations (7) sous la forme :

$$x_i + \sum_{j \in \bar{I}} t_{ij}\, x_j = t_i \qquad \forall i \in I \qquad (7)$$

Ces équations, encore une fois, sont absolument équivalentes aux équations initiales du programme (1). Alors, la matrice de base initiale (les $x_i$ de base, avec $i \in I$) est la matrice unité $m \times m$

1 0 . . . 0

0 1 . . . 0

0 0 1 . . 0

0 0 1 0

0 . . . 1 0

0 0 0 0 0 1

Si l'on fait sortir la variable $x_{i0}$ de la base et si on fait entrer dans le base $x_j$ avec $j \in \bar{I}$, on obtient la nouvelle matrice de base :

$i_0$ ↘

1 0 . $t_{ij}$ . 0

0 1 . . . 0

0 0 1 . . 0

0 0 $t_{i0j}$ 0

0 . . . 1 0

0 0 0 $t_{mj}$ 0 1

c'est-à-dire la matrice unité où l'on remplace la colonne $i_0$ par la colonne :

$$\begin{pmatrix} t_{ij} \\ . \\ t_{i0j} \\ . \\ . \\ t_{mj} \end{pmatrix}$$

Comme $t_{i0j} > 0$, le déterminant de cette nouvelle matrice n'est pas nul. On a donc bien une nouvelle matrice de base inversible.

Ce qu'il faut remarquer à présent, c'est que l'intérêt de cette opération provient du fait que l'on a choisi pour variable $x_j$ entrant dans la base une variable hors base telle que

$c_j - z_j > 0$ Or, si $x_j = x_j^0 > 0$, les autres variables hors base restant nulles, la nouvelle valeur $Z'_0$ de la fonction économique est :

$Z'_0 = Z_0 + (c_j - z_j)x_j^0$ et alors $Z'_0 > Z_0$

On a bien réussi par cette opération à augmenter la valeur de la fonction économique. Tout le principe de l'algorithme du simplexe est là :

1- partir d'une solution de base,
2- passer à une autre solution de base en faisant entrer une variable hors base dans la base et sortir une variable de base pour la mettre hors base, et en s'assurant que l'on améliore ainsi la fonction économique,
3- s'arrêter lorsque l'on ne peut plus améliorer la fonction économique.

En ce qui concerne le test d'arrêt, il est le suivant :

lorsque, pour tout $j \in I$, $c_j - z_j > 0$ *on* a atteint l'optimum qui est donné par la solution de base *I*.

Ce test est parfaitement évident si l'on prend le P.L sous la forme donnée par les équations (7) et (8) :

$$x_i + \sum_{j \in \bar{I}} t_{ij} x_j = t_i \,, \forall i \in I \; (avec \; t_i \geq 0$$

$x_i, x_j \geq 0$

$$Max \; Z - Z_0 = \sum_{j \in \bar{I}} (c_j - z_j) x_j$$

En effet, encore une fois, ce programme est équivalent au programme initial (1). La solution de ce programme, si $(c_j - z_j) \leq 0$ pour tout $j \in \bar{I}$ est bien évidemment

$x_j = 0 \; \forall J \in \bar{I}$ et donc $x_i = t_i$ i ∈ *I*

$Z = Z_0$ c'est-à-dire la solution de base considérée.

Les principes de l'algorithme du simplexe sont résumés dans l'ordinogramme de la page suivante.

**<u>Remarque importante</u> : lorsque l'on passe d'un sommet à un autre, la fonction économique devient comme on l'a vu :**

$Z'_0 = Z_0 + (c_j - z_j)x_j^0$

avec $x_j^0$ valeur de la variable entrée dans la base et $(c_j - z_j) > 0$ $c_j$, $z_j, x_j^0$ étant finis, on est sûr que, si l'on ne rencontre jamais de dégénérescences de premier type (variables de base nulles) l'algorithme du simplexe est convergent, puisqu'à chaque itération, la fonction économique s'accroît d'une quantité finie. Dans le cas contraire, il peut y avoir (dans des circonstances très rares) non convergence.

## ANNEXE

Principes de la démonstration du théorème IV :

*Si le domaine D des solutions réalisables est borné, toute solution réalisable peut se mettre sous la forme d'une combinaison linéaire convexe des solutions de base* $x^1, x^2, \cdots x^p$.

Remarquons tout d'abord qu'on peut effectivement parler des solutions de base (ou points extrêmes) $x^1, x^2, \ldots x^p$ puisque l'on a montré que ces solutions de base étaient en nombre fini.

Il s'agit de montrer que toute solution réalisable $x$ du P.L peut se mettre sous la forme

$$x = \sum_{i=1}^{p} \lambda_i x^i$$

avec $0 \leq \lambda_i \leq 1$ et $\sum_{i=1}^{p} \lambda_i$=1

Donnons les grandes lignes de la démonstration.

Soit donc une solution réalisable $x$. Numérotons $x_1, x_2, \ldots x_q$ les composantes strictement positives de $x$, les $n + m - q$ autres étant nulles. On a alors, avec les notations précédentes, $x_1 P_1, x_2 P_2, \ldots x_q P_q = b$

Si les vecteurs $P_1, P_{2,\ldots} P_q$ sont linéairement indépendants, $x$ est un point extrême de $D$. Supposons qu'il n'en soit pas ainsi. Nous allons mettre $x$ sous la forme :

$$x = \lambda E_1 + (1-\lambda)\, E_2$$

où $0 \leq \lambda \leq 1$ et où $E_1 et\, E_2$ sont deux solutions réalisables ayant au plus $q - 1$ composantes non nulles.

Pour ce faire, nous supposerons que le rang du système de vecteurs $P_1, P_{2,\ldots} P_q$ est $r(r < q)$ que $P_1, P_{2,\ldots} P_r$ sont linéairement indépendants, après renumérotation éventuelle. On a :

$x_1 P_1 + \cdots + x_r P_r +, x_{r+1} P_{r+1} \ldots x_q P_q = b$ (1)

pour choisir $E_1$, opérons de la façon suivante :

(1) peut toujours s'écrire, $\theta$ étant un nombre quelconque :

$x_1 P_1 + \cdots + \theta x_r P_r + \theta x_{r+1} P_{r+1} + (1-\theta) x_{r+1} P_{r+1} \ldots x_q P_q = b$ (2)

Or, comme $P_1, P_{2,\ldots} P_r$ sont indépendants et que le rang du système $P_1, P_{2,\ldots} P_q$ est $r$, on peut exprimer $P_{r+1}$ d'une façon unique en fonction de $P_1, P_{2,\ldots} P_r$ soit

$$P_{r+1} = \sum_{i=1}^{r} \propto_i P_i \quad (3)$$

(2) et (3) donnent alors :

(4)

$$x_1 + (1-\theta)\alpha_1 x_{r+1})P_1 + \cdots + (x_r + (1-\theta)\alpha_r x_{r+1})P_r + \theta x_{r+1}P_{r+1} + x_{r+2}P_{r+2} + \cdots + x_q P_q = b$$

Soit $E$ le point :

$$E = \begin{pmatrix} x_1 + (1-\theta)\,\alpha_1\, x_{r+1} \\ \vdots \\ x_r + (1-\theta)\,\alpha_r\, x_{r+1} \\ \theta\, x_{r+1} \\ x_{r+2} \\ \vdots \\ x_q \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

pour $\theta = 1,\ E = x$. Faisons croître alors $\theta$ de 1 à $+\infty$. Si les 2 premières composantes de $E$ étaient des fonctions croissantes de $\theta$ (tous les $\alpha_i < 0$), elles tendraient vers l'infini avec $\theta$, et cela voudrait dire que le domaine $D$ n'est pas borné, ce qui est contraire à l'hypothèse faite.

Il existe donc au moins un $\propto_i$ positif et une composante de $E$ décroissante avec $\theta$. Soit $\theta_i$ la plus faible valeur de $\theta$ qui annule une des $r$ premières composantes de $E$, la valeur de $\theta_1$ est :

$$\theta_1 = \min_{i \mid \alpha_i > 0} \left[ 1 + \frac{x_i}{\alpha_i\, x_{r+1}} \right]$$

le point $E_1$ sera

$$E_1 = \begin{pmatrix} x_1 + (1-\theta_1)\alpha_1\ x_{r+1} \\ \vdots \\ x_r + (1-\theta_1)\alpha_r\ x_{r+1} \\ \theta_1\ x_{r+1} \\ x_{r+2} \\ \vdots \\ x_q \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Le point $E_1$ a au moins une composante nulle parmi ses $r$ premières composantes. Par ailleurs, toutes les autres sont positives. Enfin, l'équation (4) assure que $E_1$ est une solution réalisable du P.L. Donc, $E_1$ est une solution réalisable ayant au plus $q-1$ composantes non nulles.

Cherchons maintenant $E_2$ Puisque l'on doit avoir :

$$x = \lambda\ E_1 + (1-\lambda)E_2$$

on en déduit :

$$E_2 = \frac{1}{1-\lambda}(x - \lambda\ E_1)$$

On vérifie aisément que si $x$ et $E_1$ sont deux solutions réalisables, $E_2$ l'est également. En effet :

$$AE_2 = \frac{1}{1-\lambda}\ (Ax - \lambda A\ E_1)$$

$$AE_2 = \frac{1}{1-\lambda}\ (b - \lambda\ b) = b$$

Les coordonnées de $E_2$ sont :

$$E_2 = \begin{pmatrix} x_1 - \dfrac{\lambda}{1-\lambda}(1-\theta_1)\,\alpha_1\, x_{r+1} \\ \vdots \\ x_r - \dfrac{\lambda}{1-\lambda}(1-\theta_1)\,\alpha_r\, x_{r+1} \\ x_{r+1}\left(\dfrac{1-\theta_1\,\lambda}{1-\lambda}\right) \\ x_{r+2} \\ \vdots \\ x_q \\ 0 \\ 0 \end{pmatrix}$$

Pour $\lambda = 0$, on retrouve bien $E_2 = 0$

Faisons croître $\lambda$ de 0 à 1. On voit qu'il existe une valeur $\lambda_1$ *de* $\lambda$ qui annule au moins une coordonnée de $E_2$ en laissant les autres positives, sinon les $r+1$ premières coordonnées deviendraient infinies lorsque $\lambda$ tend vers 1, ce qui est incompatible avec l'hypothèse faite, à savoir que le domaine $D$ des solutions réalisables est borné.

Pour cette valeur $\lambda_1, E_2$ solution réalisable possède au plus $q-1$ composantes non nulles. On a bien mis $x$ sous la forme :

$$x = \lambda_1\, E_1 + (1-\lambda_1)\, E_2$$

avec $0 \le \lambda \le 1$ et $E_1$ et $E_2$ t appartenant à $D$ et ayant au plus $q-1$ composantes non nulles.

À présent, si $E_1$ n'est pas un sommet, on le décompose de la même façon en

$$E_1 = \lambda_2\, E_3 + (1-\lambda_2)\, E_4$$

$E_3 et\ E_4$ appartenant à $D$ et ayant au plus $q-2$ composantes non nulles. On a alors :

$$x = \lambda_1\, \lambda_2\, E_3 + \lambda_1 (1-\lambda_2)\, E_4 + (1-\lambda_1)\, E_2$$

et comme

$$\lambda_1\, \lambda_2 + \lambda_1 (1-\lambda_2) + 1 - \lambda_1 = 1$$

$x$ est combinaison linéaire convexe de $E_3, E_4, E_2$ solutions réalisables.

On poursuit le processus ainsi entamé, en décomposant tout point $E_i$ obtenu qui n'est pas un sommet. À chaque étape de la décomposition, $x$ s'exprime sous la forme d'une combinaison linéaire convexe des $E_i$. Cela se démontre facilement par récurrence.

À chaque décomposition, on augmente le nombre de composantes nulles des solutions $E_i$, on diminue le nombre de leurs composantes non nulles, donc le nombre de vecteurs $P_j$ intervenant dans la solution $E_i$ par :

$$x_1 P_1 + x_2 P_2 \cdots + x_k P_k = b$$

$$x_1, x_2 \cdots x_k > 0$$

En conséquence, on aboutit obligatoirement à des systèmes de vecteurs $P_j$ indépendants pour chaque $E_i$, et donc à des $E_i$ qui ne soient que des points extrêmes, ou des solutions de base. C'est ainsi que l'on peut démontrer le théorème IV.

# Chapitre 2 • Algorithme du simplexe

Nous avons présenté au chapitre précédent les grands principes de l'algorithme du simplexe qui permet de résoudre un programme linéairement mis sous forme standard.

Nous allons à présent appliquer ces principes sur des exemples pour dégager les règles pratiques de l'algorithme. Comme premier exemple, nous allons prendre le cas de l'entreprise fabriquant des camions qui nous avait servi d'introduction au chapitre précédent.

## 2.1. UN EXEMPLE SIMPLE

Le programme linéaire que nous avions étudié géométriquement au paragraphe I du premier chapitre était le suivant :

$$x_1 + 3x_2 \leq 450$$

$$2x_1 + x_2 \leq 350 \qquad (1)$$

$$x_1 + x_2 \leq 200$$

$$x_1, x_2 \geq 0$$

$$Max\ Z = x_1 + 2x_2$$

Nous allons d'abord mettre ce programme sous forme standard, en ajoutant les variables d'écart $x_3, x_4, x_5$.

On obtient :

$$\begin{aligned} x_1 + 3x_2 + x_3 &= 450 \\ 2x_1 + x_2 + x_4 &= 350 \qquad (2) \\ x_1 + x_2 + x_5 &= 200 \\ x_1,\ x_2, x_3, x_4, x_5 &\geq 0 \end{aligned}$$

$$Max\ Z = x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5$$

Nous pouvons à présent appliquer les règles de l'algorithme du simplexe :

**1) Trouver une solution de base.**

Ici $m = 3\ et\ n = 2$; il s'agit donc de trouver une solution réalisable, c'est-à-dire respectant les contraintes (2) telle que 3 variables parmi $x_1, x_2, x_3, x_4\ et\ x_5$ soient non nulles et 2 variables nulles.

Or, ici, une telle solution est absolument évidente.

Elle est obtenue en annulant les variables principales et en égalisant variables d'écart et second membre des équations de (2)

$$x_1 = 0$$
$$x_2 = 0$$
$$x_3 = 450$$
$$x_4 = 350$$
$$x_5 = 200$$
$$Z = 0$$

Dans l'espace des variables principales, c'est-à-dire $R^2$, cette solution de base correspond à l'origine des axes 0.

Elle est évidemment totalement inintéressante au niveau de la fonction économique $(Z = 0)$.

C'est pourquoi, comme l'indique la méthode générale de l'algorithme du simplexe, nous allons essayer de passer à une autre solution de base, en augmentant la fonction économique.

**2) Exprimer les variables de base et la fonction économique en fonction des variables hors base**. C'est-à-dire, en reprenant les notations du chapitre précédent, transformer les équations de (2) en :

$$x_i + \sum_{j \in \bar{I}} t_{ij} x_j = t_i \qquad \forall i \in I$$

et

$$z - z_o = \sum_{j \in \bar{I}} (c_j - z_j) x_j \qquad (3)$$

Là encore, cette opération est ici particulièrement facile, puisque les équations de (2) sont déjà sous cette forme. En effet $x_3, x_4, x_5$ sont les variables de base et $x_1, x_2$ les variables hors base. On a bien :

$$\begin{aligned} x_3 + x_1 + 3x_2 &= 450 \\ x_4 + 2x_1 + x_2 &= 350 \\ x_5 + x_1 + x_2 &= 200 \end{aligned} \qquad (4)$$

Quant à $Z$, elle est elle aussi directement exprimée en fonction des variables principales, donc hors base, c'est-à-dire $x_1 \; et \; x_2$.

$$Z = x_1 + 2x_2$$

**3) Choisir une variable hors base qui doit entrer dans la base.**

Comme les coefficients de $x_1 et\ x_2$ dans $Z$ sont tous les deux positifs, on peut choisir l'une ou l'autre pour la faire entrer dans la base : l'opération sera toujours intéressante. Cela dit, le coefficient de $x_2(+2)$ est plus grand que le coefficient de $x_1(+1)$. On peut donc penser qu'il est plus intéressant de faire entrer $x_2$ dans la base. Nous choisirons donc $x_2$.

Ce critère, qui consiste à choisir pour entrer dans la base la variable $x_j$ celle pour laquelle $c_j - z_j > 0$ est le plus grand possible, s'appelle le **premier critère de Dantzig** (mathématicien auquel nous devons l'algorithme du simplexe). Il faut remarquer que ce critère n'a rien d'absolu : on n'est pas sûr, en l'appliquant systématiquement, d'aller au plus vite sur la solution optimale. On améliore les chances d'une convergence rapide, c'est tout.

**4) Faire varier la variable entrant dans la base jusqu'à ce qu'une variable de base s'annule.**

On a vu au chapitre précédent que la valeur de la variable entrante $x_j$ annulant juste une variable de base est, en regard des équations (3)

$$x_j^0 = \min_{i \,|\, t_{ij} > 0} \frac{t_i}{t_{ij}}$$

$j \in \bar{I}$

$i \in I$

On peut retrouver ici ce résultat en utilisant directement les équations (4). En effet, si $x_2$ varie et si $x_1$ reste nul on a :

$$x_3 + 3x_2 = 450$$
$$x_4 + x_2 = 350$$
$$x_5 + x_2 = 200$$

On voit que si $x_2$ varie, il ne peut le faire que jusqu'à $\frac{450}{3} = 150$, valeur qui annule juste $x_3$. Pour cette valeur, $x_4$est égal à 200, $x_5$ à 50.

On a ainsi retrouvé (5). En effet, ici,

$t_3 = 450, t_4 = 350, t_5 = 200\ et\ t_{32} = 3, t_{42} = 1, t_{52} = 1$

Nous sommes donc passés d'une solution de base à une nouvelle solution de base, qui est:

$x_2 = 150, x_4 = 200, x_5 = 50$ (3 variables de base non nulles)

$x_1 = 0, x_3 = 0$ (2 variables hors base nulles).

Lorsque l'on a choisi la variable hors base entrant dans la base $x_j$ la variable sortant de la base est donc déterminée par l'indice $i_0$ tel que:

$\frac{t_{i0}}{t_{i0j}} \leq \frac{t_i}{t_{ij}}$ pour $i \in I$ tel que $t_{ij} > 0$ (6)

Cette condition, qui permet de trouver $i_0$ est appelée **second critère de Dantzig**. Ce critère, contrairement au premier, est absolu : il détermine la variable sortante sans ambiguïté, sauf si plusieurs indices $i_0$ obéissant à (6), cas que l'on étudiera plus loin.

Le passage que nous avons effectué d'une base($I = \{3,4,5\}$) à une autre ($I = \{2,4,5\}$) a été particulièrement simple du fait que les équations du P.L étaient sous la forme (3) ci-dessus (chaque variable de base et $Z$ exprimées en fonction des variables hors base). En particulier, pour savoir quelle variable il serait intéressant par la suite de faire entrer dans la base, il faudrait que la fonction économique soit exprimée uniquement en fonction des nouvelles variables hors base $(x_1, x_2)$.

Nous allons donc remettre le programme sous la forme (3), avec la nouvelle définition de $I$, donc de $\overline{I}$. Théoriquement, rien de difficile à cela : il s'agit d'inversion et de multiplication de matrices (voir les formules (5) et (6) du paragraphe II du Chapitre I).

En fait, l'opération est encore plus facile que cela et peut se faire uniquement à partir de combinaisons d'équations du programme linéaire, comme nous allons le voir.

Pour faciliter la compréhension, nous utiliserons la présentation habituellement en vigueur, dite méthode des tableaux (due à Charnes, Cooper et Hendersen).

**5) Remise en forme du programme linéaire.**

Reprenons notre exemple, c'est-à-dire le programme linéaire (2) et mettons-le sous la forme suivante :

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| 3 | 1 | (3) | 1 | 0 | 0 | 450 |
| 4 | 2 | 1 | 0 | 1 | 0 | 350 |
| 5 | 1 | 1 | 0 | 0 | 1 | 200 |
| | 1 | 2 | 0 | 0 | 0 | z |

Le grand tableau représente la matrice des contraintes$(m, m+n)$, si l'on prend les notations précédentes (à chaque variable est associée une colonne); la colonne de droite représente le second membre des contraintes.

La ligne du dessous donne les coefficients de la fonction économique.

Quant au tableau de gauche, il donne le numéro de la variable de base dont la valeur est égale, pour la solution de base considérée, à l'élément du second membre figurant sur la même ligne (par exemple sur la deuxième ligne ($x_4 = 350$).

Il ne faut jamais oublier que ce tableau ne fait que représenter les contraintes et la fonction économique, et qu'il s'agit donc d'équations résumées d'une certaine façon.

Cela dit, l'opération (3) a consisté à choisir la variable $x_2$ comme candidate à la base (flèche sous les tableaux).

L'opération 4) a, elle, abouti à extraire de la base de la variable $x_3$ : si l'on fait les rapports des éléments du second membre aux éléments des mêmes lignes de la colonne de la variable $x_2$ (variable entrante), le plus petit de ces rapports est trouvé en effet pour la première ligne (ligne de la variable de la base $x_3$).

Nous entourons l'élément du tableau qui est l'intersection de la colonne de la variable entrante et de la ligne de la variable sortante : ce sera le pivot.

Pour écrire le P.L sous la forme (3) avec la nouvelle base $I = \{2, 4, 5\}$, nous allons procéder de la façon suivante : puisque $x_4\ et\ x_5$ restent dans la base, nous allons décider que, dans le tableau des contraintes, les colonnes relatives à ces deux variables restent identiques, à savoir :

Pour $x_4 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ et pour $x_5 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

Puisque $x_2$ est dans la base, il faudrait que la colonne correspondant à $x_2$, soit maintenant : $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

En effet, s'il en est ainsi, la matrice relative aux trois variables $x_2, x_4, x_5$ est :

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Ainsi, puisque les tableaux ne sont rien d'autre que les contraintes du P.L, les variables de base sont alors directement exprimées en fonction des variables hors base. Nous allons essayer d'obtenir cette nouvelle colonne pour $x_2$ et cela en procédant à des combinaisons des lignes de nos tableaux, ce qui est possible étant donné que ces lignes, rappelons-le, ne font que représenter des égalités.

Nous voulons avoir

| | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|
| 2 | | | 1 | | 0 | 0 | | |
| 4 | | | 0 | | 1 | 0 | | |
| 5 | | | 0 | | 0 | 1 | | |
| | | | | | | | | |
| | | | | | | | | |

(Dans le tableau de gauche 2 a remplacé 3, puisque $x_2$ est variable de base et que $x_3$ est sorti de la base).

Pour obtenir cela, il faut :

a) diviser la première ligne du tableau des contraintes par 3 (division de la ligne du pivot par le pivot), pour obtenir 1 à la place du pivot :

| | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|
| 2 | | 1/3 | 1 | 1/3 | 0 | 0 | | 150 |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

b) combiner la ligne 1 et la ligne 2 de l'ancien tableau pour que le 1 qui se trouve à l'intersection de la colonne 2 et de la ligne 2 se transforme en 0.

Il faut alors faire : ligne $2 - \frac{1}{3}$ ligne 1; en effet, si l'on faisait $\frac{1}{3}$ ligne 1 - ligne 2 pour placer le résultat en ligne 2, le 1 de la 2ème ligne et 4$^{ème}$ colonne se changerait en - 1, ce qu'on ne veut pas, puisque $x_4$ reste de base. Par ailleurs, dans ce cas, on voit facilement que le second membre deviendrait négatif.

En faisant au contraire ligne $2 - \frac{1}{3}$ ligne 1, on est sûr que le 1 associé à $x_4$ ne changera pas, puisque $x_4$ étant variable de base, la colonne 4 a des zéros partout sur toutes les autres lignes que la ligne 2.

| | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|
| 2 | | 1/3 | 1 | 1/3 | 0 | 0 | | 150 |
| 4 | | 5/3 | 0 | -1/3 | 1 | 0 | | 200 |
| 5 | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

On voit par ailleurs que la combinaison de lignes effectuée de la sorte laisse le second membre positif.

On a en effet obtenu à la ligne 2 un nombre positif $\left(200 = \frac{350}{1} - \frac{450}{3}\right)$. Cela provient de l'application du second critère de Dantzig, déterminant la ligne du pivot comme étant celle pour laquelle $\frac{t_i}{t_{ij}}\left(t_{ij} > 0\right)$ est le plus petit possible.

c) combiner la ligne 1 et la ligne 3 de l'ancien tableau : on fait là aussi ligne $3 - \frac{1}{3}$ ligne 1 pour faire disparaître le 1 de la ligne 3 et de la colonne 2.

On obtient :

| | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|
| 2 | | 1/3 | 1 | 1/3 | 0 | 0 | | 150 |
| 4 | | 5/3 | 0 | -1/3 | 1 | 0 | | 200 |
| 5 | | 2/3 | 0 | -1/3 | 0 | 1 | | 50 |
| | | | | | | | | |

En lisant ce tableau, on a l'expression des nouvelles variables de base en fonction des nouvelles variables hors base

$$x_2 = 150 - \frac{1}{3}x_1 - \frac{1}{3}x_3$$

$$x_4 = 200 - \frac{5}{3}x_1 + \frac{1}{3}x_3$$

$$x_5 = 50 - \frac{2}{3}x_1 + \frac{1}{3}x_3$$

On a également les valeurs des variables de base pour la solution de base considérée :
$x_2 = 150 \qquad x_4 = 200 \qquad x_5 = 50$

d) il manque l'expression de la fonction économique en fonction des variables hors base $x_1$ et $x_3$. Pour l'avoir, on peut appliquer deux méthodes :

- soit utiliser la formule générale des équations (2)

$$Z - Z_o = \sum_{j \in \bar{I}} (c_j - z_j)\, x_j$$

avec

$$z_j = \sum_{i \in I} t_{ij}\, c_i$$

(voir chapitre 1)

Pratiquement, on calcule pour le tableau précédent (correspondant à la nouvelle base ($I = 2, 4, 5$)) les $z_j$ : pour cela, dans le tableau de gauche, on inscrit à côté des numéros des variables de base leurs coefficients dans la fonction économique. Un $z_j$ est obtenu en effectuant les produits des éléments de la colonne de la variable $x_j$ dans la matrice des contraintes ($t_{ij}$) par les coefficients $c_j$ appartenant à la même ligne, puis en additionnant ces produits. Par exemple pour $z_1$ :

| $c_i$ ↓ | | | $t_{i1}$ ↘ 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | | 1/3 | | | | | | |
| 0 | 4 | | 5/3 | | | | | | |
| 0 | 5 | | 2/3 | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

$$z_1 = t_{21}\, c_2 + t_{41}\, c_4 + t_{51}\, c_5$$

$$= 2 \times \frac{1}{3} + \frac{5}{3} \times 0 + \frac{2}{3} \times 0 = \frac{2}{3}$$

On a ensuite les $\delta_1 = c_1 - z_1$(ou gains marginaux).

Par exemple

$$\delta_1 = c_1 - z_1 = 1 - \frac{2}{3} = \frac{1}{3} \geq 0$$

Quant à $z_0$, c'est la valeur de la fonction économique pour la solution de base considérée, soit ici $Z_0 = 2 \times 150 = 300$.

- soit utiliser encore des combinaisons de lignes :

Reprenons en effet le tableau A (solution de base de départ). Nous savons que ce tableau va être transformé en un tableau correspondant à la solution de base $I = 2,4,5$ et où la fonction économique devra être exprimée uniquement en fonction des variables hors base $x_1$ et $x_3$.

Combinons alors la ligne de la fonction économique avec celle du pivot de façon à éliminer le coefficient de $x_2$ dans la fonction qui est égal à 2 ; $x_4$ et $x_5$ étant de base, il leur correspond deux zéros dans la ligne de la fonction économique : cette combinaison laissera ces zéros intacts, ce qui est le but recherché, puisque ces deux variables restent de base. On voit qu'il suffit de faire :

ligne de fonction économique $-\frac{2}{3}$ ligne du pivot

pour obtenir le tableau complet B

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | | |
| 2 | 2 | | 1/3 | 1 | 1/3 | 0 | 0 | | 150 |
| | 4 | | 5/3 | 0 | -1/3 | 1 | 0 | | 200 |
| | 5 | | 2/3 | 0 | 1/3 | 0 | 1 | | 50 |
| | | | | | | | | | |
| | | | 1/3 | 0 | -2/3 | 0 | 0 | | z-300 |

**Remarque** : le calcul du gain marginal de $x_3$ par les formules (7) donne :

$$z_3 = 2\times\frac{1}{3}+\frac{1}{3}\times 0+\frac{1}{3}\times 0 = \frac{2}{3}$$

d'où

$$\delta_3 = c_3 - z_3 = 0-\frac{2}{3} = -\frac{2}{3}$$

On obtient bien le même résultat.

Ce qu'il faut bien voir c'est que le nouveau tableau obtenu a deux significations :

1- C'est une simple combinaison des lignes du tableau initial, représentant les équations liant entre elles les variables $x_1, x_2, x_3, x_4, x_5, z$. Il représente donc toujours le même système d'équations, sous une forme équivalente. La lecture du nouveau tableau donne immédiatement les équations (3), écrites pour la nouvelle base $I = (2,4,5)$ et en particulier les éléments $t_{ij}$ de la matrice simpliciale $T$ telle que : $x_1 + Tx_{\bar{I}} = t$

ici $T = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{5}{3} & \frac{-1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{pmatrix}$

2- Il donne une solution particulière de ce système d'équations : une solution de base. Cette solution apparaît à l'évidence : il suffit d'annuler les variables hors base : on a immédiatement :

$x_2 = 150$

$x_4 = 200$

$x_5 = 50$

$Z = 300$

On a effectué ainsi la première itération de la résolution du programme linéaire, permettant de passer d'une solution de base initiale inintéressante à une autre solution de base plus intéressante ($z = 300$).

Si l'on se réfère à l'illustration géométrique du chapitre I (figure 1), on est passé du sommet 0 au sommet A du polygone des solutions réalisables.

Il faut examiner maintenant s'il est encore possible de changer de sommet en améliorant la fonction économique. Or, on constate sur le tableau B que le coefficient dans $Z$ de $x_1$ est encore positif $+\frac{1}{3}$ Nous allons donc faire entrer cette variable dans la base, et recommencer les opérations ci-dessus.

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | | 1/3 | 1 | 1/3 | 0 | 0 | | 150 |
| | 4 | | 5/3 | 0 | -1/3 | 1 | 0 | | 200 |
| | 5 | | (2/3) | 0 | 1/3 | 0 | 1 | | 50 |
| | | | | | | | | | |
| | | | 1/3 | 0 | -2/3 | 0 | 0 | | z-300 |

Le pivot est obtenu en faisant les rapports des éléments du second nombre aux éléments de la même ligne de la colonne 1 (colonne de la variable entrante), et en prenant le minimum, soit :

$$\frac{t_2}{t_{21}} = \frac{150}{\frac{1}{3}} = 450$$

$$\frac{t_4}{t_{41}} = \frac{200}{\frac{5}{3}} = 120$$

$$\frac{t_5}{t_{51}} = \frac{50}{\frac{2}{3}} = 75$$

Le pivot est donc $t_{51}$ entouré sur le tableau.

La variable sortant de la base est par conséquent $x_5$.

On combine maintenant les lignes du tableau de façon à avoir pour colonne de la variable $x_5$ de la colonne :

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

On fait donc :

$$ligne3 = ligne3 \times \frac{3}{2}$$

$$ligne2 = ligne2 - \frac{5}{2} ligne3$$

$$ligne1 = ligne1 - \frac{1}{2} ligne3$$

On obtient :

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | | 0 | 1 | 1/2 | 0 | -1/2 | | 125 |
| 0 | 4 | | 0 | 0 | 1/2 | 1 | -5/2 | | 75 |
| 1 | 1 | | 1 | 0 | -1/2 | 0 | 3/2 | | 75 |
| | | | | | | | | | |
| | | | | | | | | | |

Pour obtenir les nouveaux gains marginaux des variables hors base$(x_3, x_5)$, on fait :

$$z_3 = 2 \times \frac{1}{2} - \frac{1}{2} \times 1 = \frac{1}{2}$$

$$\delta_3 = c_3 - z_3 = -\frac{1}{2}$$

$$z_5 = -\frac{1}{2} \times 2 + \frac{3}{2} = +\frac{1}{2}$$

$$\delta_5 = c_5 - z_5 = -\frac{1}{2}$$

On a donc le tableau C :

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 2 | | 0 | 1 | 1/2 | 0 | -1/2 | | 125 |
| | 4 | | 0 | 0 | 1/2 | 1 | -5/2 | | 75 |
| | 1 | | 1 | 0 | -1/2 | 0 | 3/2 | | 75 |
| | | | | | | | | | |
| | | | 0 | 0 | -1/2 | 0 | -1/2 | | z-325 |

Tous les gains marginaux des variables hors base sont négatifs : on a donc atteint l'optimum qui est :

$x_1 = 75$

$x_2 = 125$

$x_3 = 0$

$x_4 = 75$

$x_5 = 0$

$Z = 325$

Sur la figure 1 du chapitre I, cette solution correspond au sommet B, et l'on retrouve bien la solution optimale que le raisonnement géométrique nous avait fournie.

## 2.2. CAS GENERAL. REGLES DU PIVOTAGE

Nous venons de voir sur un exemple simple comment fonctionne l'algorithme du simplexe et comment on peut, en raisonnant uniquement sur ces combinaisons d'équations, passer d'un sommet du polyèdre des solutions réalisables à un autre en améliorant chaque fois la fonction économique.

Nous allons maintenant examiner ce que ces opérations impliquent comme calculs systématiques dans le cas général. Nous allons supposer que, pour un programme linéaire quelconque, nous ayons obtenu une solution de base et que nous sachions obtenir les formules (3), c'est-à-dire (avec les notations précédentes) :

$$x_i + \sum_{j \in \bar{I}} t_{ij} x_j = t_i \ \forall i \in I$$

$$Z - Z_0 = \sum_{j \in \bar{I}} \delta_j x_j$$

Avec

$\delta_j = c_j - z_j \qquad z_j = \sum_{j \in I} t_{ij} c_j \quad et \ Z_0 = \sum_{j \in I} c_i t_j$

Si l'on résume le programme en tableaux, comme on l'a fait précédemment, on a la disposition suivante

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | j | | i | | |
| | | | | | | | | | 125 |
| $c_i$ | i | | | | $t_{ij}$ | | +1 | | $t_i$ |
| | | | | | | | | | 75 |
| | | | | | | | | | |
| | | | | | $\delta_j$ | | 0 | | z-$z_0$ |

avec $i \in I \ et \ j \in I$

Cherchons à améliorer la solution obtenue en passant à une autre solution de base. Pour cela, décrivons les mêmes phases de calcul que pour le petit exemple que nous venons de traiter :

1) on regarde si tous les $\delta_j$ (gains marginaux) sont négatifs ou nuls. Si oui, on est à l'optimum et on arrête le calcul. Sinon :

2) on choisit $j_0 \in \bar{I}$ tel que $\delta_{j0} > 0$, $x_{j0}$ est la variable entrant dans la base. En général, on choisit $x_{j0}$ de telle façon que $\delta_{j0} = max \delta_j$ (premier critère de Dantzig).

3) on détermine $j_0 \in I$ tel que

$\frac{t_{i0}}{t_{i0j0}} \leq \frac{t_i}{t_{ij0}} \ \forall i \in I \ \ t_{ij0} > 0$

le $i_0$ trouvé est tel que $x_{i0}$ est la variable sortant de la base (second critère de Dantzig).

4) on transforme le tableau central de façon que pour la colonne correspondant à $x_{j0}$ :

$t_{i0j0}$ soit remplacé par 1

$t_{i0j0} \ avec \ i \neq i_0$ soit remplacé par 0.

Pour cela :

a) on divise la ligne du pivot par $t_{i0j0}$.

Donc

$t_{i0j0} \rightarrow 1$

$t_{i0j} \rightarrow \frac{t_{i0j}}{t_{i0j0}}\ si\ j \neq j_0$

$t_{i0} \rightarrow \frac{t_{i0}}{t_{i0j0}}$

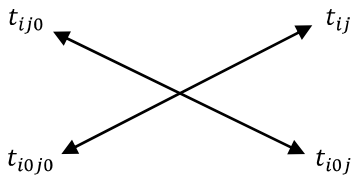
b) on prend une autre ligne que celle du pivot ($i \neq i_0$), et puisque l'on veut $t_{ij0=0}$, on fait :

$ligne\ i - \frac{t_{ij0}}{t_{i0j0}} \times ligne\ i_0$

on a alors les transformations : $t_{ij} \rightarrow t_{ij} - \frac{t_{ij0}}{t_{i0j0}} t_{i0j}$ $\quad$ $t_i \rightarrow t_i - \frac{t_{ij0}}{t_{i0j0}} t_{i0}$

on peut ainsi écrire : $t_{ij \rightarrow \frac{t_{ij}t_{i0j0} - t_{ij0}t_{i0j}}{t_{i0j0}}}$ $\quad$ $t_{ij \rightarrow \frac{t_i t_{i0j0} - t_{ij0}t_{i0}}{t_{i0j0}}}$

formules dites du « rectangle », en effet, on a le schéma ci-dessus :




Pour avoir le nouveau $t_{ij}$, on fait les produits des deux termes sur une même diagonale ($t_{ij} \times t_{i0j0}$ et $t_{ij0} \times t_{i0j}$) et on divise la différence de ces produits par le pivot.

5) on transforme la ligne de la fonction économique, pour obtenir $\delta_{j0} = 0$. Il suffit de faire :

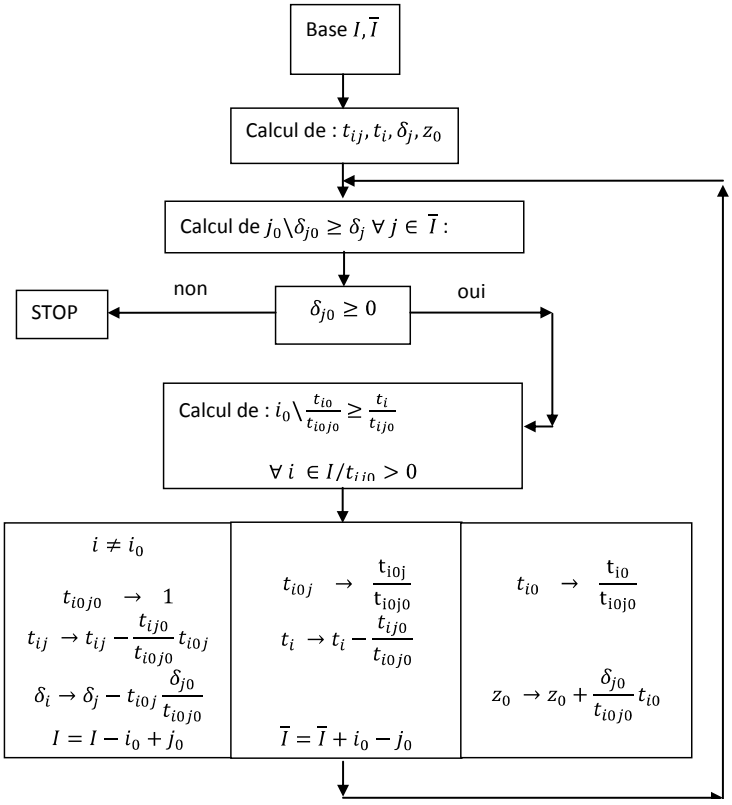
$ligne\ de\ Z - \frac{\delta_{j0}}{t_{i0j0}}\ ligne\ i_0$

d'où $\delta_j \rightarrow \delta_j - \frac{t_{i0j}}{t_{i0j0}} \delta_{j0}$ et $Z_0 \rightarrow Z_0 + \frac{\delta_{j0}}{t_{i0j0}} t_{i0}$

On constate alors que la formule du « rectangle » est valable qu'il s'agisse de calculer le nouveau second membre, la nouvelle matrice des contraintes, ou les nouveaux gains marginaux, pour toutes les lignes autres que celle du pivot.

La nouvelle base étant trouvée, ainsi que les tableaux correspondants, on revient en 1).

L'ordinogramme de la méthode est le suivant :




En fait, on voit que l'algorithme fonctionne à partir du moment où l'on a trouvé une base et où l'on a pu mettre les contraintes du programme sous la forme (3) (variables de base et fonction économique exprimées en fonction des variables hors base). Sur l'exemple que nous avons traité, ce problème avait reçu une solution tout à fait évidente à partir du moment où l'on avait introduit des variables d'écart : on pouvait prendre ces dernières comme variables de base et on avait alors directement les contraintes du P.L sous la forme (3).

Ce n'est pas toujours évident, comme on va le voir à présent

## 2.3. RECHERCHE D'UNE SOLUTION DE BASE INITIALE

Prenons le petit exemple suivant :

$$\begin{aligned} & x_1 + x_2 + x_3 \leq 4 \\ & 2x_1 + x_2 + x_3 \geq 2 \\ & x_1,\ x_2,\ x_3 \geq 0 \end{aligned} \qquad (8)$$

$$Max\ Z = x_1 + x_2 + x_3$$

On aura remarqué que, dans l'algorithme du simplexe, le second membre est toujours positif, puisque c'est le vecteur des valeurs des variables de base pour le sommet considéré.

En conséquence, si on veut transformer les inéquations de ce P.L en équations, on est obligé d'introduire deux variables d'écart $x_4$ et $x_5$ de la façon suivante :

$$\begin{aligned} & x_1 + x_2 + x_3 + x_4 = 4 \\ & 2x_1 + x_2 - x_3 - x_5 = 2 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned} \qquad (9)$$

On remarquera le signe $-$ pour la variable $x_5$, dû à l'inégalité $\geq$.

Une solution de base de ce P.L a 2 variables non nulles et 3 variables nulles.

Comme les variables de base doivent être positives, on ne peut plus prendre pour ces variables les variables d'écart (ce qui ferait $x_4 = 4$, $x_5 = -2$).

On ne dispose pas d'une solution de base de départ immédiate.

Pour ce problème, nous allons ajouter une nouvelle variable $x_6$ dans la seconde contrainte, variable dite artificielle, *a priori* redondante par rapport à $x_5$.

$$\begin{aligned} & x_1 + x_2 + x_3 + x_4 = 4 \\ & 2x_1 + x_2 - x_3 - x_5 + x_6 = 2 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned} \qquad (10)$$

On voit qu'ainsi on a une solution de base évidente

$$x_1, x_2, x_3, x_5 = 0 \qquad x_4 = 4 \qquad x_6 = 2$$

et que par ailleurs, les variables de base sont bien exprimées en fonction des variables hors base.

Cependant, il convient de noter que la solution trouvée n'appartient pas au polyèdre des solutions réalisables : en effet, la deuxième contrainte du P.L initial n'est évidemment pas respectée avec $x_1 = x_2 = x_3 = 0$

Il convient alors d'éliminer au plus vite $x_6$ de la base, et donc de l'annuler; si $x_6$ est nul, en effet, la solution de base trouvée avec $x_1, x_{2,} x_3, x_4, x_5 \geq 0$ respectera obligatoirement les contraintes (9).

Pour éliminer $x_6$, on va l'introduire dans la fonction économique avec un poids **négatif très grand.**

$$Z = x_1 + x_2 + x_3 + 0\,x_4 + 0\,x_5 - M\ x_6$$

avec $M > 0$ et très grand .

Ainsi, la variable $x_6$ doit en principe sortir rapidement de la base. Si à la fin de l'algorithme du simplexe, la variable $x_6$ est toujours dans la base, c'est que le système d'inéquations du P.L (8) est incompatible. En effet, s'il était compatible, il y aurait des solutions réalisables finies et donc la valeur optimale de $Z$ serait inférieurement bornée. Or, si $x_6$ est variable de base positive (on exclut le cas des dégénérescences) à l'optimum, on voit que l'on peut faire tendre la valeur optimale de $Z$ vers moins l'infini en augmentant $M$ indéfiniment.

Si le système d'inéquations du P.L est compatible, $x_6$ doit donc sortir de la base à une itération ou une autre du calcul; à partir de ce moment, on élimine purement et simplement $x_5, x_6$ suffisant à transformer la deuxième contrainte en égalité.

Traitons l'exemple :

On a le tableau suivant :

| | | | 1 | 2 | 3 | 4 | 5 | 6 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | | 1 | 1 | 1 | 1 | 0 | 0 | | 4 |
| -M | 6 | | 2 | 1 | -1 | 0 | -1 | +1 | | 2 |
| | | | | | | | | | | |
| | | | 1 | 1 | 1 | 0 | 0 | -M | | Z |

Une difficulté surgit dans la mesure où $Z$ n'est pas exprimée uniquement en fonction des variables hors base ($x_6$ est de base). Il faut alors utiliser les formules

$$Z - Z_o = \sum_{j \in I} (c_j - z_j)\ x_j$$

$$z_j = \sum_{i \in I} c_i\ t_{ij} \qquad Z_o = \sum_{i \in I} c_i\ x_i$$

pour calculer la ligne des gains marginaux.

On obtient :

| | | | 1 | 2 | 3 | 4 | 5 | 6 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | | 1 | 1 | 1 | 1 | 0 | 0 | | 4 |
| -M | 6 | | (2) | 1 | -1 | 0 | -1 | +1 | | 2 |
| | | | | | | | | | | |
| | | | 1+2M | 1+M | 1-M | 0 | -M | 0 | | Z+2M |

Faisons entrer $x_1$ dans la base; les règles de l'algorithme du simplexe donnent :

| | | | 1 | 2 | 3 | 4 | 5 | 6 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | | 1 | 1/2 | 3/2 | 1 | 1/2 | 0 | | 4 |
| -M | 6 | | (2) | 1/2 | -1/2 | 0 | -1/2 | 1/2 | | 2 |
| | | | | | | | | | | |
| | | | 0 | 1/2 | 3/2 | 0 | 1/2 | -M-1/2 | | Z-1 |

On peut alors éliminer $x_6$, dont on n'a plus besoin. Le tableau suivant est alors :

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | | 1 | 1/3 | 1 | 2/3 | 1/3 | | 2 |
| 1 | 1 | | 1 | 2/3 | 0 | 1/3 | -1/3 | | 2 |
| | | | | | | | | | |
| | | | 0 | 0 | 0 | -1 | 0 | | Z-4 |

On atteint une solution optimale

$x_1 = 2,\ x_2 = 0,\ x_3 = 2,\ z = 4$ [3]

**Remarque 1 :** Lorsque l'on est amené à introduire plusieurs variables artificielles, on peut les faire intervenir dans la fonction économique avec le même poids $-M$ (avec $M > 0$ très grand). Là aussi, si à la fin de l'algorithme du simplexe, une ou plusieurs variables artificielles restent dans la base, c'est que le système des contraintes du P.L est incompatible.

**Remarque 2 :** Il se peut également que l'on ait certaines contraintes qui soient des égalités (et non des inégalités $\leq$ ou $\geq$).

---

[3] La présence de gains marginaux pour des variables hors base montre qu'il existe des solutions équivalentes (par exemple $x_1 = 0,\ x_2 = 3,\ x_3 = 1, z = 4$). On a donc affaire à une dégénérescence du second type. C'est-à-dire à une infinité de solutions optimales.

Dans le cas où une solution de base n'est pas évidente, il convient alors, on le sait, de transformer la contrainte

$$a_{ij}\ x_j = b_i \qquad (a)$$

en deux contraintes :

$$\Sigma\, a_{ij}\ x_j \le b_i \qquad (a')$$

$$\Sigma a_{ij}\ x_j \ge\ b_i \qquad (a'')$$

- la contrainte (a') va générer une variable d'écart
- la contrainte (a'') va générer une variable d'écart et une variable artificielle.

## 2.4. CAS DE LA DEGENERESCENCE DU PREMIER TYPE

Prenons l'exemple :

$$x_1 + \frac{1}{2}\, x_2 + x_3 \le 2$$

$$4x_1 - x_2 + 2x_3 \le 4$$

$$x_1, x_2, x_3 \ge 0$$

$$Max\ Z = x_1 + 2x_2 + 2x_3$$

Résolvons cet exemple, en introduisant les variables d'écart $x_4$ et $x_5$. Le premier tableau est :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| 0 | 4 | 1 | 1/2 | 1 | 1 | 0 | 2 |
| 0 | 5 | 4 | -1 | +2 | 0 | 1 | 4 |
| | | 1 | 2 | 2 | 0 | 0 | Z |

Si l'on fait entrer $x_3$ tout en faisant sortir $x_4$ plutôt que $x_5$, on obtient, ce qui est arbitraire puisque :

$$\frac{t_4}{t_{43}} = \frac{t_5}{t_{53}} = 2$$

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | | 1 | 1/2 | 1 | 1 | 0 | | 2 |
| 0 | 5 | | 2 | -2 | 0 | -2 | 1 | | $\emptyset$ ← $\varepsilon$ |
| | | | | | | | | | |
| | | | -1 | +1 | 0 | -2 | 0 | | Z-4 |

On se trouve dans le cas où l'une des variables de base $x_5$ est nulle, dit cas de dégénérescence du premier type. Pour pouvoir continuer l'algorithme du simplexe, il convient de remplacer le zéro du second membre par un nombre $\varepsilon > 0$ très petit, quitte à le supprimer par la suite (ce raffinement est essentiellement destiné au calcul sur ordinateur; il est en effet nécessaire que la machine puisse reconnaître, lors de l'application du second critère de Dantzig, les quantités $^0/_{a_{ij}}$ avec $a_{ij} > 0$ des $^0/_{a_{ij}}$ avec $a_{ij} < 0$).

On obtient le tableau optimal :

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | | 2 | 1 | 2 | 2 | 0 | | 2 |
| 0 | 5 | | 6 | 0 | 4 | 2 | 1 | | $\varepsilon + \emptyset$ |
| | | | | | | | | | |
| | | | -3 | 0 | -2 | -4 | 0 | | Z-8 |

soit, en supprimant le $\varepsilon$, la solution optimale :

$$x_1 = 0 \qquad x_2 = 4 \qquad x_3 = 0 \qquad z = 8$$

Les dégénérescences de ce type sont relativement importantes sur le plan théorique, dans la mesure où elles représentent le seul cas où l'algorithme du simplexe peut ne pas converger. Dans ce cas en effet, on montre qu'il peut y avoir cyclage, c'est-à-dire qu'au cours de l'algorithme on rencontre un sommet déjà rencontré. Il existe des procédures pour éviter ce type de mésaventures, par ailleurs très rares, mais dans le cadre limité de cet exposé, nous n'en parlerons pas.

**Remarque** : On peut à propos des dégénérescences recenser les principaux « accidents » pouvant survenir au cours de l'algorithme du simplexe :

1) Un (ou plusieurs) élément du second membre est nul :

*Dégénérescence du premier type*

Signification géométrique : le sommet correspondant, dans $R^n$, est l'intersection de plus de $n$ hyperplans.

2) Un gain marginal associé à une variable hors base est nul.

*Dégénérescence du second type.*

Interprétation géométrique : l'hyperplan de la fonction géométrique est parallèle à une arête ou une face du polyèdre des solutions réalisables.

Ces deux types de dégénérescence peuvent se trouver évidemment combinées.

3) Si la variable entrante est $x_{ij}$, tous les $t_{ij}$ correspondants sont négatifs.

Interprétation géométrique : le polyèdre des solutions réalisables n'est pas borné. Solution infinie.

4) Une (ou plusieurs) variable artificielle reste dans la base.

Interprétation géométrique : le polyèdre des solutions réalisables est vide. Pas de solution.

# Chapitre 3 • Dualité

## 3.1. DEFINITION

On a vu que la forme la plus générale d'un programme linéaire était :

$$A\,x \leq b$$
$$x \geq 0 \qquad (I)$$

$Max\ Z = cx$

Avec $A$ matrice $(m, n)$

$x$ matrice $(n, 1)$

$b$ matrice $(m, 1)$

$c$ matrice $(1, n)$

On appellera dual du programme linéaire (I), appelé lui-même primal, le programme suivant :

$$yA \geq c$$
$$y \geq 0 \qquad (II)$$

$Min\ Z' = yb$

Avec $y$ vecteur ligne (1, $m$)

Ainsi, si le programme primal s'écrit :

$$a_{11}\ x_1 + a_{12}\ x_2 + \cdots + a_{1n}\ x_n \leq b_1$$
$$a_{m1}\ x_1 + a_{m2}\ x_2 + \cdots + a_{mn}\ x_n \leq b_m$$
$$x_1, x_2 \cdots x_n \geq 0$$

$Max\ Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$

le dual de programme est :

$$a_{11}\ y_1 + a_{21}\ y_2 + \cdots + a_{m1}\ y_m \geq c_1$$
$$a_{1n}\ y_1 + a_{2n}\ y_2 + \cdots + a_{mn}\ y_m \geq c_n$$
$$y_1, y_2 \cdots y_n \geq 0$$

$Min\ Z' = b_1y_1 + b_2y_2 + \cdots + b_my_m$

Pour passer du primal au dual, on voit que :

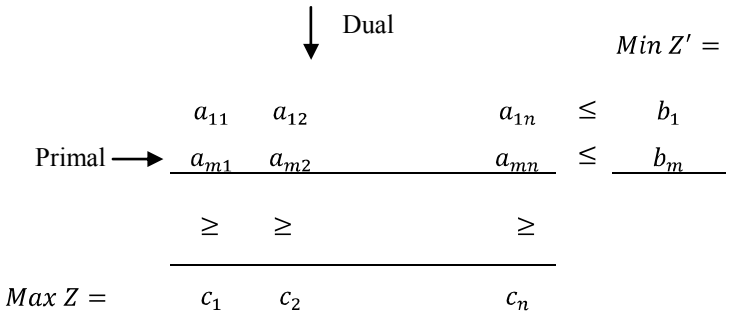
a) les termes du second membre deviennent les coefficients de la fonction économique, et réciproquement ;

b) le problème de maximisation devient un problème de minimisation ;

c) les inégalités ≤ deviennent des inégalités ≥ ;

d) la matrice $A$ se change en sa transposée.

On peut également écrire synthétiquement les deux programmes de la façon suivante :




Le sens des flèches indique « le sens de la lecture » dans les deux cas.

## 3.2. SIGNIFICATION ECONOMIQUE DU PROGRAMME DUAL

Donnons au programme primal (I) une des significations qu'il avait au premier chapitre, à savoir : sous $m$ contraintes du type $\sum a_{ij}\, x_j \leq b_i$ (facteurs de production $i$), trouver les $n$ niveaux d'activité $x_j$ de façon que le profit $\sum c_j x_j$ soit maximal.

Supposons alors que le producteur de biens $x_1, \ldots x_n$ ait le choix entre deux solutions : ou bien continuer à produire ces biens ou au contraire ne plus les produire quitte à revendre son stock de facteurs de production en quantité $b_1, b_2 \ldots b_m$ à un éventuel acheteur.

Appelons $y_1, y_2 \ldots y_m$ les prix unitaires d'achat de ces facteurs de production, prix faisant l'objet de négociations entre l'acheteur et le vendeur. L'acheteur dépensera donc $b_1y_1, b_2y_2 \ldots b_my_m$ s'il achète l'ensemble du stock; il essaiera de fixer avec le producteur les prix $y_1, y_2 \ldots y_m$ de telle façon que sa dépense soit la plus faible possible.

De son côté le producteur, lui, peut raisonner de la façon suivante : il vendait les produits $1, \ldots, n$ avec les bénéfices unitaires $c_1 \ldots c_n$ ; dans le marché, il peut s'efforcer de ne pas perdre son bénéfice, et cela sur aucun produit parmi les $n$. Or, étant donné qu'une

production d'une unité d'un bien $j$ quelconque demandait la quantité $a_{ij}$ en facteur de production $i$, il sera tenté d'affecter au produit $j$ une part de la vente égale à :
$a_{1j}\ y_1 + a_{2j}\ y_2 + \cdots + a_{mj}\ y_m$

et la vente lui sera favorable si
$a_{1j}\ y_1 + a_{2j}\ y_2 + \cdots + a_{mj}\ y_m \geq c_j$

Si producteur et acheteur se fondent sur ces deux exigences, les prix $y_1, y_2 \ldots y_m$ seront trouvés par le programme :
$a_{11}\ y_1 + \cdots + a_{m1}\ y_m \geq c_1$
$a_{1n}\ y_1 + \cdots + a_{mn}\ y_n \geq c_n$
$y_1 \cdots y_m \geq 0$

$Min\ z = b_1 y_1, b_2 y_2 \ldots b_m y_m$

qui est le programme dual (II) du programme (I).

On aurait pu donner d'autres illustrations économiques du dual. Retenons seulement pour le moment que si le programme primal a la signification économique précédente, les variables duales **peuvent être interprétées comme des prix de facteurs de production.**

Ce fait sera analysé de façon plus détaillée dans ce qui va suivre.

## 3.3. CORRESPONDANCE ENTRE VARIABLES DU PRIMAL ET VARIABLES DU DUAL

Des contraintes du programme dual s'écrivent, on l'a vu :

$$\sum_{i=1}^{m} a_{ij} y_i \geq c_j$$

On voit que dans ces contraintes, les coefficients de $y_i$ sont $a_{ij} (j = 1,2, \ldots n)$, c'est-à-dire les coefficients de la *i*ème contrainte du primal. En conséquence : **à une inéquation du primal correspond une variable principale du dual.**

Ajoutons maintenant aux deux programmes (I) et (II) des variables d'écart, en quantité $m$ pour (I) et $n$ pour (II), afin de transformer les inéquations en équations. On se ramène ainsi, pour les deux programmes, à $m + n$ variables.

A une contrainte du primal correspond une variable d'écart du primal; d'après la correspondance ci-dessus, on peut donc dire :

**à une variable d'écart du primal correspond biunivoquement une variable principale du dual.**

Par ailleurs, il est facile de voir que le programme dual du dual est le primal. En effet, soit le programme (II). on peut l'écrire ainsi :

$$A^t\ y^t \geq c^t$$
$$y^t \geq 0$$

$$Min\ z' = b^t y^t$$

en réservant le symbole $t$ aux matrices transposées des matrices figurant dans (II). On peut encore écrire ce programme :

$$-A^t\ y^t \leq -c^t$$
$$y^t \geq 0$$

$$Max\ z'' = -b^t y^t$$

Le dual de ce dernier programme s'écrit alors :

$$-(A^t)^t\ x \geq -(b^t)^t$$
$$x \geq 0$$

$$Min\ z''' = -(c^t)^t x$$

Soit

$$Ax \leq b$$
$$x \geq 0$$

$$Max\ z = cx$$

On retrouve bien le programme (I).

En conséquence, la correspondance ci-dessus marche également dans le sens dual → primal et on peut écrire :

**à une variable d'écart du dual correspond biunivoquement une variable principale du primal**. Ces correspondances sont exprimées dans le tableau suivant :

| **Primal** | **Dual** |
|---|---|
| **Variables principales** | **Variables principales** |
| **Variables d'écart** | **Variables d'écart** |

**Remarque importante :** pour le dual, l'addition de variables d'écart doit se faire de la façon suivante, pour que toutes les variables restent positives :

$$a_{11}\ y_1 + a_{21}\ y_2 + \cdots + a_{ml}\ y_m - y_{m+1} = c_1$$
$$a_{1n}\ y_1 + a_{2n}\ y_2 + \cdots + a_{mn}\ y_m - y_{m+n} = c_n$$
$$y_1 \cdots y_{m'}\ y_{m+1} \cdots y_{m+n} \geq 0$$

## 3.4. THEOREMES SUR LA DUALITE

Nous allons examiner un certain nombre de théorèmes importants mettant en évidence les rapports étroits qui lient un programme linéaire à son dual.

**<u>Théorème I</u> : Si un programme linéaire et son dual admettent tous deux une solution réalisable, chacun d'eux a un optimum fini.**

Soit un programme linéaire et son dual :

| **Primal** | **Dual** |
|---|---|
| (I) | (II) |
| $Ax \leq b$ | $yA \geq c$ |
| $x \geq 0$ | $y \geq 0$ |
| $MaxZ(x) = cx$ | $MinZ'(y) = yb$ |

Soit $x_0$ et $y_0$ deux solutions réalisables respectivement de (I) et (II).

On a : $Ax_0 \leq b$ mais comme $y_0$ est positif, on peut écrire aussi $y_0Ax_0 \leq y_0b$ donc
$y_0Ax_0 \leq Z(y_0)$ (1)

De même, puisque $y_0$ est la solution réalisable de (II), on a :
$y_o A \geq c$

et comme $x_0$ est positif ou nul :
$y_o Ax_o \geq cx_o$

d'où $y_0Ax_0 \leq Z(x_0)$ (2)

Au total, on a l'inégalité très importante :

$Z(x_0) \leq Z(y_0)$ (3)

pour toute solution réalisable $x_0$ de (I) et toute solution réalisable $y_0$ de (II).

En conséquence, s'il existe une solution réalisable $y_0$ de (II), toute solution réalisable de (I), si elle existe, est telle que

$Z(x) \leq Z(y_0)$

La fonction $Z$ est donc bornée sur le domaine des solutions réalisables de (I) et y admet donc un maximum fini. De même, l'existence d'une solution réalisable de (I) implique que $Z$ admet un minimum fini sur le domaine des solutions réalisables de (II).

Si $Z$ max est le maximum de $Z$ et $Z'$ min le minimum de $Z'$, on a

$Z(x) \leq Zmax \leq Z'min \leq Z'(y)$ (4)

pour toute solution $x$ réalisable de (I) et toute solution $y$ réalisable de (II).

**Théorème II : Les valeurs optimales des fonctions économiques d'un programme et de son dual sont égales.**

Après addition des variables d'écart, le programme primal s'écrit :

$A'x' = b$

$x' \geq 0$ (I)

$MaxZ = c'x'$

Avec $A'$ matrice $(m\,, m+n)$

$x'$ matrice $(m+n, 1)$

$b$ matrice $(m, 1)$

$c'$ avec $(1, m+n)$

Soit une base du primal, donnée par un ensemble d'indices I (voir les notations du chapitre précédent).

Effectuons les opérations classiques (voir chapitre précédent); après réordonnancement des variables, (I)' se réécrit sous la forme :

$$\left(A_I, A_{\bar{I}}\right) = \begin{pmatrix} x_I \\ x_{\bar{I}} \end{pmatrix} = b$$

$$Max\ Z = (c_I, c_{\bar{I}}) \begin{pmatrix} x_I \\ x_{\bar{I}} \end{pmatrix}$$

qui permet d'arriver aux équations connues suivantes :

$x_I = [A_I]^{-1} b - [A_I]^{-1} A_I x_I$ (5)

$Z = c_I [A_I]^{-1} b + (c_I - c_I [A_I]^{-1} A_I) x_{\bar{I}}$ (6)

$c_I - c_I [A_I]^{-1} A_I$ est le vecteur des gains marginaux.

Supposons alors que la base $I$ soit la base correspondant à l'optimum du programme (I'). On sait que l'on a dans ce cas :

$c_{\bar{I}} - c_I [A_I]^{-1} A_I \leq 0$ (7)

Par ailleurs, on peut toujours écrire l'expression analogue pour l'ensemble d'indice $I$ ; en effet on a évidemment :

$c_I - c_I [A_I]^{-1} A_I = 0$ (8)

Soit alors une des $n$ variables d'écart du programme primal, $x_{n+r}$, correspondant à la $r^{ième}$ contrainte du même programme.

Cette variable à l'optimum du primal, est, soit dans la base, soit hors base :

a) dans le premier cas, regardons l'équation (8). Supposons que $x_{n+r}$ soit à la p$^{ième}$ place dans $I$. Le $p^{ième}$ élément de $c_I$ est égal à 0, puisque $x_{n+r}$ est variable d'écart.

Le $p^{\text{ième}}$ vecteur colonne de $A_I$ s'écrit :

$$\delta(r) = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 1 \\ 0 \end{pmatrix}$$

(zéro partout sauf à +1sur la $r^{ième}$ligne), d'après la définition même d'une variable d'écart pour un programme du type $Ax \leq b$.

Le premier membre de (8) est un vecteur ligne $(1, m)$. Le $p^{\text{ième}}$ élément de ce vecteur est:

$0 - C_I[A_I]^{-1}\delta(r) = 0$

Posons alors $B = [A_I]^{-1}$ et appelons $B^r$ la $r^{ième}$ colonne de $B$.

On a $[A_I]^{-1}\delta(r) = B^r$

En conséquence, dans ce cas, on a $-c_I B^r = 0$

b) Dans le second cas, on fait le même raisonnement, mais cette fois en prenant l'équation (7). On trouve alors : $-c_I B^r \leq 0$.

Comme on peut faire ce raisonnement pour toute variable d'écart du programme (I), et comme à chaque ligne du programme correspond une variable d'écart, on peut donc écrire, à l'optimum (I)' : $-c_I B \leq 0$

ou $c_I[A_I]^{-1} \geq 0$ (9)

soit alors le vecteur ligne $(1, m)$

$y_0 c_I[A_I]^{-1}$

D'après ce que l'on vient de voir $y_0 \geq 0$.

Par ailleurs, puisque : $A' = (A_I, A_{\bar{I}})$

$y_0 A' = (y_0 A_I, y_0 A_{\bar{I}})$

$= (C_I[A_I]^{-1}A_I, C_I[A_I]^{-1}A_{\bar{I}})$

$= (C_I, C_I[A_I]^{-1}A_{\bar{I}})$

D'après (7), on peut écrire :

$y_0 A' \geq (C_I, C_{\bar{I}})$

$y_0 A' \geq C'$

Si on écrit cette inéquation en ne retenant que les colonnes de $A'$ et les termes de $c'$ correspondant aux variables principales du programme $(I')$, on obtient :

$$y_0 A \geq C$$

Donc, si I est la base correspondant à l'optimum de $(I)$, $y_0 = c_I[A_I]^{-1}$ est solution réalisable de (II).

Par ailleurs, pour la base $I$ optimale de $(I')$, la fonction $Z$ est maximale et égale à $Z_{max} = c_I[A_I]^{-1}b$

Quant à la valeur de la fonction économique du dual, pour la solution réalisable $y_0 = c_I[A_I]^{-1}$, elle est $Z(y_0) = c_I[A_I]^{-1}b$

On a donc :

$Z_{max} = Z'(y_0)$

Le résultat (4) donne aussitôt :

$Z_{max} = Z'_{min} = c_I[A_I]^{-1}b$ (10)

Le théorème (II) est donc démontré : à l'optimum des deux programmes, les valeurs des fonctions économiques sont égales.

Pour passer aux théorèmes suivants, nous allons supposer que, pour le programme dual (II), nous ne minimisons plus $Z' = yb$ mais que nous maximisons $-Z' = -yb$.

Dans ces conditions, un corollaire du théorème précédent est : la valeur optimale de la fonction économique du programme (I) est égale à l'opposé de la valeur optimale de la fonction économique $-Z' = -yb$ du programme dual (II).

**Théorème III : à l'optimum des deux programmes, la valeur d'une variable principale du programme dual est égale à l'opposé du gain marginal de la variable d'écart qui lui correspond dans le primal.**

En effet, on a vu que, en ce qui concerne les variables principales du programme dual, elles étaient données à l'optimum par $y_0 = c_I[A_I]^{-1}$.

La $r^{ième}$ variable du vecteur $y_0$ est, avec les notations précédentes ($[A_I]^{-1}$=B)

$y'_0 = c_I B^r$.

Or, on a vu, dans le paragraphe précédent, que le gain marginal de la $r^{ième}$ variable d'écart associée à la $r^{ième}$ contrainte du primal était :
$\delta_o^r = -c_I\ b^r$

D'où le résultat.

Comme le dual est le primal, on a le théorème III bis :

**Théorème III bis : À l'optimum des deux programmes, la valeur d'une variable principale du primal est égale à l'opposé du gain marginal de la variable d'écart qui lui est associée dans le dual.**

(la symétrie du théorème III et du théorème III bis est assurée par le fait que l'on maximise $-Z' = -yb$ dans le dual).

De la même façon, nous allons démontrer le théorème suivant :

**<u>Théorème IV</u> : À l'optimum des deux programmes, la valeur d'une variable d'écart du dual est égale à l'opposé du gain marginal de la variable principale qui lui est associée dans le primal.**

En effet, les contraintes du dual s'écrivent :
$$yA \geq c$$

Si on appelle $y_0$ le vecteur ligne $1 \times n$ des variables d'écart du dual, on a :
$$yA - y_o = c$$

À l'optimum du dual, on a vu que le vecteur des variables principales était

$y_0 = c_I[A_I]^{-1}$ si $I$ donne la base optimale du primal. On a donc alors :

$y_0 = c_I[A_I]^{-1}A - c.$

Or, $c_I[A_I]^{-1}A - c$ est un vecteur ligne $1 \times n$ qui est égal à l'opposé du vecteur des gains marginaux des variables principales du programme primal à l'optimum de ce même programme. D'où le théorème IV. Naturellement, on en déduit :

**<u>Théorème IV bis</u> : À l'optimum des deux programmes, la valeur d'une variable d'écart du primal est égale à l'opposé du gain marginal de la variable principale qui lui est associée dans le dual.**

Les théorèmes III, III bis, IV et IV bis aboutissent aux conséquences suivantes : on voit que pour deux variables, l'une dans le primal et l'autre dans le dual, qui se correspondent, valeur à l'optimum et gain marginal à l'optimum s'échangent (au signe près). Donc, cela signifie, si l'on ne tient pas compte de dégénérescences éventuelles, que si une variable du primal est de base à l'optimum, sa variable associée dans le dual est hors base à l'optimum de ce dernier et réciproquement.

Pour être plus clair, unifions les notations, afin que deux variables associées portent le même indice : Si le primal s'écrit, avec les variables d'écart :
$$a_{11}x_1 + \cdots + a_{1n}x_n + x_{n+1} = b_1$$
$$a_{m1}x_1 + \cdots + a_{mn}x_n + \cdots x_{m+n} = b_m$$
$$x_1, x_2 \cdots x_{m+n} \geq 0$$

$$Max\ Z = c_1x_1 + \cdots + c_nx_n$$

écrivons le dual de la façon suivante, en affectant les indices 1, 2 … aux variables d'écart.
$$-y_1 + a_{11}y_{n+1} + \cdots + a_{m1}y_{m+n} = c_1$$
$$-y_n + a_{1n}y_{n+1} + \cdots + a_{mn}y_{m+n} = c_n$$

$y_1 \cdots y_{n+m} \geq 0$

$Max\ Z' = -b_1 y_{n+1} \ldots - b_{m+n}$

On a ainsi la correspondance, puisqu'une variable principale du primal est associée à une variable d'écart du dual :
$x_i \leftrightarrow y_i$

Les résultats que l'on vient d'énoncer peuvent alors s'écrire, si on se place à l'optimum des deux programmes, et si $\delta_i$ et $n_i$ sont les gains marginaux de $x_i$ et de $y_i$.

$x_i\ de\ base\ (\delta_i = 0) \Rightarrow y_i\ hors\ base\ n_i = -x_i$

$x_i\ hors\ de\ base\ (\delta_i < 0) \Rightarrow y_i\ de\ base\ y_i = -\delta_i$ (11)

Ces relations sont appelées relations d'exclusion.

Terminons ce paragraphe en donnant la correspondance complète entre les deux tableaux optimaux du simplexe, l'un pour le primal, l'autre pour le dual.

À l'optimum du primal, on a, on le sait, si *I* est l'ensemble d'indices de base :

$x_1 + [A_I]^{-1} A_{\bar{I}} x_{\bar{I}} = A_I{}^{-1} b$

Posons comme dans le premier chapitre

$[A_I]^{-1} A_{\bar{I}} = T$ et $[A_I]^{-1} b = t$ on a donc :

$x_I + T x_{\bar{I}} = t$ (12)

La matrice $(J, T)$, $J$ étant la matrice unité, $m \times m$ représente le tableau optimal du simplexe $T$, permettant d'exprimer les variables de base en fonction des variables hors base est appelée matrice simpliciale. Par ailleurs, on a

$Z = c_I [A_I]^{-1} b + (c_{\bar{I}} - c_I [A_I]^{-1} A_{\bar{I}}) x_{\bar{I}}$

Posons $c_I [A_I]^{-1} b = Z_0$ et $c_{\bar{I}} - c_I [A_I]^{-1} A_{\bar{I}}) = -\delta_{\bar{I}}$

avec $\delta_{\bar{I}} \geq 0$, puisque l'on se place à l'optimum. Avec ces notations :

$Z = Z_0 - \delta_{\bar{I}} x_I$ (13)

D'après ce qu'on a vu, à l'optimum des deux programmes une variable de base du primal correspond à une variable hors base du dual. Par conséquent, si *I* est l'ensemble des indices de base dans le primal, avec les conventions ci-dessus, *I* sera l'ensemble des indices hors base dans le dual et $\bar{I}$ l'ensemble des indices de base.

Puisque par ailleurs, lorsque l'on passe au dual, valeur des variables et gains marginaux s'échangent (au signe près), on aura à l'optimum du dual :

$y_{\bar{I}} + T'^{y_1} = \delta_{\bar{I}}^t$ (14)

$Z' = Z_0 - t y_I$ (15)

$T'$ étant une matrice $n \times m$ ; c'est la matrice simpliciale pour le dual.

Le théorème V permet de passer directement de $T$ (matrice simpliciale du primal) à $T'$.

**Théorème V : À l'optimum des deux programmes, la matrice simpliciale du dual est égale à l'opposée de la transposée de la matrice simpliciale du primal.**

Démonstration : les équations (12) et (13) sont équivalentes aux contraintes initiales du programme (I) :

$Ax \leq b$

$Z = cx$

Donc le programme (I) :

$Ax \leq b$

$x \geq 0$

$Max\ Z = cx$

est équivalent au programme :

$x_I + Tx_{\bar{I}} = t$

$x_I, x_{\bar{I}} \geq 0$

$Max\ Z = Z_0 - \delta_{\bar{I}} x_{\bar{I}}$ $(I')$

Soit au programme$(I'')$, en prenant $x_I$ comme variables d'écart :

$Tx_{\bar{I}} \leq t$

$x_{\bar{I}} \geq 0$ $(I'')$

$Max\ Z = Z_0 - \delta_{\bar{I}} x_{\bar{I}}$

La solution de ce programme est évidente : puisque $\delta_{\bar{I}} \geq 0$ et $t \geq 0$, c'est $x_{\bar{I}} = 0$. C'est-à-dire : la solution optimale de $(I'')$ est donnée par la nullité des variables principales (on retrouve bien la solution optimale de $(I)$, puisque les variables principales de $(I'')$ sont les variables hors base à l'optimum de $(I)$).

Le dual de $(I'')$ peut s'écrire, si on définit un vecteur colonne $y(m, 1)$.

$(T)^t y \geq -\delta_{\bar{I}}^t$

$y \geq 0$

$Min\ Z' = t^t y$

soit

$-(T)^t y \leq \delta_{\bar{I}}^t$

$y \geq 0$

$Min\ -Z' = -t^t y$ $(II")$

Or, aux variables principales du primal $(I)''$ $(Indice\ \bar{I})$ correspondent les variables d'écart du dual, et aux variables d'écart du primal correspondent les variables principales du dual (ensemble d'indices $I$).

Le programme $(II'')$, dual de $(I'')$, donc de $(I)$ peut s'écrire, en ajoutant les variables d'écart, et en indiçant convenablement :

$y_{\bar{I}} - (T)^t y_I = \delta_{\bar{I}}^t$

$y_I, y_{\bar{I}} \geq 0$

$Max - Z' = -t^t y_I$

L'optimum de ce programme est donné par $y_I = 0$, puisque $t^t \geq 0$ et $\delta \frac{t}{I} \geq 0$.

On a donc ci-dessus l'écriture à l'optimum du programme dual du programme (I). En rapprochant (16) des équations (14) et (15), on a donc :

$T' = -(T)^t$ (17)

et le théorème est ainsi démontré.

## 3.5. PASSAGE DU PRIMAL AU DUAL SUR UN EXEMPLE

Reprenons le cas de notre usine de camions que nous avons examiné dans les deux chapitres précédents. Le programme s'écrivait :

$x_1 + 3x_2 \leq 450$

$2x_1 + x_2 \leq 350$

$x_1 + x_2 \leq 200$

$x_{1,} x_1 \geq 0$

$Max\ Z = x_1 + 2x_2$

Le tableau simplexe, à l'optimum de ce programme est ($x_3, x_4, x_5$ variables d'écart)

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | 1 | 0 | -1/2 | 0 | +3/2 | | 75 |
| 2 | 2 | | 0 | 1 | +1/2 | 0 | -1/2 | | 125 |
| 0 | 4 | | 0 | 0 | +1/2 | 1 | -5/2 | | 75 |
| | | | | | | | | | |
| | | | 0 | 0 | -1/2 | 0 | -1/2 | | z-325 |

Le programme dual de ce programme s'écrit :

$$y_1 + 2y_2 + y_3 \geq 1$$
$$3y_1 + y_2 + y_3 \geq 2$$
$$y_1, y_2, y_3 \geq 0$$

$$Min\ Z' = 450y_1 + 350y_2 + 200y_3$$

Changeons alors la numérotation des variables du dual, en ajoutant les variables d'écart, numérotées $y_1$ et $y_2$, les variables principales devenant $y_3, y_4, y_5$ , on a :

$$-y_1 + y_3 + 2y_4 + y_5 = 1$$
$$-y_2 + 3y_3 + y_4 + y_5 = 2$$
$$y_1, y_2, y_3, y_4, y_5 \geq 0$$

$$Max\ Z' = -450y_3 - 350y_4 - 200y_5$$

On a ainsi la correspondance facile, d'après l'analyse faite ci-dessus sur les liaisons entre variables du primal et variables du dual :

$$x_i \leftrightarrow y_i \qquad i = 1, 2, 3, 4, 5$$

Par ailleurs, les théorèmes III à IV bis, donnent pour les caractéristiques des variables du dual à l'optimum de ce programme (voir relation 11) :

$x_1$ de base $= 75 \rightarrow y_1$, hors base ; gain marginal $= -75$

$x_2$ de base $= 125 \rightarrow y_2$, hors base ; gain marginal $= -125$

$x_3$ hors base, gain marginal $= -\frac{1}{2} \rightarrow y_3$ de base $= +\frac{1}{2}$

$x_4$ de base $= 75 \rightarrow y_4$, hors base ; gain marginal $= -75$

$x_5$ hors base, gain marginal $= -\frac{1}{2} \rightarrow y_5$ de base $= +\frac{1}{2}$

$Z' = -325$

Le tableau de l'optimum du dual se présente donc de la façon suivante :

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| -450 | 3 | | | | 1 | | 0 | | 1/2 |
| -200 | 5 | | | | 0 | | 1 | | 1/2 |
| | | | | | | | | | |
| | | | -75 | -125 | 0 | -75 | 0 | | Z'+325 |

Pour remplir les cases restantes du tableau central, il suffit de se reporter au théorème V, ou à la relation (17), pour avoir immédiatement le tableau simplexe final :

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| -450 | 3 | | 1/2 | -1/2 | 1 | -1/2 | 0 | | 1/2 |
| -200 | 5 | | -3/2 | +1/2 | 0 | +5/2 | 1 | | 1/2 |
| | | | | | | | | | |
| | | | -75 | -125 | 0 | -75 | 0 | | Z'+325 |

On constate donc qu'il est aisé de passer des résultats d'un programme linéaire à ceux du programme dual (et réciproquement), et une conséquence de ce fait est qu'il convient toujours d'examiner si la résolution du programme dual ne risque pas d'être plus facile que celle du primal. Ce peut être le cas, par exemple, lorsqu'il y a beaucoup d'inéquations dans le primal de la forme

$\sum a_{ij}x_i \geq b_i$ avec $b_i \geq 0$.

En effet, s'il en est ainsi et si l'on résout le primal, on sera amené à ajouter un grand nombre de variables artificielles, d'où un alourdissement des calculs que l'on peut éventuellement éviter en passant au dual.

On verra par ailleurs dans le chapitre suivant quelle peut être au niveau de la technique de résolution des programmes linéaire, l'utilité de la théorie de la dualité.

## 3.6. INTERPRETATION ECONOMIQUE DES VARIABLES DU DUAL

Reprenons l'illustration économique classique des programmes linéaires que nous avons rappelée au paragraphe II de ce chapitre : $\sum a_{ij}x_j \leq b_i$ est une contrainte de limitation de la ressource $i$ s'appliquant sur les $n$ activités $1,2, \ldots n$.

Il s'agit de maximiser le profit total : $Z = \sum a_j x_j$

Supposons donc que nous ayons l'optimum de ce programme, donné par un ensemble d'indices de base $I$, et donc un ensemble d'indices hors base $\overline{I}$. Avec les notations

classiques, on a pour cette base :

$c_{\bar{I}} - c_I[A_I]^{-1}A_{\bar{I}} \leq 0$ (18)

Supposons maintenant que nous fassions varier marginalement les ressources $b_i$. Posons $b' = b + db$ La relation (18) ne dépendant pas de $b, I$ reste l'ensemble d'indices de base optimal, si les variables de base restent non négatives, ce qui est toujours possible pour $db$ suffisamment petit et s'il n'y a pas de dégénérescence, c'est-à-dire de variables de base nulles à l'optimum du programme linéaire initial.

La valeur de la fonction économique, qui était $Z_0 = c_I[A_I]^{-1}b$, devient

$Z'_0 = c_I[A_I]^{-1}b' = Z_0 = c_I[A_I]^{-1}db.$

D'où, si $dZ$ est la variation de la fonction économique $dZ = c_I[A_I]^{-1}db$

Or, on a vu que, si $I$ est l'ensemble de base optimal dans le primal, le vecteur

$y_0 = c_I[A_I]^{-1}$ donnait les valeurs des variables principales à l'optimum du dual, d'où

$dZ = y_0 db$

Si $y_0 = (y_1, \ldots y_m)$, on a donc

$y_0 = \frac{\partial Z}{\partial b_i}$ (19)

Donc, à l'optimum du dual, la variable principale attachée à la 1ère contrainte du primal est égale à la variation de la fonction économique optimale du primal lorsque l'on augmente d'une unité la quantité de la ressource i.

Deux cas sont à considérer :

a) $y_i$ hors base dans le dual, donc nul, alors $dZ = 0$. La variable d'écart correspondante dans le primal est de base et positive, ce qui signifie que la contrainte à laquelle est associée cette variable d'écart est non saturée à l'optimum. Il est donc normal qu'un variation marginale de la quantité de ressource $i$ donne une variation nulle de la fonction économique, puisque pour le sommet optimal, la $i^{ième}$ contrainte ne joue aucun rôle.

b) $y_i$ de base à l'optimum du dual et donc positif; alors $dZ = y_i db_i$. La contrainte correspondante du primal (la $i^{ième}$) est saturée et $y_i$ mesure la variation de $Z$ consécutive à une variation d'une unité de la ressource $i$.

C'est en ce sens que $y_i$ peut être compris comme un prix. En effet, si l'entreprise se pose la question de savoir si une augmentation de la quantité de la ressource $i$ peut être intéressante ou non, elle sera amenée à comparer le bénéfice qu'elle en retire (mesuré marginalement par $dZ = y_i db_i$) au coût de cette acquisition. Si le prix sur le marché d'une unité du bien $i$ est supérieur à $y_i$, l'opération n'est pas rentable pour l'entreprise; elle l'est si, au contraire, ce prix est inférieur à $y_i$.

$y_i$, valeur à l'optimum de la variable duale associée à la contrainte i mesure donc le prix maximum auquel l'entreprise peut acquérir une unité du bien i pour augmenter son profit.

**Remarque :** Encore une fois, nous n'avons fait que donner un "habillage économique" à des considérations d'ordre mathématique. Il ne faudrait pas, en particulier, conclure de ce qui précède que toute entreprise négocie ses prix d'acquisition des facteurs de production sur la base d'un programme linéaire :

*Exemples* : sur l'exemple, on avait (voir tableau optimal du dual)

$y_3 = \frac{1}{2}$

$y_4 = 0$

$y_5 = \frac{1}{2}$

$y_4 = 0$ veut dire que $y_4 > 0$, ce qui signifie que la $2^{\text{ième}}$ contrainte du primal est non saturée à l'optimum de ce programme, ce qui est bien le cas.

$y_3 = \frac{1}{2}$ , $y_5 = \frac{1}{2}$ signifie que les contraintes correspondantes du primal sont saturées (1ère et 3ème contraintes).

Par ailleurs, le prix d'acquisition maximale de l'unité de la ressource correspondante est dans les deux cas de 0,5 en fait de 0,5×4000 = 2000, puisque la contrainte à maximiser était $4000x_1 + 8000x_2$ (voir le chapitre I pour l'énoncé du problème). Donc, pour que le profit augmente, il suffit que le coût d'acquisition d'une heure de travail sur les ateliers I et III soit inférieur à 2000.

# Chapitre 4 • Paramétrisation

Nous allons examiner dans ce chapitre des programmes linéaires où certains éléments (soit de la matrice des contraintes, soit du second membre, soit de la fonction économique) dépendent eux-mêmes d'un paramètre qui peut varier dans certaines limites connues ou inconnues. Cette analyse est très importante dans la mesure où, dans la plupart des cas pratiques, les données ne sont connues qu'avec une certaine précision; il est donc utile de savoir si l'optimum trouvé peut changer lorsque l'on fait évoluer les données à l'intérieur de leurs limites de variations.

Dans la mesure où cette analyse ne met en œuvre aucune technique nouvelle autre que l'algorithme du simplexe et n'exige aucun développement mathématique original, nous allons exposer les problèmes de la paramétrisation sur un exemple, une fois de plus celui de l'entreprise de camions qui nous a déjà abondamment servi dans les chapitres précédents. Rappelons que dans cet exemple, il s'agissait de résoudre le programme linéaire :

$x_1 + 3x_2 \leq 450$

$2x_1 + x_2 \leq 350$

$x_1 + x_2 \leq 200$

$x_{1,} x_1 \geq 0$

$Max\ Z = x_1 + 2x_2$

Rappelons que $x_1$et $x_2$ mesurent les nombres de camions de chaque type produit mensuellement, que les 3 contraintes correspondent aux limitations d'heures de travail dans les trois ateliers, et que la fonction économique représente le profit total, à un facteur près (4000).

Le tableau optimal de ce programme est, rappelons-le : (variables d'écart : $x_3, x_4, x_5$)

| | | | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | | 0 | 1 | +1/2 | 0 | -1/2 | | 125 |
| 2 | 4 | | 0 | 0 | +1/2 | 1 | -5/2 | | 75 |
| 0 | 1 | | 1 | 0 | -1/2 | 0 | 3/2 | | 75 |
| | | | | | | | | | |
| | | | 0 | 0 | -1/2 | 0 | -1/2 | | Z-325 |

Nous allons examiner maintenant dans quelle mesure une variation sur une des données pourrait modifier cet optimum.

## 4.1. PARAMETRISATION DE LA FONCTION ECONOMIQUE

Nous allons supposer que le profit unitaire sur le camion de type I est mal connu (il avait été fixé à 4000 précédemment), et qu'il convient de regarder ce que deviennent les résultats si ce profit unitaire est 4000 (1+λ) où λ varie entre -1 et +∞. Le programme devient :

$x_1 + 3x_2 \leq 450$

$2x_1 + x_2 \leq 350$

$x_1 + x_2 \leq 200$

$x_1, x_1 \geq 0$

$Max\ Z = 4000(1+\lambda)x_1 + 2x_2$

On pourrait bien sûr essayer de résoudre tel quel ce programme en discutant à chaque fois suivant les valeurs de λ. Cependant, on remarquera que **ni les coefficients des contraintes ni le second membre ne dépendent de λ** (seule la fonction économique varie avec ce paramètre), et donc, que le **polyèdre convexe attaché au P.L reste inchangé quelle que soit la valeur de λ, ainsi que les sommets de ce polyèdre convexe.**

On peut donc **partir de la solution optimale trouvée précédemment** qui correspond d'ailleurs à λ=0.

Par rapport au tableau précédent, seule la ligne de la fonction économique va être changée. On calculera les nouveaux gains marginaux par les formules classiques (voir chapitre I).

$$\delta_j = c_j - z_j$$

$$z_j = \sum_{i \in I} c_i\, t_{ij}$$

On obtient le tableau II :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 0 | 1 | +1/2 | 0 | -1/2 | 125 |
| | 4 | 0 | 0 | +1/2 | 1 | -5/2 | 75 |
| (1+λ) | 1 | 1 | 0 | -1/2 | 0 | 3/2 | 75 |
| | | 0 | 0 | (-1/2+1/2λ) | 0 | (-1/2-3/2λ) | Z-325-75λ |

On a, en effet, d'après les formules ci-dessus :

$$\delta_3 = 0-1+\frac{1}{2}(1+\lambda) = -\frac{1}{2}+\frac{1}{2}\lambda$$

$$\delta_5 = 0+1-\frac{3}{2}(1+\lambda) = -\frac{1}{2}-\frac{3}{2}\lambda$$

C'est à partir de là qu'une discussion sur λ s'impose : on voit que $\delta_5$ peut être positif si $\lambda < -\frac{1}{3}$ et que $\delta_3$ peut être positif si $\lambda > 1$.

Distinguons alors 2 cas : $-1 \leq \lambda \leq -\frac{1}{3}$ et $\lambda > -\frac{1}{3}$

1) $-1 \leq \lambda \leq -\frac{1}{3}$ comme $\delta_5 > 0$, il faut faire entrer $x_5$ dans la base. On obtient le tableau III :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 1/3 | 1 | +1/3 | 0 | 0 | 150 |
| 0 | 4 | 5/3 | 0 | -1/3 | 1 | 0 | 200 |
| 0 | 51 | -2/3 | 0 | -1/3 | 0 | 1 | 50 |
| | | $c_i$: 1 + λ | 2 | | | | |
| | | λ+1/3 | 0 | -2/3 | 0 | 0 | Z-300 |

Comme nous sommes dans le cas où $-1 \leq \lambda \leq -\frac{1}{3}$ les gains marginaux du tableau III sont négatifs ou nuls; cet optimum est donc stable pour tout λ tel que $-1 \leq \lambda \leq -\frac{1}{3}$. Passons donc au cas où $\lambda > -\frac{1}{3}$

2) $\lambda > -\frac{1}{3}$

Sur le tableau II, on voit que dans ce cas, $\delta_5 < 0$ Par contre $\delta_3$ est positif si λ>1. On a donc 2 cas :

$-\frac{1}{3} \leq \lambda \leq 1$ et λ>0

3) $-\frac{1}{3} \leq \lambda \leq 1$

L'optimum est celui représenté par le tableau II, puisque dans ce cas $\delta_3$et $\delta_5$ sont négatifs ou nuls.

4) λ > 1

On doit faire entrer la variable $x_3$ dans la base.

On obtient le tableau :

| | | **1** | **2** | **3** | **4** | **5** | |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 0 | 1 | 0 | -1 | (2) | 50 |
| 0 | 3 | 0 | 0 | 1 | 2 | -5 | 150 |
| (1+λ) | 1 | 1 | 0 | 0 | 1 | -1 | 150 |
| | | 0 | 0 | 0 | (1-λ) | (λ-3) | Z-250-150λ |

Sur ce tableau, on voit que seul $\delta_5$ peut être positif si $\lambda > 1$, et c'est vrai si $\lambda > 3$ ; donc 2 cas :

$1 \leq \lambda \leq 3$ et $\lambda > 3$

5) $1 \leq \lambda \leq 3$. Le tableau optimal est le tableau IV.

6) $\lambda > 3$

On fait entrer $x_5$ dans la base. On obtient le tableau V :

| | | **1** | **2** | **3** | **4** | **5** | |
|---|---|---|---|---|---|---|---|
| 0 | 5 | 0 | 1/2 | 0 | -1/2 | 1 | 25 |
| 0 | 3 | 0 | 5/2 | 1 | -1/2 | 0 | 275 |
| (1+λ) | 1 | 1 | 1/2 | 0 | 1/2 | 0 | 175 |
| | | 0 | -1/2(1+λ) | 0 | -1/2(1+λ) | 0 | Z-175(1+λ) |

Aucun gain marginal ne peut devenir positif ; on a donc fini la discussion.

Récapitulons les résultats dans un tableau, donnant en fonction de $\lambda$ les valeurs des variables et de la fonction économique.

| $\lambda$ | -1 | -1/3 | 1 | 3 |
|---|---|---|---|---|
| $x_1$ | 0 | 75 | 150 | 175 |
| $x_2$ | 150 | 125 | 50 | 0 |
| $x_3$ | 0 | 0 | 150 | 275 |
| $x_4$ | 200 | 75 | 0 | 0 |
| $x_5$ | 50 | 0 | 0 | 25 |
| Z | 300 | 325+75λ | 250+150λ | 175(1+λ) |

Les variations de la fonction économique en fonction de λ peuvent être représentées par la courbe :

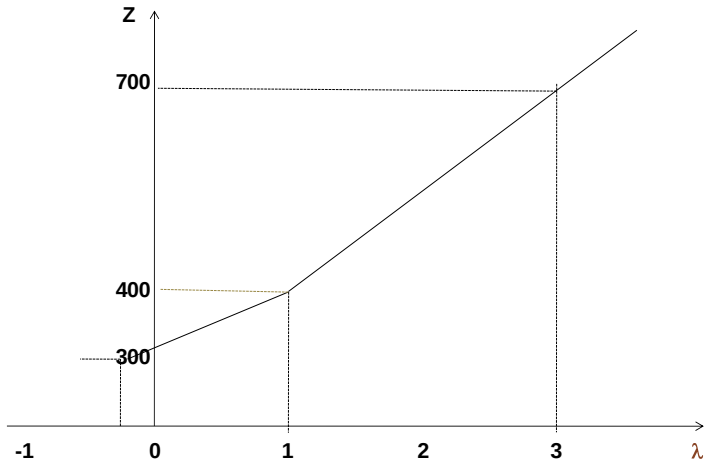



*Figure 1*

Interprétation géométrique : comme au chapitre II, traçons dans $R^2$ le polygone convexe des solutions réalisables du programme linéaire.

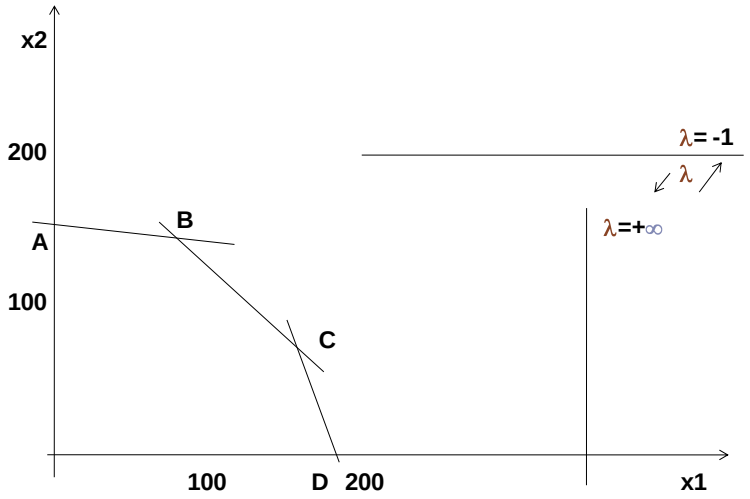



*Figure 2*

Paramétriser un coefficient de la fonction économique, c'est rendre variable la direction de la droite Z = constante ; plus précisément lorsque $\lambda = -1$, $Z = 2x_2$ et les droites Z = constante sont parallèles à l'axe des $x_1$. Sur la figure, l'optimum du programme est alors évidemment *A*. Lorsque λ augmente, la direction de la fonction économique pivote

comme l'indique la figure; on trouve alors, comme optima successifs le point $B$, puis le point $C$, puis le point $D$, qui reste optimum stable jusqu'à $\lambda = +\infty$. On retrouve bien les résultats précédents.

## 4.2. PARAMETRISATION DU SECOND MEMBRE

Supposons maintenant que le nombre d'heures disponible pour l'atelier $I$ soit $450(1+\lambda)$ avec $-1 \leq \mu \leq +\infty$ et non plus 450, la fonction économique n'étant plus paramétrée, c'est-à-dire s'écrivant $Z = x_1 + 2x_2$. Le programme linéaire devient alors :

$x_1 + 3x_2 \leq 450(1+\mu)$

$2x_1 + x_2 \leq 350$

$x_1 + x_2 \leq 200$

$x_1, x_1 \geq 0$

$Max\ Z = x_1 + 2x_2$

Pour résoudre ce problème, on pourrait *a priori* là aussi partir du tableau $I$, quitte à recalculer les seconds membres, qui dépendent de $\mu$, puis discuter suivant les valeurs de $\mu$. Cependant, il convient de rappeler que **l'algorithme du simplexe exige que les seconds membres restent constamment positifs ou nuls,** ce qui n'est peut-être pas assuré du fait de l'intervention du paramètre $\mu$ : si un élément du second membre est négatif, en effet, c'est qu'il s'agit là d'une solution irréalisable.

Pour résoudre cette difficulté, il suffit de se souvenir que lorsque l'on passe au dual, second membre et fonction économique s'échangent. Donc si l'on paramétrise le second membre du primal, cela veut dire que l'on paramétrise la fonction économique du dual.

En conséquence, si, au lieu de rester dans le primal, on travaille dans le dual, on se ramène au problème traité précédemment (paramétrisation de la fonction économique avec un domaine des solutions réalisables fixe).

Là aussi, on peut partir du tableau optimal du dual sans paramétrisation, tel qu'il est écrit au chapitre précédent, à condition bien sûr, de recalculer les gains marginaux en fonction de $\mu$. On a le tableau VI :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| -450(1+$\mu$) | 3 | 1/2 | -1/2 | 1 | -1/2 | 0 | 1/2 |
| 200 | 5 | -3/2 | 1/2 | 0 | (5/2) | 1 | 1/2 |
| Cj : | | | | | -450(1+$\mu$) -350 | -200 | |
| | | -75+225$\mu$ | -125-225$\mu$ | 0 | -75-225$\mu$ | 0 | Z'+325+225$\mu$ |

Rappels : on a

$$\delta_1 = \frac{1}{2}\,450(1+\mu) - 300$$
$$= -75 + 225\mu$$
$$\delta_2 = -225(1+\mu) + 100 = -125 - 225\mu$$
$$\delta_4 = -350 - 225(1+\mu) + 500$$
$$= -75 - 225\mu$$

On voit que $\delta_4$ est positif si $\mu < -\frac{75}{225}$ soit $< -\frac{1}{3}$. Deux cas se présentent donc tout d'abord :

1) $\mu < -\frac{1}{3}$

Faisons entrer $y_4$ dans la base.

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| -450(1+$\mu$) | 3 | +1/5 | -2/5 | 1 | 0 | 1/5 | 3/5 |
| -350 | 4 | -3/5 | +1/5 | 0 | 1 | 2/5 | 1/5 |
| Cj : | | | | -450(1+$\mu$) | -350 | -200 | |
| | | 90$\mu$-120 | -180$\mu$-110 | 0 | 0 | 30+90$\mu$ | Z'+340+270$\mu$ |

On voit alors que $\delta_2$ est positif si $\mu < -\frac{11}{18}$. Deux cas sont alors à distinguer.

2) $-\frac{11}{18} < \mu < -\frac{1}{3}$

Le tableau VII est le tableau optimal du dual. Le tableau du primal s'obtient de façon immédiate :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | -1/5 | +3/5 | 0 | 120-90$\mu$ |
| 2 | 2 | 0 | 1 | +2/5 | -1/5 | 0 | 180$\mu$ + 110 |
| 0 | 5 | 0 | 0 | -1/5 | -2/5 | 1 | -30-90$\mu$ |
| | | 0 | 0 | -3/5 | -1/5 | 0 | Z-340-270$\mu$ |

3) $-1 < \mu < -\frac{11}{18}$

On fait entrer $y_2$ dans le tableau VII du dual. On obtient :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| -450(1+$\mu$) | 3 | -1 | 0 | 1 | 2 | 1 | 1 |
| 0 | 2 | -3 | 1 | 0 | 1 | 2 | 1 |
| Cj : | | -450(1+$\mu$) | -350 | | -200 | | |
| -450(1+μ) | | 0 | 0 | 900μ+550 | 450μ+250 | | Z'+450(1+μ) |

Tous les gains marginaux de ce tableau sont négatifs ou nuls pour $-1 < \mu < -\frac{11}{18}$. Cet optimum est stable sur l'intervalle de $\mu$ considéré. On a, en passant au primal :

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| 1 | 1 | +3 | +1 | 0 | 0 | $450(1+\mu)$ |
| 4 | 0 | -1 | -2 | 1 | 0 | -550-900$\mu$ |
| 5 | 0 | -2 | -1 | 0 | 1 | -250-450$\mu$ |
| | -1 | 0 | 0 | -1 | 0 | Z-450(1+$\mu$) |

On a terminé d'examiner le cas où $\mu < -\frac{1}{3}$. Passons au cas où $\mu > -\frac{1}{3}$. Sur le tableau VI, on voit que $\delta_1$ peut être positif pour $\mu > -\frac{75}{225}$ soit $\mu > \frac{1}{3}$.

Distinguons alors 2 cas :

4) $-\frac{1}{3} < \mu \leq +\frac{1}{3}$

Le tableau optimal du dual reste le tableau VI, et le tableau optimal du primal est le tableau optimal sans paramétrisation, (tableau I), si on prend soin de changer le second membre :

$x_1 = 75 - 225\mu$

$x_2 = 125 + 225\mu$

$x_3 = 0$

$x_4 = 75 + 225\mu$

$x_5 = 0$

par ailleurs $Z = 325 + 225\mu$

5) $\mu > \frac{1}{3}$

Faisons entrer $y_1$ dans la base; on passe du tableau VI au tableau IX :

| | | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | -1 | 2 | -1 | 0 | 1 |
| -200 | 5 | 0 | -1 | 3 | 1 | 1 | 2 |
| | | 0 | -200 | -450$\mu$+150 | -150 | 0 | Z'+400 |

Tous ces gains marginaux y sont négatifs ou nuls.

De ce tableau, on passe immédiatement au tableau correspondant dans le primal :

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| 2 | 1 | 1 | 0 | 0 | 1 | 200 |
| 3 | -2 | 0 | 1 | 0 | -3 | 400$\mu$-150 |
| 4 | 1 | 0 | 0 | 1 | -1 | 150 |
| | -1 | 0 | 0 | 0 | -2 | Z-400 |

La discussion est ainsi terminée et peut se résumer dans le tableau récapitulatif suivant :

| $\lambda$ | -1 | -11/18 | -1/3 | +1/3 | $+\infty$ |
|---|---|---|---|---|---|
| $x_1$ | 450(1+$\mu$) | 120-90$\mu$ | 75-225$\mu$ | 0 | |
| $x_2$ | 0 | 110+180$\mu$ | 125+225$\mu$ | 200 | |
| $x_3$ | 0 | 0 | 0 | 450$\mu$-150 | |
| $x_4$ | -550-900$\mu$ | 0 | 75+225$\mu$ | 150 | |
| $x_5$ | -250-450$\mu$ | -30-90$\mu$ | 0 | 0 | |
| Z | 450(1+$\mu$) | 340+270$\mu$ | 325+225$\mu$ | 400 | |

Les variations de la fonction économique en fonction de μ peuvent être représentées de la façon suivante :

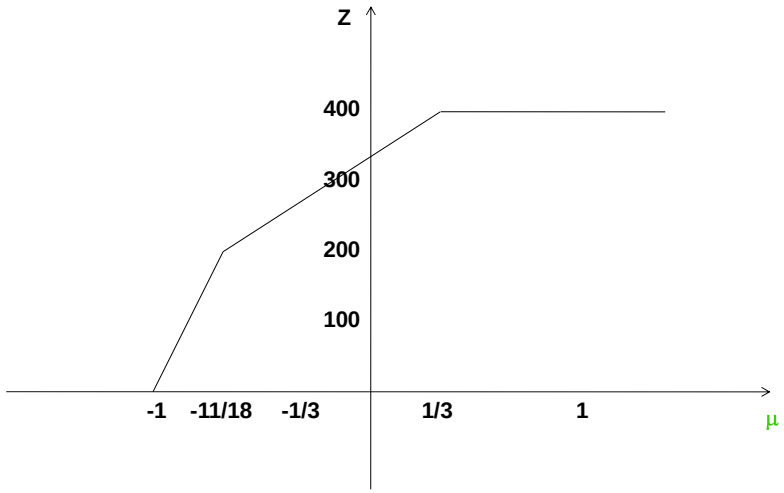



*Figure 3*

**Interprétation géométrique**

Reprenons la figure 2 de la page 87 et examinons quels sont les effets de la paramétrisation du second membre de la 1ère inéquation.

Les contraintes 2 et 3 sont fixes ainsi que la direction de la fonction économique. On a alors immédiatement l'interprétation des résultats trouvés ci-dessus lorsque μ varie de -1 à +∞.

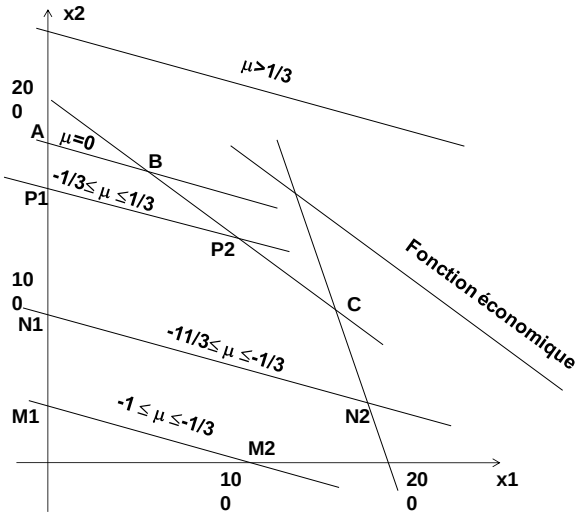



*Figure 4*

Lorsque $-1 < \mu \leq -\frac{11}{18}$, le polygone des solutions réalisables est le triangle $OM_1M_2$ (les autres contraintes sont superflues).

L'optimum est au point $M_2(x_1 = 450(1+\mu);\ x_2 = 0$. Lorsque $-\frac{11}{18} < \mu \leq -\frac{1}{3}$ le polygone des solutions réalisables est le quadrilatère $ODN_1N_2$. L'optimum est en $N_2$.

Lorsque, $-\frac{1}{3} < \mu \leq +\frac{1}{3}$, le polygone des solutions réalisables est $OP_1P_2CD$. L'optimum est en $P_2$ (pour $\mu = 0$, on retrouve $B$).

Si $\mu > \frac{1}{3}$, le polygone des solutions réalisables est $OQCD$, la première contrainte devient superflue et l'optimum est en $Q(x_1 = 0, x_2 = 200)$

**Remarque :** On pouvait naturellement se passer d'utiliser l'algorithme du simplexe sur ce petit exemple à deux dimensions. Evidemment, cette solution géométrique devient impossible dès que le nombre de variables dépasse 3.

## 4.3. PARAMETRISATION D'UN COEFFICIENT DE LA MATRICE DES CONTRAINTES

On peut penser examiner également la sensibilité des résultats donnés par le programme linéaire aux variations d'un des coefficients, intervenant dans les contraintes en posant par exemple $a'_{kl} = a_{kl}(1+\theta)$ pour un certain $k$ et un certain $l$. Le problème est en général beaucoup plus ardu que les deux précédents; il est facile de voir en effet que, à une itération donnée du simplexe la matrice simpliciale $((A_I)^{-1}A_{\bar{I}}$, le second membre $(A_I{}^{-1}b)$ et la ligne des gains marginaux $(c_I - c_I(A_I)^{-1}A_{\bar{I}})$ dépendent tous du paramètre en question, et parfois de façon assez complexe. On se retrouve donc aux prises avec le problème du second membre pouvant être négatif, combiné avec les difficultés posées par le fait d'avoir le paramètre $\theta$ dans tous les éléments des tableaux. Les calculs sont donc là beaucoup plus difficiles et requièrent d'autres algorithmes, en particulier l'algorithme composite (qui permet de traiter simplement les seconds membres négatifs sans passer au dual). Ces calculs dépassent le cadre de ce simple exposé.

# Chapitre 5 • Compléments et autres algorithmes

Ce chapitre bref est destiné à indiquer au lecteur un certain nombre de travaux parfois récents visant soit à proposer d'autres méthodes de résolution que l'algorithme du simplexe soit à prolonger la programmation linéaire elle-même vers la résolution de problèmes différents de ceux qui ont été traités jusqu'ici.

## 5.1. AUTRES METHODES

L'algorithme du simplexe est jugé de façon générale comme performant. Il existe de nombreux programmes dans les bibliothèques des ordinateurs relevant des principes de calcul que nous avons développés et capables de traiter des problèmes de grande taille (plusieurs milliers de contraintes et de variables) dans des temps acceptables. L'algorithme du simplexe n'est pas exactement programmé comme l'indique l'organigramme de la page 59. On utilise en général une forme un peu différente, plus économe en place mémoire, et fondée sur le fait que lorsque l'on passe d'une itération à l'autre, on change à la marge la matrice $A_1$ .

Cela dit, ces performances mêmes ne manquent pas d'étonner. En effet, l'algorithme du simplexe n'a aucune raison *a priori* de converger rapidement. Souvenons-nous en effet que l'optimum de la fonction linéaire est trouvé en un sommet du polyèdre convexe formé par les contraintes (celles de non négativité des variables incluses) et que ce nombre de sommets est borné par $C_{m+n}^{m}$ Ce nombre croît exponentiellement avec $m$ et $n$, et même si l'algorithme n'explore pas tous les points correspondants (l'intersection de $n$ hyperplans pris parmi les contraintes peut n'être pas réalisable), il peut en explorer un grand nombre. Plus précisément il est facile de voir que le temps de calcul d'une application du simplexe à un problème donné ne peut être borné par une fonction polynomiale de la taille de ce problème (représentée par le nombre de bits nécessaire pour entrer les données dans un ordinateur). On a affaire, et on verra cela plus précisément à propos de l'application de la théorie des graphes, à un algorithme non polynomial.

Les spécialistes de programmation linéaire se sont longuement inquiétés de ce fait, uniquement rassurés par les performances empiriques du simplexe, sans trouver de parade, jusqu'à ce qu'un article du russe N. Karmarkar, publié en 1984, propose un algorithme polynomial et provoque une véritable avalanche d'approches nouvelles, se présentant souvent comme des variantes peu différentes les unes des autres. En fait la publicité faite à cette contribution ne doit pas cacher que d'autres auteurs avaient déjà trouvé auparavant de telles méthodes, mais c’est une autre histoire

Il est vrai également que d'autres spécialistes se sont ingéniés à construire des problèmes pour lesquels l'algorithme du simplexe s'est montré particulièrement « poussif » , comme le problème suivant :

$$Max \sum_{j=1}^{n} 10^{(j-1)} x_j$$

$$x_i + 2 \sum_{j>1} 10^{(j-i)}\, x_j \leq 10^{(2n-2i)}\ i = 1 \ldots n$$

$$x_j \geq 0$$

Pour $n = 100$, on évalue le temps de calcul d'un ordinateur effectuant un million d'itérations par seconde (ce qui n'est pas mal) à 400 trillions d'années ! Il est vrai que le problème est très spécifique, ne correspond à aucun exemple économique réel, et est clairement *ad hoc*, pour les besoins de la démonstration. Il n'en reste pas moins qu'il paraît légitime de rechercher si d'autres méthodes que le simplexe ne pourraient pas le traiter plus rapidement.

Ces méthodes existent maintenant et présentent deux différences de principe essentielles avec le simplexe :

- elles ne suivent pas le « contour » du polyèdre en passant de sommet en sommet, mais partent d'un point intérieur à ce polyèdre (une solution réalisable) en essayant de progresser vers sa frontière en direction de l'optimum
- elles ne visent pas à rencontrer au bout d'un certain nombre d'itérations l'optimum mais à s'en approcher d'une distance fixée à l'avance, faible, introduite sous la forme d'un seuil. A ce titre, elles s'apparentent davantage à des méthodes classiques de convergence pas à pas vers des optima de fonctions à plusieurs variables, telles qu'on peut les consulter dans des ouvrages généraux sur l'optimisation (alors que la forme particulière d'un programme linéaire donne lieu à une méthode de résolution spécifique -en l'occurrence le simplexe- et, il faut bien le dire, particulièrement élégante).

On distingue classiquement deux types d'approches pour les nombreuses techniques relevant de ce chapitre de la R.O., lui-même restant largement de l'ordre de la recherche :

- l'approche projective
- les méthodes de chemin intérieur

En fait, toutes ces méthodes sont assez voisines et d'ailleurs comme on le verra sont toutes fondées sur l'idée d'un cheminement interne au polyèdre. Comme leur exposé détaillé devient rapidement technique, nous nous bornerons, dans le cadre de cet ouvrage de base, à en décrire les grands principes.

### 5.1.1. Les approches projectives

Ce nom vient du fait que l'on a affaire avec un programme linéaire à l'optimisation d'une fonction de *n* variables soumises à *m* contraintes. Si l'on ne fait pas attention dans un premier temps à la linéarité de la fonction et des contraintes, une méthode classique d'optimisation consiste à partir d'une solution réalisable et à l'améliorer en partant dans une direction appropriée, ce processus étant répété un certain nombre de fois, jusqu'à ce que l'on estime, grâce à un critère prédéfini, s'être approché avec une précision suffisante de la solution. Une telle direction est donnée par le gradient de la fonction, qui maximise la croissance de la fonction (dans un problème de maximisation) pour une petite variation des variables. (Rappelons que le gradient d'une fonction $f(x_1, \dots x_j, \dots x_u)$ est donné par le vecteur des dérivées premières, si elles existent $\left(\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_j} \dots \frac{\partial f}{\partial x_u}\right)$). Le problème est que si l'on part du programme linéaire mis sous forme standard (augmenté de ses *m* variables d'écart permettant de convertir les inégalités de départ en égalités), il est alors facile de voir qu'en suivant cette direction, on sort immédiatement du domaine des solutions réalisables.

Pour s'en convaincre, prenons un petit exemple. Soit le programme linéaire suivant :

$Max\ x_1 + x_2$

$x_1 + 2x_2 \leq 8$

$x_1, x_2 \geq 0$

Après addition de la variable d'écart nous obtenons :

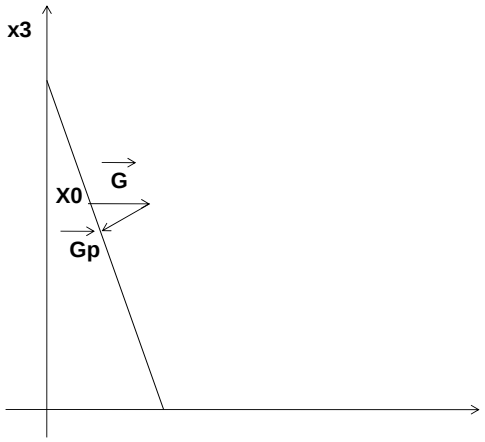
$Max\ x_1 + x_2 + 0x_3$

$x_1 + 2x_2 + x_3 = 8$

$x_1, x_2, x_3 \geq 0$

Si nous partons du point $x_0 \begin{pmatrix} 1 \\ 1 \\ 5 \end{pmatrix}$ il s'agit clairement d'une solution réalisable. Le gradient de la fonction économique est $\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ et tout point s'écrivant $x_0 + \propto \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ avec $\propto > 0$ ne répond pas à la contrainte.

Ce qui est alors proposé dans l'approche projective est de projeter le vecteur du gradient sur l'espace des solutions réalisables. Sur le dessin (qui représente le plan passant par l'axe $x_1 = x_2 = 0$ et le point $X_0$ cela consiste à tracer dans $R^3$ une droite partant de l'extrémité du gradient $\overrightarrow{G_P}$ et à calculer l'intersection de cette droite avec le plan de la contrainte. Le vecteur $\vec{G}$ donne alors la direction dans laquelle se diriger.




En algèbre vectorielle on démontre que ce vecteur est donné par l'équation suivante :

$G_P = [I - A^t(AA^t)^{-1}A]G$ (1)

où $I$ est la matrice unité $3 \times 3$, $A$ la matrice des contraintes. Ici $A = (1,2,1)$ et $A^t$ est la transposée de $A$.

On trouve ici : $\begin{pmatrix} 0,5 \\ 0 \\ -0,5 \end{pmatrix}$

Ainsi est fixée la direction à suivre. Reste à savoir la longueur du pas que nous allons parcourir, ou encore la valeur du paramètre $\propto$ dans :

$X_1 = X_0 + \propto G_P \qquad \propto > 0$

Dans l'optimisation classique où l'on utilise des procédures de type Cauchy ou Taylor, on utilise les dérivées secondes de la fonctionnelle pour fixer ce pas. Ici, compte tenu de la linéarité de la fonction, une telle démarche n'a pas de sens. On se fonde plutôt sur le fait qu'aucune des variables (principales ou d'écart) ne doit être négative. Comme $G_P$ est une projection sur l'espace définie par la contrainte, on ne viole pas cette dernière. Reste à s'assurer que $x_j \geq 0$.

Cette condition donne :

$\propto = min \left| \frac{x_j^0}{g_j^p} \right| \; pour g_j^p < 0$ (2)

$g_j^p$ étant l'élément courant de $g_p$

C'est ici que dans les algorithmes proposés, une considération intuitive intervient : le pas effectué sera d'autant plus grand que le point de départ $X_0$ sera équidistant des axes de coordonnées. En effet, d'après la formule ci-dessus, si un des $X_j$ est faible (point $X_0$ proche de l'axe correspondant), le paramètre $\propto$ sera faible.

L'idéal est de partir du point $\begin{pmatrix}1\\1\\1\end{pmatrix}$. Un moyen simple d'obtenir un tel point est de faire subir à notre problème de départ une transformation affine. Posons $\widehat{X_0}$ matrice carrée à trois dimensions dans notre exemple (à $m+n$ dimensions dans le cas général), dont la diagonale principale est composée des $x_j^0$, dans l'ordre, les autres éléments étant nuls. Il est facile de voir que la transformation

$$\widetilde{X_0} = (\widehat{X_0})^{-1} X_0$$

donne un vecteur colonne rempli de 1. Evidemment il faut transformer également la matrice (la ligne dans l'exemple) et la fonction économique, par les formules suivantes :

$$\tilde{A} = A\widehat{X_0}$$

$$\tilde{c} = c\widehat{X_0}$$

($A$ matrice des contraintes, $c$ vecteur des coefficients de la fonction, le PL étant sous forme standard).

On s'assure sans mal que ce nouveau PL est équivalent au précédent.

Prenons alors le point de départ $\begin{pmatrix}1\\1\\1\end{pmatrix}$ (vecteur colonne rempli de 1). C'est sur ce point de départ et sur le PL transformé que nous faisons les opérations précédentes. Après calcul de la projection du gradient sur l'espace des contraintes, la longueur du pas est *a priori* donnée par la formule (2). Sauf que si nous répétons cette séquence d'opérations, transformation affine comprise (pour chaque nouveau point trouvé, on le transforme, ainsi que le PL), par la formule (3), il est clair que nous nous interdisons que l'un quelconque des $x_j$ soit nul. On introduit alors un autre coefficient, $\mu$, choisi pour aller vite (en général $0{,}90 < \mu < 1$), qui réduit un peu la longueur du pas trouvé.

De façon générale, et sur un problème $(m, n)$, cette méthode se traduit par les itérations suivantes (avec une adaptation des notations précédentes):

a) $X_k$ solution itération $k$

Faire $\widetilde{X_k} = (\widehat{X_k})^{-1} X_k$ $\widetilde{A_k} = A(\widehat{X_k})^{-1}$ $\tilde{c} = (\widehat{X_0})^{-1} c$

b) $G_k^p$ (projection du gradient $G_k$ sur l'espace des solutions réalisables)

$$G_k^p = [I - (\widehat{A_k})^t [(\widetilde{A_k})(\widehat{A_k})^t]^{-1} \widetilde{A_k}]\, G_k$$

c) $\propto_K = \min \left[\frac{\tilde{x}_j^k}{g_{kj}^p}\right] pour\ g_{kj}^p < 0 \quad j = 1 \ldots n+m$

d) $\tilde{x}_{k+1} = \tilde{x}_k + \mu \alpha_k G_k^p$

e) $X_{k+1} = \hat{X}_{k+1} \tilde{X}_{k+1}$

On s'arrête lorsqu'un critère de convergence est respecté. Il peut être de plusieurs types. Classiquement, si $x_k$ est la valeur de la fonction économique à l'itération $k$, on peut s'arrêter lorsque

$$\frac{|z^{k+1} - z^k|}{|z_k|} < \varepsilon$$

$\varepsilon$ étant un seuil prédéfini faible ($10^{-5}$ par exemple).

D'autres critères peuvent être utilisés comme la longueur du gradient lui même. Un critère plus sophistiqué et plus sûr utilise le dual du PL. En effet, on peut, parallèlement à la séquence d'itérations décrites ci-dessus et effectuées sur le primal, traiter le dual de la même façon. On sait que lorsque l'on a deux solutions réalisables, l'une dans le primal et l'autre dans le dual, on a toujours (cf. chapitre 3) :

$cx \leq yb$

On s'arrête alors lorsque :

$|Y_k b - cX_k| < \varepsilon$

Quant au choix du point de départ, il n'est pas toujours évident. Il doit en effet correspondre à une solution réalisable; évidemment la situation idéale est celle où le point $x_j = \forall j$ satisfait aux contraintes. Mais, comme on le sait, les PL n'ont pas toujours le bon goût de se présenter comme cela. Des procédures existent, qui dépassent le cadre de cet exposé, mais qui peuvent sensiblement affecter la performance de ce type de méthode.

Sur l'exemple choisi, une telle procédure fournit l'optimum $x_1 = 8\, x_2 = 0\, z = 8$ en sept itérations avec une précision de $10^5$. Le simplexe de son côté donne l'optimum exact en une itération seulement. Cela dit, il est clair que ces algorithmes de point intérieur se justifient sur des PL plus « gros » et /ou plus « tordus ».

**Remarque** : il s'agit d'une procédure particulière. De nombreuses variantes existent. D'ailleurs, l'algorithme initiateur de ce courant, celui de Karmarkar, ne se présentait pas tout à fait de la même façon. Il proposait une transformation initiale du PL de telle façon qu'aux contraintes classiques s'ajoute $\sum x_j = 1$ (ce qui est toujours possible si on calcule les bornes supérieures des $x_j$ et si l'on pose $x_j = \frac{x_j}{M}$ où M est la somme de ces bornes). Ensuite la transformation projective utilisée était un peu plus complexe que celle proposée ci-dessus. Enfin, Karmarkar exploitait le fait que le point trouvé à chaque itération pouvait être considéré comme le centre d'une sphère inscrite dans le simplexe dans $R^{m+n}$, c'est-à-dire l'espace défini par $\sum x_j = 1$. Le point suivant était défini alors par l'intersection de la projection du gradient avec cette sphère.

Peu importe : le but de cet exposé n'est pas de développer toutes ces méthodes mais d'en faire comprendre la logique générale qui, comme on le voit, est qualitativement différente de celle qui préside à l'algorithme du simplexe.

### 5.1.2. Méthodes de chemin intérieur

La dénomination de ce type de méthode n'est pas très explicite, car avec les méthodes précédentes, on a affaire clairement à un cheminement intérieur. Leur réservant cependant ce terme, voyons les principes sur lesquelles elles se fondent (elles aboutissent d'ailleurs *in fine* à des formalisations voisines de celles que l'on a vues).

L'idée est la suivante : si nous reprenons notre petit exemple, on peut penser se débarrasser des contraintes embêtantes que sont $x_j \geq 0$. Pour les autres contraintes (en l'occurrence une seule dans notre petit exemple), elles sont transformées en égalités grâce aux variables d'écart (ici $x_3$). Compliquons la formule de la fonction économique de la façon suivante :

$$f(x) = x_1 + x_2 + \mu[Logx_1 + Logx_2 + Logx_3]$$

Où μ est un nombre très petit. On s'interdit ainsi d'avoir des variables, principales ou d'écart, qui soient négatives ou nulles (on a introduit des fonctions « barrières »). μ étant très petit, on traite à peu près la même fonctionnelle. On peut s'inquiéter du fait que l'on n'atteint jamais un $x_j = 0$ , ce qui se produit à l'optimum pour un certain nombre d'entre eux, mais justement, comme précédemment, l'ambition de ces méthodes est uniquement d'approcher la solution à un seuil près, faible, et donc cela ne pose pas de problème non plus.

Mais on se trouve alors confronté au problème classique d'optimisation d'une fonction de *n* variables soumises à *m* contraintes d'égalité. (on n'a plus d'inégalités) et on peut appliquer le dispositif des multiplicateurs de Lagrange. Pour l'exemple, si le multiplicateur est λ, on a le lagrangien suivant :

$$L = x_1 + x_2 + \mu[Logx_1 + Logx_2 + Logx_3] - \lambda(x_1 + 2x_2 + x_3 - 8$$

Et les conditions du premier ordre donnent :

$$x_1 + \mu - \lambda x_1 = 0$$

$$x_2 + \mu - 2\lambda x_2 = 0$$

$$\mu - \lambda x_3 = 0$$

Inconnues : $x_1, x_2, x_3, \lambda$

$$x_1 + 2x_2 + x_3 = 8$$

Evidemment, même sur cet exemple très simple, la résolution de ce système d'équations, non linéaire (où les inconnues sont les variables), n'est pas évident (cela n'est pas pour surprendre; dans le cas contraire, on n'aurait pas inventé l'algorithme du simplexe et on ne se serait pas intéressé non plus aux programmes linéaires).

Ecrivons le même système d'équations dans le cas général, mais en l'enrichissant par un calcul analogue sur le dual (l'algorithme correspondant porte le nom de primal-dual intérieur). Sous forme matricielle, prenons les notations suivantes : $A$ matrice des contraintes du primal sous forme canonique, $X$ et $Y$ vecteurs colonnes des variables principales du primal et du dual, $X^e$ et $Y^e$ vecteurs colonnes des variables d'écart, $b$

colonne du second membre du primal, *c* ligne des coefficients de la fonction économique du primal. Alors primal et dual s'écrivent :

$AX + IX_e = b$

$Max\ cX + 0X_e$ (Primal)

$A^tY - IY_e = c^t$

$Min\ b^t + 0Y_e$ (dual)

Où 0 désigne un vecteur ligne rempli de 0, où on a omis la dimension, par simplification, et *I* la matrice carrée unité, dimensions également omises.

Les fonctions « barrières », pour les deux programmes, s'écrivent :

$$L_\rho = cX + \mu\left[\sum_{j=1}^{n} Log\, x_j + \sum_{j=n+1}^{m+n} Log\, x_j^e\right] \qquad (\mathrm{Pr}imal)$$

$$L_D = b^tY - \mu\left[\sum_{i=1}^{n} Log\, y_i + \sum_{i=m+1}^{m+n} Log\, y_i^e\right] \qquad (Dual)$$

Les conditions du premier ordre, après calculs, donnent :

$AX + I\,X_e = b$

$A^tY - IY_e = c^t$

$y_{m+j}^e\, x_j = \mu \qquad \forall_j = 1.....n$

$x_{n+i}^e\ y_i = \mu \qquad \forall_i = 1.....m$

Il s'agit d'un système non linéaire. Si μ = 0, on retrouve les résultats généraux sur les optima des deux programmes : lorsqu'une variable est de base - positive sauf dégénérescence -, la variable qui lui correspond dans l'autre programme est nulle. En effet dans ce cas les deux dernières conditions donnent : $xy = 0$ (avec $x\ et\ y$ se correspondant).

En général, on ne peut pas résoudre de façon directe ce système, mais on peut adopter une méthode de résolution pas à pas. De telles méthodes existent (Newton-Raphson par exemple) et sont exposées dans des ouvrages généraux sur l'optimisation. Comme elles consistent à partir d'une solution admissible et à la transformer progressivement de façon à s'orienter vers les valeurs des variables qui respectent l'ensemble des contraintes, on adopte alors ce cheminement ici, en partant à chaque fois du point obtenu à l'étape précédente, et en diminuant progressivement μ. (Prendre μ d'emblée très faible conduit à

une impasse au niveau des calculs). L'arrêt est donné lorsque l'écart entre fonction primale et fonction duale est faible (ici ce test est particulièrement aisé à manipuler parce que l'on démontre aisément que cet écart est égal à $(n + m)\mu$).

Là aussi il s'agit de principes généraux, mobilisés dans de nombreuses variantes.

Pour finir sur ces méthodes de point intérieur, la question que l'on est en droit de se poser est : sont-elles réellement plus efficaces que l'algorithme du simplexe ? La réponse est en fait ambiguë; on sent bien qu'il n'y a pas de résultats généraux qui permettraient de statuer sur la supériorité d'une méthode dans tous les cas, et que cela doit dépendre de la taille et de la forme des PL. Si bien que l'on ne peut faire en la matière qu'œuvre empirique, c'est-à-dire tester les deux types de méthode sur de nombreux programmes. Les expériences qui ont été menées jusqu'ici conduisent à des conclusions variables : même avec des programmes de grande taille l'algorithme du simplexe peut être meilleur en temps de calcul qu'une méthode de point intérieur. Finalement, la méthode non polynomiale (alors que Karmarkar, par exemple, a montré que son algorithme convergeait en un temps borné par une fonction en $n$ de la taille $n$ du PL - on précise ces notions plus loin) se révèle particulièrement efficace ! Attendons les résultats des recherches en cours pour y voir plus clair...

## 5.2. PROLONGEMENTS

Il existe de nombreux prolongements de la programmation linéaire, au sens où l'on a toujours affaire à l'optimisation d'une fonctionnelle linéaire, les variables étant elles-mêmes soumises à des contraintes linéaires, mais où l'on ajoute des spécifications qui peuvent notoirement compliquer la résolution du problème.

***Programmes linéaires en variables entières***

Une de ces complications, et non la moindre, est celle où l'on garde la forme générale du PL, telle qu'on l'a introduite en début de cette partie, mais où l'on astreint les variables à prendre des valeurs entières. Cette spécification se présente dans de nombreux cas concrets. Prenons notamment le programme de production, exemple phare de la programmation linéaire, mais où les quantités de produit sont nécessairement entières ; si pour un constructeur aéronautique, à la production nécessairement limitée quantitativement, on trouve par PL une production de 12, 57 avions d'un type donné par an, on peut penser que la réponse n'est pas entièrement satisfaisante (la même problématique se pose différemment pour un constructeur automobile dont la production annuelle se chiffre en dizaines de milliers d'unités, ce qui permet sans état d'âme de prendre les arrondis de la solution réelle trouvée en faisant abstraction de la contrainte d'intégrité sur les variables).

L'autre exemple typique est celui des choix d'investissements : si l'on prend une entreprise qui est confrontée à $n$ investissements possibles, chacun étant caractérisé par une performance économique (valeur actuelle, par exemple), et sachant que le budget d'investissement est limité par une quantité $B$ la question est : quels investissements choisir ? Il est facile de voir que ce problème peut se mettre sous la forme suivante : on

définit une variable $x_i$ pour chaque investissement $i$, avec $x_i = 0$ si on ne fait pas l'investissement et $x_i = 1$ si on le fait. Répondre à la question, c'est alors résoudre le PL suivant :

$$Max \sum_{i=1}^{n} v_i x_i$$

$$\sum_{i=1}^{n} x_i B_i \leq B$$

$$x_i = 0 \; ou \; 1$$

mais où les variables sont astreintes à prendre deux valeurs et deux valeurs seulement, 0 ou 1. C'est un cas particulier des programmes linéaires en nombres entiers, dit « à variables bivalentes ».

Le problème général des PL en variables entières est qu'il faut se garder de l'intuition : résoudre le PL en variables réelles et prendre la solution « arrondie » aux entiers inférieurs peut non seulement conduire à une solution non optimale, mais aussi à une solution non réalisable !

Si bien que là aussi de nombreuses recherches se sont développées sur ce type de programme linéaire, qui s'est révélé particulièrement rebel à l'algorithmique.

Sans entrer dans la description de ces méthodes, on peut simplement dire ici qu'elles se classent en deux catégories : les méthodes dites de troncature, où l'on introduit des contraintes supplémentaires qui, dans l'espace des solutions réalisables, définissent des hyperplans tels que tous les points réalisables entiers du domaine sont dans un seul des demi-espaces délimités par ces hyperplans. On résout alors une série de PL « emboîtés » en variables réelles, ajoutant à chaque fois une de ces troncatures, jusqu'à trouver un optimum entier.

Un autre type de méthode consiste à utiliser des procédures d'optimisation arborescentes, telles qu'elles sont définies dans la partie suivante.

D'autres méthodes mixent les deux approches. Comme on n'a pas encore trouvé pour ce problème de méthodes performantes (au sens où l'on est jamais sûr, pour un cas donné, de la convergence en un temps informatique raisonnable) on peut également utiliser des heuristiques, telles que les algorithmes génétiques, eux aussi évoqués plus loin.

***Goal programming***

Ce terme vient du fait que les finalités sur lesquelles repose le problème que l'on veut traiter, et qui jusqu'ici étaient résumées par la fonction économique, peuvent être multiples, plus ou moins contradictoires et floues (on veut par exemple maximiser le surplus de l'entreprise tout en diminuant si possible la pollution de l'environnement). On peut alors introduire des seuils à atteindre pour certains critères et optimiser l'un d'entre eux, ou encore introduire les critères avec des marges autour de « buts » à atteindre. En

d'autres termes, on introduit, sous des formes variables suivant le problème, de la flexibilité dans les calculs.

### *Programmes linéaires et graphes*

Certains programmes linéaires, comme on le verra, ont des formes très particulières (programmes de transport, d'affectation, flots dans les réseaux, etc.) et seront résolus de façon beaucoup plus aisée par la théorie des graphes que par l'algorithme du simplexe (partie suivante).

Enfin, signalons que récemment se sont développées des méthodes économétriques performantes utilisant la programmation linéaire : il s'agit de la méthode DEA (Data Envelopment Analysis) qui permet de comparer l'efficience d'unités de production à produits similaires mais multiples, à partir du moment où l'on connaît leur production et leurs inputs (main d'œuvre, consommables, équipements etc.). La programmation linéaire permet ainsi de résoudre un vieux problème des économistes, très difficile à traiter et qui consiste à comparer la productivité d'entités à produits multiples.

# Chapitre 6 • Problèmes de chemins

## 6.1. DEFINITION D'UN GRAPHE

Dans la pratique, un graphe se présente de la façon suivante :

- un ensemble de points, fini
- des flèches reliant certains de ces points entre eux.

*Par exemple :*

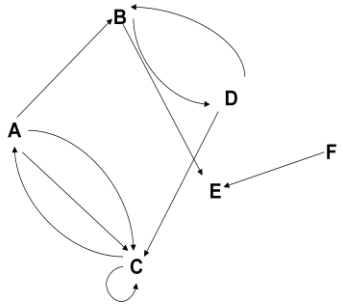



*Figure **1***

Les points seront appelés sommets, les flèches arcs.

D'une façon plus précise, un graphe $G = (X, U)$ est constitué par le couple :

- d'un ensemble $X = (x_1, x_2, \dots x_n)$ de sommets, fini
- d'une liste d'éléments du produit cartésien $X \times X = \{(x, y)/x \in X, y \in X\}$

On notera que l'ordre $x, y$ a de l'importance (il définit l'orientation des arcs). On remarquera également que l'arc $(x, y)$peut revenir plusieurs fois dans la liste.

Pour un arc quelconque $(x, y)$, $x$ sera appelé *l'extrémité initiale* et $y$ *l'extrémité terminale.*

Un arc du type $(x, x)$ sera appelé boucle (sur le graphe de la figure 1, un tel arc relie $C$ à lui-même).

Si le nombre d'arcs reliant un sommet quelconque $x_i$ à un sommet quelconque $y_j$ ne dépasse pas $p$, on dira qu'on a affaire à un *p-graphe*.
(Le graphe de la figure 1 est un 2-graphe).

Dans certains problèmes, l'orientation des arcs n'importe pas : ce qu'il s'agit de savoir, c'est si deux sommets sont reliés par un arc ou pas, et éventuellement le nombre d'arcs reliant ces sommets.

Dans ces conditions, l'ordre $(x, y)$ n'est plus important. On appellera arête toute paire de sommets reliés par un arc, et au lieu de se donner la liste des arcs, on se donnera la liste $E$ des arêtes. On obtiendra un graphe $G = (X, E)$ *sans orientation.* On parlera alors de *multigraphe.*

À un graphe orienté, on peut faire correspondre un multigraphe, en supprimant les orientations. Par exemple, le graphe de la figure 1 donne le multigraphe suivant, si l'on supprime les orientations des arcs :

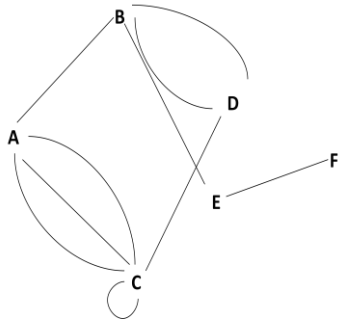



*Figure 2*

Le concept de graphe est aujourd'hui très utilisé dans des disciplines diverses (psychologie, sociologie, mathématique, etc.). En ce qui concerne la recherche opérationnelle, nous allons voir que ce concept est particulièrement bien adapté au traitement de certains problèmes combinatoires, c'est-à-dire de problèmes tels que l'exploration systématique de toutes les solutions serait possible, mais s'avèrerait beaucoup trop coûteuse en temps.

C'est ainsi que nous examinerons :

- *des problèmes de « chemins minimaux »*. (sur un graphe donné, par exemple des villes - les sommets - et des routes - les arcs - où à chaque arc est associée une longueur, trouver le chemin entre deux villes données qui soit de longueur totale minimale)
- des *problèmes de flots* (sur un graphe donné représentant par exemple un réseau de canalisations, où à chaque arc est associée une capacité, faire passer le flot total maximal),
- *des problèmes d'ordonnancement* (par exemple le fameux problème du voyageur de commerce, qui doit passer une fois et une seule dans un certain nombre de villes et qui se pose la question de l'ordre de visite de ces villes de façon que la distance totale parcourue soit la plus faible possible), ainsi qu'un certain nombre d'autres problèmes que la théorie des graphes permet de résoudre.

Nous commencerons dans ce chapitre à analyser des problèmes de "chemins". Auparavant, il est indispensable de donner quelques éléments succincts du vocabulaire des graphes.

## 6.2. ELEMENTS DE VOCABULAIRE DE LA THEORIE DES GRAPHES

### 6.2.1. Application

On peut associer à un graphe $G = (X, U)$ une correspondance de $X$ dans lui-même. Pour cela, on fait correspondre à $x$ l'ensemble des $y$ tels qu'il existe un arc d'extémité initiale $x$ et d'extrémité terminale $y$. On appellera $\Gamma$ cette correspondance, qui est une application multivoque.

S'il s'agit d'un 1-graphe (entre un sommet $x$ quelconque et un sommet $y$ quelconque, il ne peut y avoir que 0 ou 1 arc) le graphe est complètement donné par $X$ et $\Gamma$ et on pourra noter : $G = (X, \Gamma)$

Sur le graphe de la figure 1, où l'on a :
$X = \{A, B, C, D, E, F\}$

on peut écrire :
$\Gamma(A) = \{B, C\}$
$\Gamma(B) = \{D, E\}$
$\Gamma(C) = \{A, C\}$ etc.

mais ici le graphe n'est pas entièrement donné par $\Gamma$(il y a deux arcs $AC$).

### 6.2.2. Image réciproque

Si $x$, sommet d'un graphe $G = (X, U)$est l'extrémité terminale d'au moins un arc de $G$, on appelle image réciproque de $x$ le sous-ensemble des extrémités initiales des arcs qui y aboutissent, c'est-à-dire :
$\Gamma^{(-1)}(x) = \{y \mid x \in \Gamma(y)\}$

Ainsi sur le graphe de la page 1, les images réciproques de $A, B, C$ sont elles :
$\Gamma^{-1}(A) = \{C\}$
$\Gamma^{-1}(B) = \{A, D\}$
$\Gamma^{-1}(C) = \{A, C, D\}$ etc.

### 6.2.3. Graphe partiel - sous-graphe

Soit un graphe $G = (X, U)$; si l'on supprime certains arcs dans le graphe $G$, on obtient un graphe $G' = (X, U')$ avec $U' \subset U$. $G'$est appelé *graphe partiel* de $G$.

Maintenant, si l'on supprime dans $G$ certains sommets, ainsi que les arcs dont ils sont soit l'extrémité terminale, soit l'extrémité initiale, on obtient un graphe

$G'' = (X'', U'')$ avec $X' \subset X$ et$U'' \subset U$, appelé *sous-graphe* de $G$.

Ainsi le graphe :

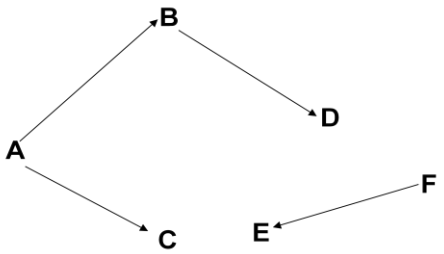



*Figure 3*

est un graphe partiel du graphe de la figure 1 et le graphe :

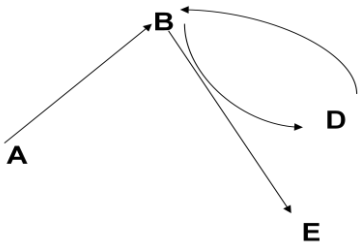



*Figure 4*

est un sous-graphe du graphe de la figure 1, par suppression des sommets $C \; et \; F$.

### 6.2.4. Adjacence, incidence, demi-degrés

On appelle *arcs adjacents,* deux arcs distincts qui ont une extrémité commune.

Deux sommets distincts liés par un arc sont également appelés adjacents.

Un arc, dont les deux extrémités ne sont pas confondues, est incident vers l'intérieur au sommet qui constitue son extrémité terminale, incident vers l'extérieur au sommet qui constitue son extrémité initiale.

*Le demi degré intérieur* d'un sommet $x$ est le nombre d'arcs incidents vers l'intérieur à ce sommet; on le note $d^-(x)$.

*Le demi degré extérieur* d'un sommet $x$ est le nombre d'arcs incidents vers l'extérieur à ce sommet; on note $d^{\mp}(x)$.

Si $d^+(x) = 0 \; et \; d^-(x) \neq 0$ $x$ est appelé une sortie du graphe.

Si $d^-(x) = 0 \; et \; d^+(x) \neq 0$ $x$ est appelé une entrée du graphe

Si $d^-(x) = d^+(x) = 0$ $x$ est un sommet isolé

Si $d(x) = d^-(x) + d^+(x)$ est le nombre d'arcs ayant $x$ pour extrémité

### 6.2.5. Chemins et chaînes

Soit un graphe $G(X, U)$.

Un *chemin* dans le graphe $G$ est constitué par une suite d'arcs $(u_1, u_2, \ldots u_k \ldots)$ telle que tout arc de cette suite, sauf le dernier, a pour extrémité terminale l'extrémité initiale de l'arc suivant.

Un chemin fini dont le sommet terminal coïncide avec le sommet initial est appelé *circuit*.

La *longueur* d'un chemin est le nombre d'arcs qu'il comporte.

Un chemin est dit *simple* s'il ne comporte pas plusieurs fois le même arc; il est *élémentaire* s'il ne passe pas plus d'une fois par le même sommet.

Un chemin qui passe une fois et une seule par tous les sommets du graphe est dit *hamiltonien*; si en plus, c'est un circuit, on dira que c'est un circuit hamiltonien.

Si le nombre de sommets $|x| = n$, tout chemin hamiltonien comporte $n - 1$ arcs, et tout circuit hamiltonien $n$ arcs.

Par exemple sur le graphe ci-dessous :

la suite : $(AE, ED, DE, ED, DC)$ constitue un chemin qui n'est ni simple ni élémentaire. Par contre le chemin de longueur 3 : $(AE, ED, DC)$ est simple et élémentaire ainsi que le circuit $(AE, ED, DC, CA)$, de longueur 4.

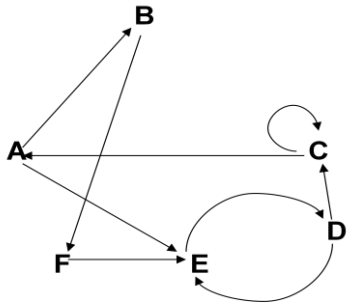



*Figure 5*

Enfin, on peut trouver un chemin hamiltonien : $(AB, BF, FE, ED, DC)$ et un circuit hamiltonien : $(AB, BF, FE, ED, DC, CA)$.

Si l'on étudie à présent des graphes non orientés, on peut définir des notions analogues :

Une *chaîne* est une suite d'arêtes telle que chacune d'elles est rattachée à la précédente par une extrémité et à la suivante par l'autre extrémité (sauf la dernière).

Une chaîne qui se referme sur elle-même est un *cycle*.

Une chaîne peut être *simple* ou *élémentaire*, ou encore *hamiltonienne.* On définira les mêmes termes pour un cycle.

Une chaîne *eulérienne* est une chaîne qui utilise toutes les arêtes du graphe une fois et une seule fois; c'est un cycle eulérien si elle revient à son point de départ.

**Exemple** : si l'on supprime les orientations du graphe précédent, on obtient, en numérotant les arêtes :

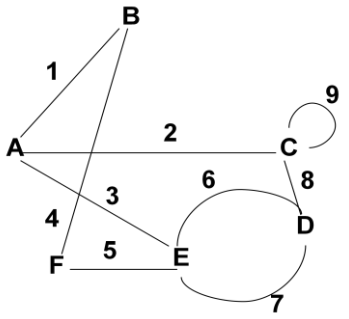



*Figure 6*

La suite (1, 2, 8, 6 ) constitue une chaîne simple et élémentaire.

(3, 6, 8, 2 ) constitue un cycle élémentaire et simple.

(2,8,7,5,4,1) constitue un cycle hamiltonnien.

### 6.2.6. Connexité. Forte connexité

Considérons un graphe *sans orientation*; il est *connexe* si toute paire de sommets distincts est reliée par au moins une chaîne.

Soit la relation :

$xRy$ si $x$ et $y$ sont reliés par au moins une chaîne avec $x$ et $y \in X$; il est clair que $R$ est une relation d'équivalence.

On peut donc décomposer $X$ en classes d'équivalence suivant la relation $R$. Si l'on considère le sous-graphe correspondant à une classe d'équivalence, ce sous-graphe constitue une *composante connexe* de $G$. Si $C_1$ et $C_2$ sont deux composantes connexes, il n'existe aucune arête reliant un sommet de $C_1$ à un sommet de $C_2$ ;

**Exemple** : le graphe ci-dessous a deux composantes connexes.

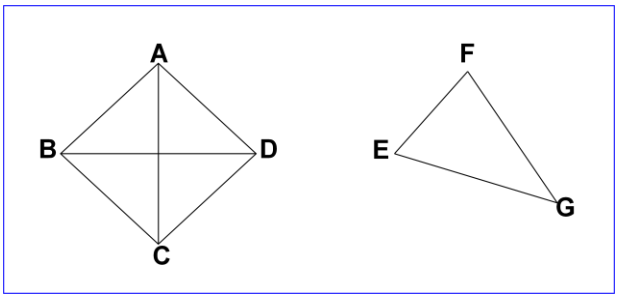



*Figure 7*

La forte connexité, elle, s'applique au concept orienté : un graphe est *fortement connexe* si pour tout couple de sommets $x$ et $y \in X$, il existe un chemin reliant $x$ à $y$ et un chemin reliant $y$ à $x$.

Nous serons amenés à définir d'autres concepts relatifs aux graphes au fur et à mesure que nous en analyserons les applications.

## 6.3. PROBLEMES DE CHEMINS

### *Problème 1*

Soit un graphe $G = (X, U)$ et deux sommets distincts $x$ et $y$ de ce graphe. Existe-t-il un chemin entre $x$ et $y$ ?
Pour résoudre ce problème, nous allons introduire une notion nouvelle : la matrice binaire associée à un graphe, ou matrice d'adjacence.
Les sommets du graphe étant numérotés de 1 à $n$, nous définirons la matrice binaire associée $M(n \times n)$ par ses éléments $a_{ij}$ tels que :
$a_{ij} = 1$ s'il existe un arc $ij$
$a_{ij} = 0$ s'il n'existe pas d'arc $ij$.
Pour le graphe suivant :

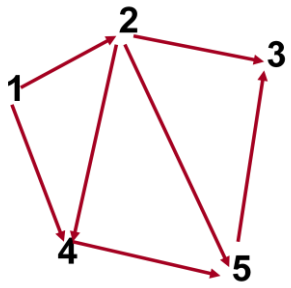



*Figure 8*

on a la matrice binaire associée suivante :

**M=**

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 1 | 0 | 0 |

**Remarque** : la somme des éléments d'une ligne donne le demi-degré extérieur du sommet correspondant à la ligne; de même la somme des éléments d'une colonne fournit le demi-degré intérieur du sommet associé à cette colonne.

Effectuons maintenant le produit de $M$ par elle-même, les opérations correspondantes étant les opérations classiques dans l'anneau des matrices et le corps des réels.

Un élément $a_{ij}^{(2)}$ de la matrice $M^2$ sera donné par :

$$a_{ij}^{(2)} = \sum_k a_{ik}\, a_{kj}$$

Le produit $a_{ik}a_{kj}$ est égal à 1 si $a_{ik}$ et $a_{kj}$ sont égaux à 1, et égal à 0 si l'un au moins des deux termes $a_{ik}$ et $a_{kj}$ est nul.

Or $a_{ik} = 1$ s'il existe un arc reliant $i$ à $k$ et $a_{kj} = 1$ s'il existe un arc reliant $k$ à $j$. En conséquence, si $a_{ik}a_{kj} = 1$ c'est qu'il existe un chemin de longueur 2 entre $i$ et $j$.

Dans ces conditions, $a_{ij}^{(2)}$ représente le nombre de chemins de longueur 2 entre $i$ et $j$. Par exemple :

$$a_{13}^{(2)} = 0\times 0 + 1\times 1 + 0\times 0 + 1\times 0 + 1\times 0 + 1\times 0 = 1$$

Il y a un chemin de longueur 2 entre 1 et 3 : c'est le chemin (1, 2, 3).

La matrice $M^2$ complète est :

$M^2=$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 2 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |

$M^3=$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 2 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |

Un raisonnement analogue nous montre que $M^3$ fournit nombre de chemins de longueur 3 existant entre chaque paire de sommets.

Enfin $M^4$, qui fournit les chemins de longueur égale à 4 est :

$M^4=$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |

$M^5= 0$

On vérifiera sans peine que $M^5 = 0$, ce qui est évident sur le graphe en question.

D'une façon générale, si $M^P = 0$ à partir d'un certain exposant, cela veut dire que le graphe est sans circuit. En effet, si le graphe avait un circuit, il est évident que pour tout $k$ entier on pourrait choisir deux sommets du circuit tels qu'il existe un chemin de longueur $k$ entre ces deux sommets, donc que $M^k \neq 0 \; \forall k$, ce qui est incompatible avec $M^P = 0$ à partir d'un certain $p$.

Réciproquement, si le graphe est sans circuit, et puisqu'il est fini, un chemin est au plus de longueur $n - 1$, si $n = |X|$ ; on est alors sûr que pour $p \geq n \;\; M^P = 0$.

L'examen successif des matrices, $M, M^2, \ldots M^n$ ... répond au problème 1. Pour un couple de sommets $ij$ si un $a_{ij}(n)$ est non nul, c'est qu'il existe un chemin menant du sommet $i$ au sommet $j$.

Cependant, toujours dans le cadre du problème 1, on préfère utiliser la matrice $M$ d'une autre manière.

Les éléments de $M$ seront à présent considérés comme les éléments du calcul *booléen* ; ces éléments (0 et 1) peuvent, on le sait, se combiner suivant les deux opérations : addition (+) et multiplication (.) booléennes suivant les tables :

$0 + 0 = 0 \qquad 0 + 1 = 1 + 0 = 1 \qquad 1 + 1 = 1$

$0.0 = 0 \qquad 0.1 = 1.0 = 1 \qquad 1.1 = 1$

Soit alors la matrice $M$, considérée comme formée d'éléments booléens et faisons le produit $M \times M$ où l'opération « produit matriciel » garde la même signification qu'auparavant, mais où les opérations sur les $a_{ij}$ sont booléens. Soit alors :

$$a_{ij}^{(2)} = a_{i1}.a_{1j} + a_{12}.a_{2j} + \ldots\ldots + a_{ik}.a_{kj} + \ldots\ldots + a_{in}.a_{nj}$$

Il est clair que si $a_{ij}^{(2)} = 1$, cela signifie qu'il existe au moins un chemin de longueur 2 entre $i$ et $j$ et $a_{ij}^{(2)} = 0$ signifie qu'il n'existe pas de chemins de longueur 2 entre $i$ $et$ $j$.

De même $M^p$, calculée sur la base des opérations ci-dessus permet de savoir s'il existe un chemin de longueur $p$ entre un sommet quelconque et un autre.

Si alors, on fait la somme $\widehat{M} = I + M + M^2 + M^3 + \ldots M^p + \ldots$

avec $I$ : matrice unité,

la matrice $\widehat{M}$ obtenue permet de savoir s'il existe un chemin entre un sommet quelconque et un autre (avec la convention: il existe un chemin de longueur 0 entre $x$ et ).

Par exemple si l'on prend une ligne de $\widehat{M}$, associée a un sommet $x$ quelconque, cette ligne fournit tous les sommets que l'on peut atteindre par un chemin partant de $x$.Cet ensemble de sommets s'appelle *la fermeture transitive de* $x$. Si l'on se réfère à l'application définie plus haut, la fermeture transitive de $x$ peut s'écrire:

$\widehat{\Gamma} = xU\Gamma(x)U\Gamma^2(x) \ldots U\Gamma^k \ldots$ avec

$$\Gamma^k(x) = \Gamma\underbrace{(\Gamma(\Gamma \ldots \Gamma(x)) \ldots)}_{k\ fois}$$

Pratiquement, pour calculer $\widehat{M}$ , il suffit d'effectuer la somme ci-dessus jusqu'à $M^n$. En effet :

a) si le graphe est sans circuit, $M^n = M^{n+1} \ldots M^{n+k} = 0$

b) si le graphe comporte des circuits, il existe des $a_{ij}$ non nuls pour $p \geq n$ ; mais cela signifie qu'il existe entre $i$ et $j$ un chemin de longueur $p$ qui emprunte au moins un circuit.

En enlevant tous les circuits de ce chemin, on obtient un chemin reliant $i$ à $j$ de longueur inférieure à $n$. Donc il existe $k < n$ tel que $a_{ij}^k = 1$ ; et l'élément $a_{ij}^{(p)}$ avec $p \geq n$ n'ajoute rien à $\widehat{M}$.

On peut donc avoir $\widehat{M}$ par :

$\widehat{M} = I \dot{+} M \dot{+} M^2 \dot{+} \ldots M^{n-1}$

En fait, on peut calculer $\widehat{M}$ d'une autre façon encore : il est facile de voir que

$(I + M)^p = I \dot{+} M \dot{+} M^2 \dot{+} \ldots M^p$ pour tout $p$

En effet , cette expression est vraie pour $p = 1$

Supposons qu'elle le soit pour $p - 1$

$(I \dot{+} M)^{p-1} = I \dot{+} M \dot{+} M^2 \dot{+} \ldots M^{p-1}$ $alors$

$(I \dot{+} M)^{p-1}(I \dot{+} M) = (I + M)^p = I \dot{+} M \dot{+} M^2 \dot{+} \ldots M^{p-1} \dot{+} M \dot{+} M^2 \ldots \dot{+} M^P$

Mais comme $M^k \dot{+} M^k = M^k$, alors

$(I \dot{+} M)^p = I \dot{+} M \dot{+} M^2 \dot{+} \ldots \dot{+} M^p$

On en conclut :

$$\widehat{M} = (I \dot{+} M)^{n-1}$$

Dans la pratique, pour calculer $\widehat{M}$, on prendra $(I \dot{+} M)$ ; on calculera $(I \dot{+} M)^2$, puis $(I \dot{+} M)^4 \dot{+} \ldots + (I \dot{+} M)^{2k}$, en s'arrêtant au premier $k$ entier tel que : $2^n \geq n - 1$.

Sur le graphe de la figure 7, il suffit de calculer $(I + M), (I + M)^2 et\ (I + M)^4$. On trouve

$\widehat{M}=$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 0 | 1 | 0 | 1 |

Par l'intermédiaire de cette matrice, on pourrait isoler les sous-graphes fortement connexes d'un graphe donné, en repérant dans $M$ des matrices carrées remplies uniquement de 1 et dont les lignes et les colonnes correspondent aux mêmes sommets. Sur l'exemple choisi, on constate que cette opération n'est pas possible.

***Problème 2 : Le problème du plus court chemin.***

Soit $G = (X, U)$, un graphe dans lequel chaque arc est valué, c'est à dire qu'à chaque arc $u$ est associé un nombre entier $l(u)$.Trouver un chemin μ allant d'un sommet $a$ donné à un sommet $b$ donné tel que la valeur totale de ce chemin

$\sum_{u \in \mu} l(u)$ soit minimale

Il existe à l'heure actuelle de nombreux algorithmes traitant ce problème. Nous n'exposerons ici que les plus connus d'entre eux.

Nous distinguons deux cas: celui où $l(u) \geq 0$ pour tout $u$ (qui est à dire vrai le cas le plus courant en recherche opérationnelle) et le cas $l(u)$ où est de signe quelconque.

**Algorithme 1 :** $l(u) \geq 0 \; \forall u$ **Algorithme de Moore-Dijkstra.**

Cet algorithme permet de déterminer le plus court chemin entre un sommet particulier d'un graphe $G = (X, U)$, aux valeurs d'arcs positives ou nulles, et tous les autres sommets.

Par commodité, nous appellerons les sommets du graphe $X = \{1, 2 \ldots n\}$.

Nous nommerons $l_{ij}$ la valeur de l'arc reliant $i$ à $j$ .S'il n'y a pas d'arc entre $i$ et $j$, $l_{ij} = +\infty$ .

Soit $\lambda_i^*$ la valeur du chemin le plus court entre $1 \; et \; i$ . L'algorithme consiste à trouver $\lambda_i^*$ en affectant à chaque sommet $i$ une valeur provisoire $\lambda_i$ qui se stabilise en un nombre d'itérations finies à la valeur $\lambda_i^*$.

Pour cela, nous séparerons, à chaque itération, l'ensemble des sommets $X$ en deux parties $S$ et $X - S$, avec

$S$ : ensemble des $i$ tels que $\lambda_i = \lambda_i^*$ avec $1 \in S$. ( Pour ces sommets on a trouvé la valeur du plus court chemin).

$X - S$ : on n'est pas certain pour ces sommets que $\lambda_i$ mesure la valeur du chemin le plus court entre $1 \; et \; i$ , mais on fait systématiquement :

$$\lambda_i = \min_{k \in S}[\lambda_k^* + l_{ki}] \qquad (1)$$

Donc, pour $i \in X - S, \lambda_i$ donne la valeur du chemin le plus court entre $1 \; et \; i$, *tel que tous les sommets de ce chemin excepté i sont dans S*.

L'algorithme repose alors sur le lemme suivant:

**Lemme**

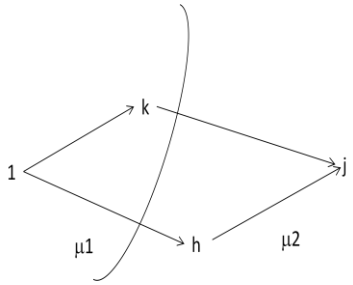
Soit $j \in X - S \; tel \; que$

$\lambda_j = min \lambda_i$

$i \in X - S$

Alors $\lambda_j = \lambda_j^*$

**Démonstration :**




Tout d'abord, $\lambda_j$ a été calculé par la formule (1) à partir d'un sommet $k \in S$. Le fait que $\lambda_j$ est fini indique qu'il y a un chemin entre $1$ $et$ $j$ passant par $k$ ( Si $\lambda_j$ n'est pas fini, il n'existe pas de chemin entre les sommets de $S$ et ceux de $X - S$, et l'algorithme s'arrête). Pour montrer que ce chemin est le chemin de valeur minimale entre$1$ $et$ $j$, prenons un autre chemin : il se décompose en deux sous-chemins, l'un $\mu_1$ jusqu'au premier sommet de $X - S$ rencontré, soit $h$ et l'autre $\mu_2$ entre $h$ $et$ $j$. Par construction de $\lambda_h$ par la formule (1), on a : valeur de $\mu_1 \geq \lambda_h$, et puisque $j$ est caractérisé par le minimum des $\lambda_i$ sur $X - S$.

Valeur de $\mu_1 \geq \lambda_h \geq \lambda_j$

Par ailleurs, puisque les valeurs des arcs sont positives ou nulles

Valeur de $\mu_2 \geq 0$ au total

Valeur de $\mu \geq \lambda_j$

$\mu$ étant quelconque, $\lambda_j$ est la valeur du chemin le plus court entre $1$ à $j$.

D'où l'algorithme :

$\underline{Etape\ a.\ Initialisation}$

$$S = \{1\} \quad X - S = \{2 \ldots . n\}$$

$$\lambda_1 = 0 \quad \lambda_i = l_{1i} \quad \text{pour } i \neq 1$$

$$(éventuellement\ l'infini\ s'il\ n'y\ a\ pas\ d'arc\ entre\ 1\ et\ j)$$

$\underline{Etape\ b}$

$$Sélectionner\ j \in X - S\ tel\ que$$
$$\lambda_j = \min_{i \in X - S} \lambda_i$$

$$S\ devient\ S + \{j\}$$

$$et\ X - S \qquad X - S - \{j\}$$

$Si\ S = X\ ;\ fin, sinon\ aller$ à $l'étape\ c$

Etape c

Pour tout $i \in \Gamma_j\ et\ \in X - S\ faire$

$$\lambda_i = min\ (\lambda_i, \lambda_j + l_{ji})$$

$(il\ suffit\ en\ effet\ de\ corriger\ les\ \lambda_i, si\ besoin\ est, des\ suivants\ de\ j$

$Retourner$ à $l'étape\ b.$

Lorsque $S = X$, tous les $\lambda_i$ sont égaux aux $\lambda_i^*$ et donnent donc les valeurs du plus court chemin entre $1\ et\ i$ .Pour avoir le chemin le plus court lui même il faut repérer à partir de quel sommet $j_1$ la valeur $\lambda_i^*$ a été calculée, puis le sommet $j_2$ à partir duquel la valeur $\lambda_2^*$ a été calculée etc.

**Exemple :** Soit le graphe suivant, où les valeurs des arcs sont indiquées sur ces derniers (figure 9)

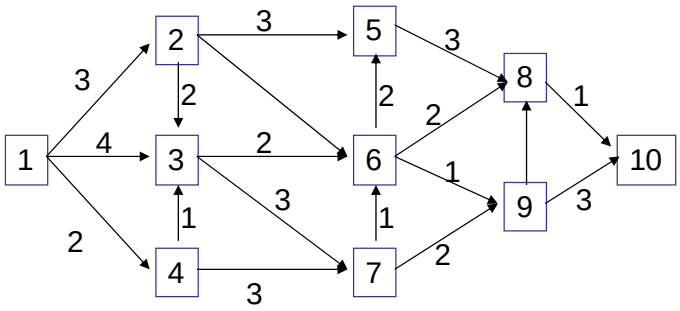



*Figure 8*

L'algorithme de Moore-Dijkstra donne les opérations suivantes:

1) $S = \{1\}\quad \lambda_2 = 3\quad \lambda_3 = 4\quad \lambda_4 = 2\quad \lambda_5 = \lambda_6 = \cdots \lambda_{10} = +\infty$

2) $Min_{X-S}\, \lambda_i \rightarrow 4.\ S = \{1,4\}$

$\lambda_3 = Min\ (4, 2 + 1) = 3$

$\lambda_7 = 5$

(le reste inchangé)

3) $Min_{X-S}\, \lambda_i \rightarrow 2.\ S = \{1,2,4\}$

$\lambda_5 = 6$

$\lambda_6 = 7$

(le reste inchangé)

4) $Min_{X-S}\,\lambda_i \rightarrow 3.\ S = \{1,2,3,4\}$

$\lambda_6 = 5$

(le reste inchangé)

5) $Min_{X-S}\,\lambda_i \rightarrow 6.\ S = \{1,2,3,4,6\}$

$\lambda_8 = 7$

$\lambda_9 = 6$

6) $Min_{X-S}\,\lambda_i \rightarrow 7.\ S = \{1,2,3,4,6,7\}$

$\lambda$ inchangé

7) $Min_{X-S}\,\lambda_i \rightarrow 5.\ S = \{1,2,3,4,5,6,7\}$

$\lambda$ inchangé

7) $Min_{X-S}\,\lambda_i \rightarrow 5.\ S = \{1,2,3,4,5,6,7\}$

$\lambda$ inchangé

8) $Min_{X-S}\,\lambda_i \rightarrow 9.\ S = \{1,2,3,4,5,6,7,9\}$

$\lambda_{10} = 9$

9) $Min_{X-S}\,\lambda_i \rightarrow 8.\ S = \{1,2,3,4,5,6,7,8,9\}$

$\lambda_{10} = 8$

10) $Min_{X-S}\,\lambda_i \rightarrow 10.\ S = \{1,2,3,4,5,6,7,8,9,10\}$

$\lambda_i =$ inchangé

La valeur du plus court chemin entre 1 et 10 est égale à 8.

$\lambda_{10} = 8$ a été calculé par le choix de 8

$\lambda_8 = 7$ a été calculé par le choix de 6

$\lambda_6 = 5$ a été calculé par le choix de 3

$\lambda_3 = 3$ a été calculé par le choix de 4

$\lambda_4 = 2$ a été calculé par le choix de 1

Le chemin optimal est donc, 1, 4, 3, 6, 8, 10.

On a vu que la démonstration de l'algorithme précédent suppose que les valeurs des arcs sont positives ou nulles. Lorsque ce n'est pas le cas, c'est-à-dire que les valeurs des arcs sont de signe quelconque, on utilise d'autres algorithmes valables pour les graphes qui

n'ont pas de circuit de valeur négative (s'il existe de tels circuits en effet, il n'existe pas entre certains sommets de chemin de valeur minimale, puisqu'en empruntant une infinité de fois de tels circuits, on obtient des valeurs de chemin infiniment négatives).

Une problématique se pose de façon constante sur les algorithmes : c'est celle de l'estimation de leur temps de calcul. On l'a déjà rencontrée à propos de la programmation linéaire (chapitre 4) lorsque nous avons introduit les méthodes de point intérieur.

Qu'en est-il de l'algorithme de chemin de valeur minimale que nous venons d'exposer?

D'une façon générale, on définit la complexité d'un algorithme comme un majorant du nombre d'opérations élémentaires qu'il doit effectuer pour trouver la solution. Par opération élémentaire, on entend toute opération simple effectuée sur deux nombres, réels ou entiers, comme la substitution, la comparaison, ou encore les opérations arithmétiques.

La rapidité de l'algorithme dépend évidemment de la taille du problème (définie grosso modo comme le nombre de caractères élémentaires qui permettent d'entrer les données dans un ordinateur et résumées par des caractéristiques simples du problème, comme le nombre de sommets et d'arcs dans un graphe ou le nombre de contraintes et de variables dans un programme linéaire) et des données pour les variables utilisées (les valeurs des arcs par exemple ou les valeurs des coefficients des contraintes et du second membre pour les PL). On appelle instance cet ensemble de variables. La complexité est une fonction de la taille, représentant le nombre maximal d'opérations élémentaires effectuées sur l'ensemble des instances possibles, pour cette taille. En général, cette fonction est impossible à connaître, mais on peut lui trouver un majorant qui donne des indications précieuses sur la rapidité de l'algorithme. Si par exemple, on démontre que le nombre d'opérations en question est borné par une fonction polynomiale de la taille du problème, alors on dira que l'algorithme est polynomial; dans le cas contraire, il est dit exponentiel.

L'algorithme de Moore-Dijkstra est polynomial.

En effet, à chaque itération $k$, on doit d'abord trouver le minimum de $k$ valeurs. Ceci peut se faire pour $k$ comparaisons. Comme $k$ varie de 1 à $n-1$, la somme de ces $k$ comparaisons est de l'ordre de $n^2$(on utilisera la notation O($n^2$)).

Ensuite, si l'on appelle $d(j)$ le demi-degré extérieur du sommet $j$ trouvé par l'opération précédente, il faut faire $d(j)$ opérations et comparaisons. On explore en fait, sur l'ensemble des itérations, l'ensemble des $m$ arcs. Cette phase est donc en $O(m)$.

Au total, la complexité est de l'ordre $O(n^2) + O(m)$, et on a bien affaire à un algorithme polynomial.

Cette notion est importante, car il est bien évident que les algorithmes polynomiaux donnent une meilleure assurance de rapidité de traitement que les algorithmes qui ne le sont pas. On a déjà donné pour les PL des exemples de temps de traitement rédhibitoires pour un algorithme exponentiel comme le simplexe, même avec les ordinateurs les plus performants.

Voici l'un des plus anciens et des plus célèbres algorithmes, mais non le plus performant dû au mathématicien américain Ford.

**Algorithme 2:**

**Algorithme de Ford**

a) Si $n$ est le nombre de sommets du graphe, numéroter les sommets $1, 2, \ldots n$, de façon quelconque.

b) Affecter provisoirement à tout sommet *i* un poids $\lambda_i$ égal à *0* si $i = 1$ et à $+\infty$ si $i \neq 1$

c) Chercher un arc $\overline{ij}$ tel que $\lambda_j - \lambda_i > l_{ij}$ ; on remplace alors $\lambda_j$ par $\lambda_j = \lambda_i + l_{ij}$

d) Reprendre en c) jusqu'à ce qu'aucun arc ne permette plus de diminuer les $\lambda_i$.

Cette procédure fournit bien le ( ou les ) chemins de valeur minimale.

En effet, il est évident que les $\lambda_i$ sont des majorants des valeurs des plus courts chemins entre $1 \; et \; i$ . Comme le graphe est sans circuit négatif, ce plus court chemin entre 1 et $i$, s'il existe, a une valeur finie.

Et comme, d'après l'étape $d$, les $\lambda_i$ sont continument décroissantes tout en restant positives, les $\lambda_i$ tendent vers une limite.

Montrons que cette limite $\lambda_i$ est bien la valeur du plus court chemin entre $1 \; et \; i$. Il existe un sommet $i_1$ tel que :

$\lambda_j - \lambda_{i1} > l_{i1j}$

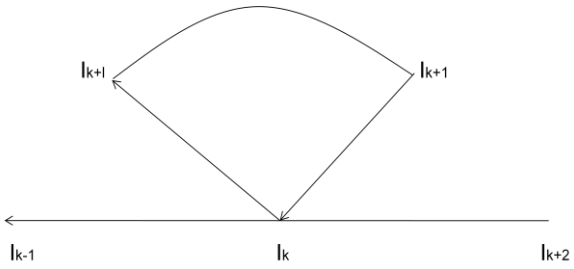
($i_1$est le dernier sommet utilisé pour diminuer $\lambda_j$. De même il existe un sommet $i_2$ tel que :

$\lambda_{j1} - \lambda_{i2} > l_{i2j1}$

etc.

On constitue ainsi une suite $i, i_1, i_2, i_3$ ... de sommets tels que $\lambda_{ij} - \lambda_{ij+1} = l_{ij+1ij}$ .

Montrons que s'il n'existe pas de circuit de valeur nulle, les sommets de la suite précédente sont tous distincts. En effet, si l'on trouve un sommet $i_k$ deux fois, c'est que l'on a la disposition suivante :




Pour le circuit$i_k, i_{k+l}, i_{k+1}, i_k,$ on a :

$\lambda_{ik+l} - \lambda_{ik} = l_{ikik+l}$

$\lambda_{ik} - \lambda_{ik+1} = l_{ik+1ik}$

Soit en sommant :

$0 = l_{ikik+l} + \cdots + l_{ik+1ik}$

Ce que l'on a exclu. De là, il résulte que, comme la suite $i, i_1, i_2$ ... a tous ses éléments distincts et comme le nombre de sommets du graphe est fini, on trouve au bout d'un certain temps le sommet 1. On a alors :

$\lambda_i - \lambda_{i_1} = l_{i_1 i_2}$

$\lambda_{i_1} - \lambda_{i_2} = l_{i_2 i_1}$

$\lambda_{i_s} - \lambda_1 = l_{1 i_s}$

…

Soit en sommant

$$\lambda_i - \lambda_1 = \lambda_i = \ell_{1 i_s} + \ell_{i_s i_{s-1}} + \ldots\ldots + \ell_{i_1 i}$$

$\lambda_i$ mesure la valeur du chemin $1, i_s, i_{s-1}, \ldots i_2, i_1, i$ trouvé par cette méthode.

Montrons que ce chemin est bien le chemin le plus court. Soit en effet un autre chemin

$\mu\ 1k_1\, k_2 \ldots k_l i$

Puisqu'on a appliqué l'algorithme, on a obligatoirement

$\lambda_{k_1} - \lambda_1 \leq l_{1k_1}$

$\lambda_i - \lambda_{k_l} \leq l_{kl_i}$

Soit en sommant

$$\lambda_i \leq \ell_{1k_\ell} + \ldots\ldots + \ell_{k_\ell i}$$

$$\lambda_i \leq \sum_{u \in \mu} \ell_u$$

La valeur du chemin $\mu$ est donc supérieure ou égale à la valeur $\lambda_i$ du chemin trouvé par la procédure ci-dessus.

**Remarque** : il peut y avoir naturellement plusieurs chemins de valeur minimale.

Pratiquement, on examine les $\lambda_i$ dans l'ordre de numérotation des sommets et en testant la règle c) sur les arcs issus de chaque sommet. Dans la mesure où la numérotation des sommets est arbitraire, on peut avoir des arcs $x_i x_j$ avec $i > j$. Alors, une valeur $\lambda_i$ peut

être modifiée plusieurs fois, ainsi que toutes les valeurs $\lambda_k$ avec $k > j$. Nous allons constater une telle situation sur l'exemple précédent :

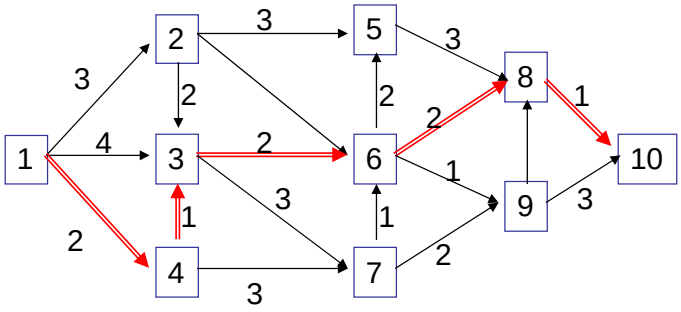



*Figure 10*

Au début :

$\lambda_1 = 0 \quad \lambda_2 = \quad \lambda_3 = \quad \lambda_4 \ldots\ldots = \lambda_{10} \quad = +\infty$

Examinons 1 : on modifie $\lambda_2,\ \lambda_3, \lambda_4$

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 4 \quad \lambda_4 = 2 \quad \lambda_5 = \lambda_6 = \lambda_7 = \lambda_8 = \lambda_9 = \lambda_{10} = \infty$

À partir de 2 ($\lambda_5$ $et$ $\lambda_6$ diminuent)

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 4 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 7 \quad \lambda_7 = \lambda_8 = \lambda_9 = \lambda_{10} = \infty$

À partir de 3 : ($\lambda_6$ diminue, et $\lambda_7$ aussi)

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 4 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 6 \quad \lambda_7 = 7 \quad \lambda_8 = \lambda_9 = \lambda_{10} = \infty$

À partir de 4 : ($\lambda_3$ diminue, et $\lambda_7$)

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 3 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 6 \quad \lambda_7 = 5 \quad \lambda_8 = \lambda_9 = \lambda_{10} = \infty$

Comme $\lambda_3$ a diminué, nous sommes obligés de remonter à 3.

À partir de 3 :

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 3 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 5 \quad \lambda_7 = 5 \quad \lambda_8 = \lambda_9 = \lambda_{10} = \infty$

À partir de 4 : inchangé.

À partir de 5 :

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 3 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 5 \quad \lambda_7 = 5 \quad \lambda_8 = 9 \quad \lambda_9 = \lambda_{10} = \infty$

À partir de 6 :

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 3 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 5 \quad \lambda_7 = 5 \quad \lambda_8 = 7 \quad \lambda_9 = 6 \quad \lambda_{10} = \infty$

À partir de 7 : inchangé

À partir de 8 :

$\lambda_1 = 0 \quad \lambda_2 = 3 \quad \lambda_3 = 3 \quad \lambda_4 = 2 \quad \lambda_5 = 6 \quad \lambda_6 = 5 \quad \lambda_7 = 5 \quad \lambda_8 = 7 \quad \lambda_9 = 6 \; \lambda_{10} = 8$

À partir de $x_8$ : inchangé.

Le chemin de valeur minimale est donc : $(1, 4, 3, 6, 8, 10)$

Il est indiqué en trait renforcé sur la figure. Il est obtenu en remontant à partir de 10 comme l'indique la procédure.

$$\lambda_{10} - \lambda_8 = 1 = 1_{810}$$
$$\lambda_8 - \lambda_6 = 2 = 1_{68}$$
$$\lambda_6 - \lambda_3 = 2 = 1_{36}$$
$$\lambda_3 - \lambda_4 = 1 = 1_{43}$$
$$\lambda_4 - \lambda_1 = 2 = 1_{14}$$

A cause de la possibilité de retourner en arrière, la complexité de cet algorithme est en $O(n^2 m)$. ($n$ nombre de sommets, $m$ nombre d'arcs)

Remarquons qu'il est plus simple de repérer, lorsqu'un $\lambda_i$ diminue, quel est le sommet *i* qui a contribué à cette diminution. On a alors le chemin cherché immédiatement.

Les retours en arrière tels que ceux que l'on a constatés sur l'exemple traité peuvent être très gênants au niveau du temps de calcul. Pour pallier cet inconvénient, il est toujours possible de réorganiser le graphe (à condition qu'il soit *sans circuit*) de façon que ces retours en arrière ne soient pas possibles.

Par exemple le graphe traité de la figure 8 peut se représenter de la façon suivante :

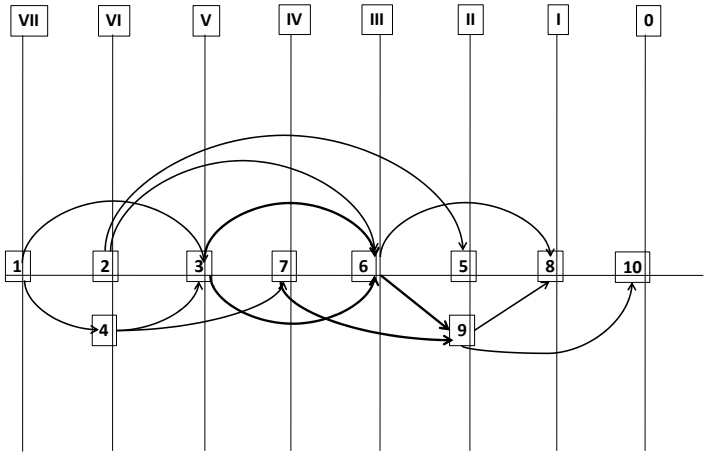



*Figure 10*

Cette représentation est dite « en niveaux » (numéroté ici de 0 à VII de droite à gauche) : chaque sommet $x_i$ d'un niveau donné est tel que tous les sommets $x_j$ qui le suivent immédiatement (existence d'un arc $x_i x_j$) se trouvent dans les niveaux de numéro inférieur et au moins un dans le niveau de numéro immédiatement inférieur.

On renumérote alors les sommets du graphe (de haut en bas et de gauche à droite). On obtient :

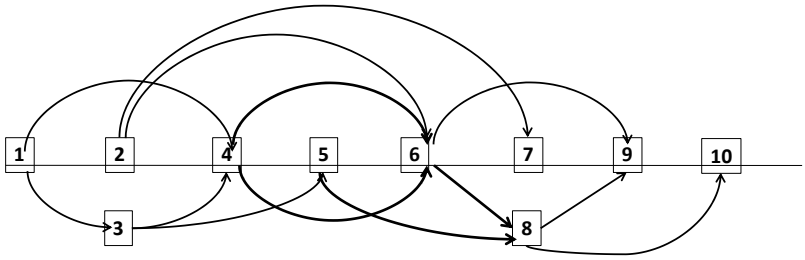



*Figure 11*

Sur un tel graphe, on n'a pas d'arc $ij$ avec $i > j$. En conséquence il est exclu que l'algorithme de Ford conduise à des retours en arrière.

La méthode utilisée pour réécrire le graphe par niveaux est remarquablement simple. Il suffit de prendre la matrice binaire associée au graphe, c'est-à-dire si l'on considère l'exemple traité :

**Matrice binaire associée au graphe**

| Niveaux | Sommets | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Demi-degrés ext. | Successifs | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 1 | | 1 | 1 | 1 | | | | | | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 0 |
| 6 | 2 | | | 1 | | 1 | 1 | | | | | 3 | 3 | 3 | 2 | 1 | 1 | 0 | |
| 5 | 3 | | | | | | 1 | 1 | | | | 2 | 2 | 2 | 2 | 1 | 0 | | |
| 6 | 4 | | | 1 | | | | 1 | | | | 2 | 2 | 2 | 2 | 2 | 1 | 0 | |
| 2 | 5 | | | | | | | | 1 | | | 1 | 1 | 0 | | | | | |
| 3 | 6 | | | | | 1 | | | 1 | 1 | | 3 | 3 | 1 | 0 | | | | |
| 4 | 7 | | | | | | 1 | | | 1 | | 2 | 2 | 2 | 1 | 0 | | | |
| 1 | 8 | | | | | | | | | | 1 | 1 | 0 | | | | | | |
| 2 | 9 | | | | | | | | 1 | | 1 | 2 | 1 | 0 | | | | | |
| 0 | 10 | | | | | | | | | | | 0 | | | | | | | |

La colonne immédiatement à droite de la matrice représente les demi-degrés extérieurs de chaque sommet. Dans la classe de niveau 0, on place les sommets de demi-degré extérieur nul (ici 10). On barre les lignes et colonnes correspondant à ces sommets (c'est-à-dire qu'on enlève au graphe le sous-graphe constitué par ces sommets) et on recalcule les demi-degrés extérieurs sur la nouvelle matrice. Dans la classe de niveau 1, on place les sommets de demi-degrés extérieurs nuls dans la nouvelle matrice (ici 8), on barre les lignes et les colonnes correspondantes, on recalcule les demi-degrés extérieurs, etc. On trouve bien ainsi sur le graphe considéré les 8 classes de la figure 10.

**Remarque** : l'algorithme de Ford peut être facilement adapté au cas où l'on recherche *dans un graphe sans circuit* le - ou les - chemins de valeur maximale (et non plus le

chemin de valeur minimale). Il suffit en effet d'affecter à chaque sommet un poids $\lambda_i$ égal à 0 et de recalculer les $\lambda_j = \lambda_i + l(ij)$.

Si $\lambda_i + l(x_i x_j) > \lambda_j \; ou \; \lambda_j - \lambda_i \; < l(x_i x_j)$

**Algorithme 3 : Algorithme de Bellmann-Kalaba**

Cet algorithme va nous donner l'occasion d'utiliser le principe d'optimalité de Bellmann, qui est à la base d'une technique d'optimisation appelée « programmation dynamique », et dont la simplicité n'empêche pas la fécondité.

Dans le cas qui nous intéresse, c'est-à-dire la détermination du chemin de valeur minimale entre $1$ à $n$. Le principe d'optimalité s'énonce de la façon suivante : soit $1, k_1, \ldots k_p \; n$ le chemin de valeur minimale entre $1 \; et \; n$ ; alors, pour tout $k_i \in k_1 \ldots k_p$ le chemin $k_i, k_{i+1} \ldots k_p \; n$ est le chemin de valeur minimale entre $k_i \; et \; n$.

Plus synthétiquement : *tout sous-chemin d'un chemin optimal est optimal.*

Ce principe est parfaitement évident : si en effet il existait un chemin $k_i, k_j \ldots k_l$ meilleur que $k_i, k_{i+1} \ldots k_p \; n$ alors le chemin $lk_1 n, k_i k_j \ldots k_l \; n$ serait meilleur que $lk_1, k_i k_{i+1} \ldots k_p \; n$, qui a été pourtant supposé optimal.

Plus précisément, dans l'algorithme de Bellmann-Kalaba, on utilise ce principe de la façon suivante :

soit un sommet $i$ du graphe, et considérons l'ensemble $E_i$ des suivants $j$ de $i$ (j$\in \Gamma(i)$). Prenons maintenant le chemin de valeur minimale entre $i \; et \; n$ *de longueur inférieure ou égale à* $p$. Soit $i \; i_1 \; i_2 \ldots i_l \; n$ ce chemin ; $i_1 \in E_i$. Il est évident que le chemin $i \; i_1 \; i_2 \ldots i_l \; n$ est le chemin reliant $i_1$ à $n$ de *longueur inférieure ou égale à* $p - 1$ et de valeur minimale.

Pratiquement on opère ainsi : on appelle $l_{ij}$ la valeur de l'arc reliant $i$ à $j$, si cet arc n'existe pas, $l_{ij} = +\infty$. On appelle $\beta_i^{(p)}$la valeur du chemin de valeur minimale de longueur inférieure ou égale à $p$ reliant $i$ à $n$.

À la première étape on a :

$\beta_i^{(p)} = l_{in}$ avec

$\beta_n^{(p)} = 0$

puis on calcule pour $p = 2, 3 \ldots et \; i = 0, 1, \ldots n - 1 \; \beta_i^{(p)}$ par la formule :

$\beta_i^{(p)} = Min(l_{ij} + \beta_j^{(p-1)})$

$j \neq i$

en posant pour tout $p$

$\beta_n^{(p)} = 0$

Comme le chemin de valeur minimale entre un sommet $i$ quelconque et $n$ a une longueur inférieure ou égale à $n - 1$, on arrête les calculs lorsque

$\beta_i^{(p)} = \beta_i^{(p-1)} \; \forall_i$

Sur l'exemple de la figure 8, cette méthode donne les vecteurs $\beta$ suivants :

| Sommets | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |
| Itérations | | | | | | | | | | |
| p = 1 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 1 | 3 | 0 |
| p = 2 | ∞ | ∞ | ∞ | ∞ | 4 | 3 | 5 | 1 | 3 | 0 |
| p = 3 | ∞ | 7 | 5 | 8 | 4 | 3 | 4 | 1 | 3 | 0 |
| p = 4 | 9 | 7 | 5 | 6 | 4 | 3 | 4 | 1 | 3 | 0 |
| p = 5 | 8 | 7 | 5 | 6 | 4 | 3 | 4 | 1 | 3 | 0 |
| p = 6 | 8 | 7 | 5 | 6 | 4 | 3 | 4 | 1 | 3 | 0 |
| p = 7 | 8 | 7 | 5 | 6 | 4 | 3 | 4 | 1 | 3 | 0 |
| p = 8 | 8 | 7 | 5 | 6 | 4 | 3 | 4 | 1 | 3 | 0 |

On pouvait s'arrêter à $p = 6$

Pour trouver le chemin de valeur minimale, on cherche le sommet $i_1$tel que :

$$\beta_1^{(5)} - \beta_{i_1}^{(5)} = \ell_{1i_1}$$

puis le sommet $i_1$ tel que :

$$\beta_{i_1}^{(5)} - \beta_{i_2}^{(5)} = \ell i_1 i_2$$

Etc.

L'algorithme de Bellmann-Kalaba est souvent plus performant que l'algorithme de Ford au niveau du temps de calcul sur ordinateur par le fait des retours en arrière fréquents dans l'algorithme de Ford. Sa complexité est en $O(mn)$.

**Méthode matricielle**

C'est une généralisation de l'algorithme précédent (Bellmann-Kalaba) ; c'est une bonne méthode lorsque :

a) il s'agit d'obtenir les distances minimales entre chaque couple de sommets de graphe;

b) il n'est pas utile de conserver les itinéraires de valeur minimale entre les couples de sommets, mais seulement les valeurs de ces itinéraires.

Cette méthode s'appuie sur une opération matricielle d'un type particulier : pour un graphe quelconque, on considère la matrice $M_1$ constituée par les valeurs $V_{ij}^{(1)} = l_{ij}$ des arcs du graphe (s'il n'y a pas d'arcs reliant $i$ à $j$, $V_{ij}^{(1)} = +\infty$). Soit alors la matrice :

$$M_2 = M_1 \oplus M_1$$

L'opération $\oplus$ ayant le sens suivant :

Si $V_{ij}^{(2)}$ est l'élément de $M_2$ :

$V_{ij}^{(2)} = min_k(V_{ik}^{(1)} + V_{kj}^{(1)}$

On voit que $V_{ij}^{(2)}$ représente la valeur du chemin de valeur minimale parmi les chemins de longueur inférieure ou égale à 2 reliant $i$ à $j$. Il suffit alors de composer :
$M_2 \oplus M_2 = M_4$

De la même façon pour obtenir la valeur des chemins minimaux et de longueur inférieure ou égale à 4 reliant deux sommets quelconques. On constitue en composant :
$M_k \oplus M_k = M_{2k}$

En principe, on s'arrête lorsque $k$ atteint ou dépasse $n, n$ étant étant le nombre de sommets. En fait, on peut voir facilement qu'il suffit de s'arrêter lorsque $M_k = M_{k-1}$.

On remarquera que cette méthode impose en principe de réserver en mémoire d'ordinateur la place suffisante pour deux matrices $(n \times n)$ : pour un $k$ donné $M_k$ et $M_{2k}$. Pour des graphes importants, cette nécessité est susceptible de créer quelque limitation au niveau de l'utilisation de cette méthode. On remarquera néanmoins que lorsque l'on a $M_k$et que l'on calcule $M_{2k}$, on peut remplacer au fur et à mesure les $m_{ij}^{(k)}$ de $M_k$ par les $m_{ij}^{(k)}$de $M_{2k}$, ce qui permet de ne réserver de la place en mémoire que pour une matrice $(n \times n)$.

Il existe encore de nombreux autres algorithmes de détermination de plus courts chemins sur un graphe valué par des longueurs positives, dont certains ne constituent que des variantes des algorithmes que nous venons d'exposer.

Ils ont chacun leurs avantages et leurs inconvénients propres, variables d'ailleurs suivant la configuration du graphe utilisé. Avant de choisir l'utilisation de tel ou tel algorithme, il convient de se poser un certain nombre de questions, par exemple :

- veut-on la description complète des chemins de valeur minimale ou seulement la valeur de ces chemins (choix entre algorithmes de type Ford ou Moore Dijsktra et algorithmes matriciels) ?
- veut-on connaître le chemin le plus court reliant deux sommets particuliers ou les chemins reliant plusieurs couples de sommets (choix entre Ford et Bellmann-Kalaba ou Moore Dijkstra) ?
- dans quelle mesure la taille du graphe peut-elle limiter l'utilisation de l'ordinateur par les problèmes d'encombrement-mémoire (difficulté de la méthode matricielle, en particulier) ?
- aura-t-on à refaire plusieurs fois les calculs sur le même graphe (dans le cas par exemple où les longueurs de certains arcs peuvent être sujettes à modification : certains algorithmes permettent alors de ne pas recommencer totalement la procédure) ?

Enfin, on n'oubliera pas que dans le cas de la recherche de chemins de valeur optimale, il convient avant tout calcul d'examiner si le graphe comporte ou non des circuits : dans l'affirmative, en effet, le chemin trouvé risque fort d'être de valeur infinie.

## 6.4. APPLICATION DE LA NOTION DE CHEMINEMENT DANS UN GRAPHE : LES PROBLEMES D'ORDONNANCEMENT

### 6.4.1. Généralités

Dans cette section, nous allons examiner les problèmes posés par la réalisation d'un objectif pouvant se décomposer en tâches élémentaires multiples. C'est par exemple le cas de la construction d'une maison ou encore le problème très célèbre du passage de pièces sur un certain nombre de machines.

Résoudre le problème d'ordonnancement attaché à un objectif donné, c'est déterminer l'évolution dans le temps de la réalisation de cet objectif, c'est-à-dire, fixer à l'avance les dates d'exécution de chacune des tâches élémentaires dont est constitué l'objectif, et cela de façon :

- à respecter un certain nombre de contraintes auxquelles sont assujetties les tâches élémentaires.
- à satisfaire au mieux un ou plusieurs critères précisés à l'avance : il peut s'agir de critères de type coût minimal ou durée minimale ou encore, consommation minimale d'un certain type de produit...

Quant aux contraintes, elles peuvent être de plusieurs sortes :

- soit il s'agit de contraintes *potentielles*, qui indiquent par exemple des successions sur les tâches (la tâche $n°j$ doit commencer après la fin de la tâche $n°i$) ou encore des localisations temporelles obligées ( la tâche $i$ doit commencer après la date $t$).
- soit il s'agit de contraintes *disjonctives* qui expriment que certaines tâches ne peuvent s'effectuer simultanément : c'est le cas par exemple de tâches nécessitant l'utilisation d'une même machine, disponible seulement pour l'une d'entre elles.
- soit il s'agit encore de contraintes dites *cumulatives* : celles-ci expriment qu'une certaine matière ou qu'un certain moyen de production, qui peut être utilisée pour plusieurs tâches, n'est cependant disponible qu'en quantité limitée; ainsi si $\Gamma_k(t)$ est la quantité maximale du facteur $k$ disponible à $t$ et si $\gamma_{ik}(t)$ est la quantité de $k$ requise par la tâche $i$ à l'instant $t$, on doit avoir :

$$\sum_i \gamma_{ik}(t) \leq \Gamma_k(t) \qquad \forall_t$$

Un problème d'ordonnancement est, on le voit, l'exemple type de problèmes combinatoires : dès que le nombre de tâches devient quelque peu important, il est difficile de trouver des solutions respectant les contraintes et par ailleurs satisfaisantes du point de vue des critères choisis.

Jusqu'à une époque relativement récente (1957), on ne disposait d'algorithmes que dans des cas très limités, et pour le cas général, on se servait d'un outil assez fruste destiné d'avantage à la visualisation qu'à la résolution du problème, à savoir le diagramme de Gantt.

En abscisse de ce diagramme, on représente les temps $t_i$ de début des tâches $i$, en ordonnée le numéro $i$ des tâches. Une barre de longueur $d_i$ égale à la durée de la tâche est dessinée avec pour extrémité initiale le point de coordonnées $(t_i, i)$.

On peut ainsi, par tâtonnements successifs, constituer la "visualisation" d'un ordonnancement compatible avec les contraintes. Mais ce diagramme ne permet guère d'optimiser l'ordonnancement et par ailleurs, toute modification dans les contraintes nécessite en général la reconstruction totale du graphique.

Depuis, se sont développées des méthodes plus perfectionnées, quoique souvent fondées sur des principes simples, qui font un large usage de la théorie des graphes. Il s'agit essentiellement de la méthode américaine *PERT* (Project Evaluation and Review Technique) et de la méthode française dite *méthodes de potentiels.* Nous allons donner une illustration de ces deux méthodes sur un exemple simple.

### 6.4.2. Un exemple de problème d'ordonnancement

Soit la réalisation d'un certain projet que l'on peut décomposer en 15 opérations élémentaires. Le tableau suivant donne la durée de chacune de ces tâches ainsi que les tâches qui doivent la précéder.

| Tâches | Durée | Tâches antécédentes |
|---|---|---|
| A | 6 | - |
| B | 4 | A |
| C | 5 | A |
| D | 4 | A, C |
| E | 6 | D |
| F | 7 | A, B |
| G | 8 | D, F, K, L |
| H | 2 | B, F, G |
| I | 5 | G, M, N |
| J | 2 | A |
| K | 7 | J |
| L | 6 | A, B |
| M | 5 | B, J, K, L |
| N | 7 | K, L, M |
| O | 3 | E |

**Remarque importante** : on n'a affaire ici qu'à des contraintes potentielles. Ne sont introduites ni des contraintes disjonctives, ni des contraintes cumulatives.

La première question que nous allons nous poser sera la suivante : comment ordonner les différentes tâches en respectant les contraintes de succession ci-dessus pour minimiser le temps total de réalisation du projet ?

Nous allons exposer les deux méthodes citées ci-dessus permettant de résoudre ce problème simple d'ordonnancement, ne faisant intervenir que des contraintes de succession.

### *6.4.2.1. Réduction des relations de succession.*

Auparavant, il convient de simplifier au maximum ces contraintes, en supprimant les contraintes redondantes, telles que :

$A$ précède $C$

$C$ précède $D$

$A$ précède $D$

Contraintes qui figurent dans le tableau ci-dessus. La troisième contrainte est évidemment redondante.

Pour éliminer ces contraintes pour chaque tâche, on dresse la liste des tâches antécédentes (c'est celle dont on dispose déjà) puis la liste des antécédentes de ces antécédents etc... Dès qu'une tâche se trouve à la fois dans la liste des antécédentes immédiates et dans la liste d'antécédentes plus éloignées, on peut la supprimer.

| Tâches | 1ère antécédente | Antécédents seconds | Antécédents troisièmes | Antécédents quatrièmes |
|---|---|---|---|---|
| A | - | - | - | - |
| B | A | - | - | - |
| C | A | - | - | - |
| D | ~~A~~, C | ~~A~~ | - | - |
| E | D | A, C | A | - |
| F | ~~A~~,B | ~~A~~ | | - |
| G | D, F, K,L | A, C, B, J | A | - |
| H | ~~B~~, ~~F~~, G | A, ~~B~~, D, ~~F~~, K, L | A, C, B, J | A |
| I | G, ~~M~~, N | D, F, K, L, B, J, ~~M~~ | C, B, J, B, A | A |
| J | A | - | - | |
| K | J | A | - | |
| L | ~~A~~,B | ~~A~~ | - | |
| M | ~~B~~, ~~J~~, K, L | A, ~~J~~, ~~B~~ | A | - |
| N | ~~K~~, ~~L~~, M | J, B, ~~K~~, ~~L~~ | A | - |
| O | E | D | C | |

On poursuit la procédure jusqu'à épuisement des antécédents. On en déduit, pour l'exemple considéré, la liste pertinente des tâches antécédentes :

| | |
|---|---|
| A | - |
| B | A |
| C | A |
| D | C |
| E | D |
| F | B |
| G | D, F, K, L |
| H | G |
| I | G, N |
| J | A |
| K | J |
| L | B |
| M | K, L |
| N | M |
| O | E |

Une fois les relations de succession simplifiées, on s'efforcera de représenter le problème par un *graphe.*

#### *6.4.2.2. Représentation par un graphe*

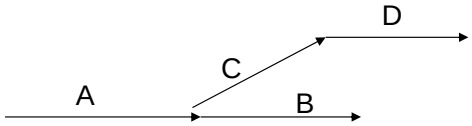
C'est à ce niveau que les deux méthodes citées, (PERT et potentiels) diffèrent essentiellement, bien que comme on le verra, le dessin du graphe ne soit absolument pas nécessaire pour la méthode des potentiels.

a) Dans la méthode PERT, on construit un graphe dont l*es sommets représentent des évènements et les arcs des opérations,* c'est-à-dire les tâches élémentaires dont l'objectif est constitué.

Les arcs sont valués par les durées des tâches élémentaires, et la succession des arcs respecte les contraintes de succession.

La construction du graphe, à partir du moment où l'on possède les contraintes de succession, peut ne pas être facile si le projet est constitué d'un nombre important de tâches.

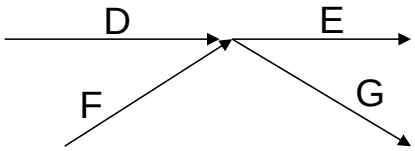
En effet, le tracé de certaines parties du graphe ne pose aucun problème; par exemple, pour le cas que nous traitons, les 4 premières contraintes peuvent se représenter très facilement :




En revanche, d'autres parties sont plus délicates. Par exemple, comment représenter les contraintes :

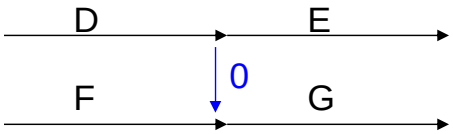
$D$ précède $E$
$D$ précède $G$
$F$ précède $G$ ?

On ne peut proposer la représentation suivante :




car cela impliquerait que F précède E, ce qui n'apparaît pas dans les contraintes.

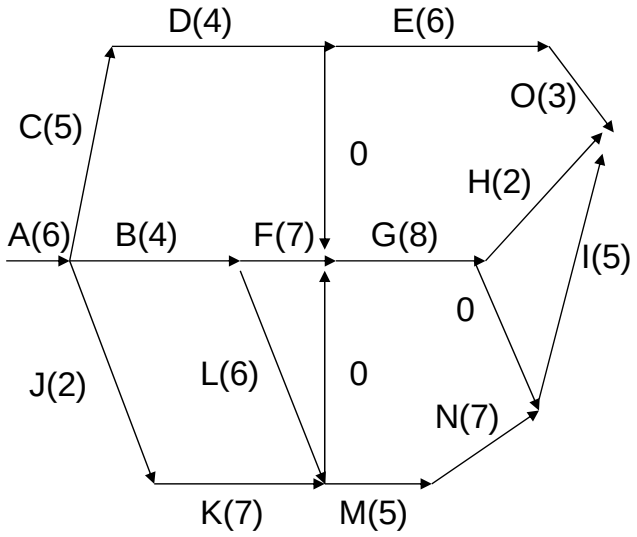
On introduit alors une tâche fictive, de durée nulle, de la façon suivante :




qui exprime bien que G suit D sans que F précède E.

Si alors, on essaie de tracer le graphe complet correspondant au tableau des contraintes ci-dessus, on peut proposer le dessin suivant : (les valuations des arcs sont portées entre parenthèses).

Les arcs n'ayant pas d'antécédents ont leurs extrémités initiales confondues et ceux n'ayant pas de successions ont leurs extrémités finales confondues. Le graphe a ainsi une entrée et une sortie.



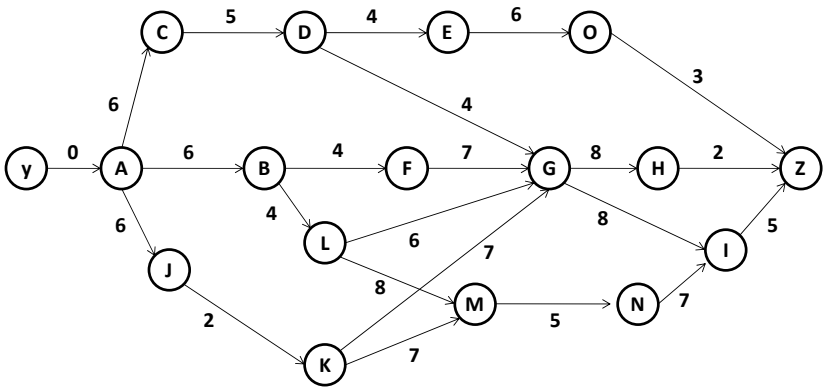

On voit que l'on a été obligé d'ajouter trois tâches fictives pour éviter d'introduire des contraintes inexistantes; cette opération peut devenir complexe si le nombre de tâches élémentaires est important.

Remarquons qu'un tel graphe est obligatoirement :

- connexe, sinon l'on pourrait décomposer le projet en deux sous-projets indépendants,
- sans circuit, car cela impliquerait des contradictions dans les contraintes de succession.

b) Dans la méthode des potentiels, le graphe que l'on dessine a pour sommets les tâches et pour arcs les contraintes, ici, les contraintes de succession. Les arcs sont valués par le délai qui doit s'écouler entre le début de la tâche correspondant à l'extrémité initiale et le début de la tâche correspondant à l'extrémité finale; dans l'ensemble traité, ces valuations sont simplement les durées des tâches associées aux extrémités initiales des arcs.

On introduit un sommet terminal *Z* indiquant la fin des opérations et un sommet initial, *Y* début des opérations. Dans ces conditions, le graphe de la méthode des potentiels, pour l'exemple étudié peut donc être tracé de la façon suivante :




On voit que le tracé de cet arc ne nécessite pas l'introduction d'arcs fictifs.

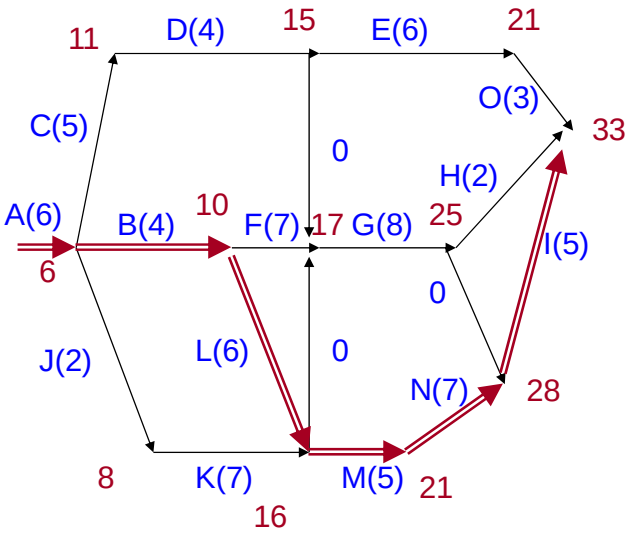
#### *6.4.2.3. Détermination de la durée minimale des travaux.*

Pour déterminer la durée minimale des travaux, prenons d'abord le graphe de PERT.

Cette durée est donnée par la valeur totale du chemin de valeur maximale entre l'entrée et la sortie du graphe.

En effet, étant donné que toutes les tâches doivent être exécutées, *tous* les chemins du graphe doivent être parcourus, et la durée minimale de chacun des chemins est obtenue si l'on considère que les tâches sont exécutées sans temps-mort entre deux d'entre elles. La valeur du plus long de ces chemins est alors la durée minimale que l'on peut attendre pour l'ensemble du projet.

Pour trouver le chemin de valeur maximale, on peut utiliser l'algorithme de Ford, tel qu'il a été modifié pour ce type de problème. On obtient ainsi le chemin suivant (en traits doubles), ainsi que le délai minimum total des travaux (33).




Ce chemin est appelé *le chemin critique* [4]. Le moindre retard dans l'exécution d'une tâche de ce chemin entraîne une augmentation du délai minimal total des travaux.

Sur le graphe, on a indiqué en chaque sommet la valuation telle qu'elle ressort de l'algorithme de Ford.

Le chemin critique, on le sait, est obtenu en partant du sommet sortie du graphe, et en remontant vers l'entrée en sélectionnant à chaque fois l'arc tel que :

$$\lambda_j - \lambda_i = t_{ij}$$

si $\lambda_j$ est la valuation du sommet terminal de l'arc considéré

$\lambda_i$ celle du sommet initial

$t_{ij}$ la durée de la tâche correspondant

La valuation $\lambda_i$ indique donc que la tâche correspondant à l'arc ne pourra pas être entreprise avant $\lambda_i$.

Les valuations données par l'algorithme de Ford sont alors les *dates au plus tôt.* Ces dates au plus tôt sont attachées aux évènements représentés par les sommets du graphe, et qui constituent les étapes successives de la réalisation du projet.

On définit aussi une *date au plus tard* pour chaque sommet : si l'évènement attaché à ce sommet survient après cette date, la durée totale du projet augmente.

---

[4] Il peut y avoir plusieurs chemins critiques.

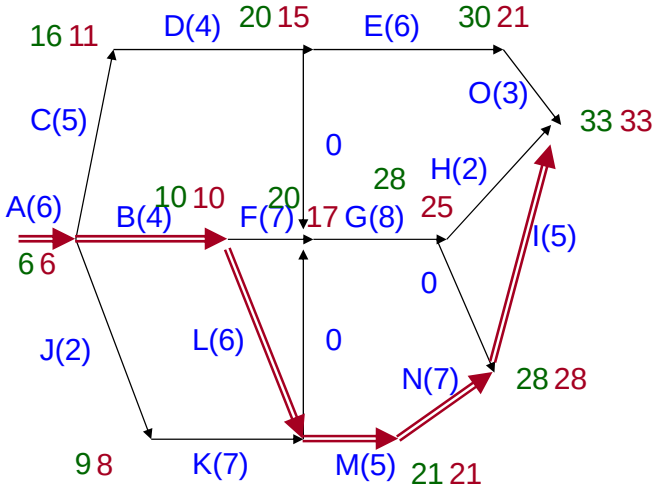
Pour calculer ces dates au plus tard, il suffit de partir du sommet terminal, en lui affectant la durée minimale déjà calculée (ici 33); puis on remontera dans le graphe vers l'entrée, en affectant à chaque sommet la date au plus tard calculée de la façon suivante :

soient les dates au plus tard des sommets terminaux des arcs issus d'un sommet considéré, on retranche à chacune de ces dates la valuation de l'arc correspondant. Le plus petit des nombres obtenus est la date au plus tard du sommet examiné.

Formellement, si $\Gamma(i)$ indique les suivants du sommet $i$, si $t_{ij}$ indique la valuation de l'arc joignant le sommet $i$ au sommet $j$, si $t_i^*$ désigne la date au plus tard du sommet $i$, on a :

$$t_i^* = \min_{j \in \Gamma(i)} (t_j^* - t_{ij})$$

Pour l'exemple considéré, les dates au plus tôt et les dates au plus tard sont indiquées sur le graphe suivant :




Pour les évènements appartenant au chemin critique, dates au plus tôt et dates au plus tard sont évidemment confondues.

Ainsi, par exemple, pour l'évènement D/E (fin de D ou début de E) la date au plus tôt égale à 15 indique que la fin de D (ou le début de E) ne peut survenir après 15 unités de temps; la date au plus tard égale à 20 indique que si D est achevée avant 20 unités de temps, la durée totale du planning ne sera plus 33, mais sera augmentée et cela d'autant que la date de l'évènement considéré dépassera 20 (du moins si cet évènement est le seul à subir un retard). L'intervalle $[15, 20]$ est appelé intervalle de flottement.

Passons maintenant à la méthode des potentiels.

Si l'on considère le graphe potentiel tel qu'il est dessiné ci-dessus, il n'y a rien à ajouter à ce que nous venons de dire pour le graphe PERT : On peut déterminer exactement de la même façon le chemin critique, les dates au plus tôt et les dates au plus tard. Ces tâches concernent ici sans ambiguïté le début des tâches symbolisées par les sommets du graphe.

Mais en fait, on peut très bien ici se passer du dessin du graphe, et utiliser une procédure automatique particulièrement simple. Cette procédure consiste à dresser un tableau, constitué d'autant de colonnes qu'il y a de tâches, plus une colonne pour la tâche fictive fin de travail (Z dans l'exemple pris). Dans chacune de ces colonnes, on établit une sous-colonne indiquant les tâches devant précéder la tâche correspondant à la colonne, ainsi que les durées de ces tâches.

Pour l'exemple considéré, le début de ce tableau donne :

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Y : 0 | A : 6 | A : 6 | C : 5 | D : 4 | B : 4 | D : 4<br>F : 7<br>K : 7<br>L : 7 | G : 8 | G : 8<br>N : 7 | A : 6 |

Lorsqu'une opération n'est précédée d'aucune autre (comme A) on la fait précéder par l'opération fictive Y, de durée nulle.

Sur ce tableau, on va calculer facilement les dates au plus tôt et déterminer le chemin critique. On notera les dates au plus tôt, d'une part devant le sigle de chaque tâche, en tête de colonne, d'autre part devant les tâches des sous-colonnes.

On commence par affecter 0 à *Y* comme date au plus tôt. Lorsque la sous-colonne des dates au plus tôt d'une colonne donnée est entièrement remplie, on prendra pour date de début au plus tôt de la tâche associée à la colonne le nombre calculé de la façon suivante:

pour chaque tâche antécédente (donc figurant à droite) on ajoute sa date au plus tôt (figurant à gauche) et sa durée.

On prend le plus grand des nombres ainsi calculés, et le nombre obtenu constitue la date au plus tôt de la tâche correspondant à la colonne examinée. Il est évident que cette opération correspond exactement à celle que l'on pourrait faire sur le graphe :

si $\Gamma^{-1}(i)$ désigne les tâches antécédentes à la tâche $i$, si $t_i$ désigne la date au plus tôt de la tâche $i$, si $t_{ij}$ représente la valuation de l'arc reliant $i$ à $j$ (durée de la tâche $i$) la date au plus tôt $t_i$ est égale à :

$$t_i = max[t_j + t_{ji}]$$

$$j \in \Gamma^{-1}(i)$$

Le tableau complètement rempli donne :

| 0 | A | 6 B | 6 C | 11 D | 15 E | 10 F | 17 G | 25 H | 28 I | 6 J |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Y : 0 | 0 A:6 | 0 A:6 | 6 C:5 | 11D:4 | 6 B:4 | 11 D:4<br>10 F:7<br>8 K:7<br>10 L :7 | 17 G:8 | 17 G:8<br>21 N:7 | 0 A:6 |

| | | | | | |
|---|---|---|---|---|---|
| 8 K | 10 L | 16 M | 21 N | 21 O | 33 Z |
| 6 J:2 | 6 B:4 | 8 k:7 | 16 M:5 | 15 E:6 | 21 O:3 |
| | | 10 L:6 | | | 25 H :2 |
| | | | | | 28 I :5 |

Pour chaque colonne, on a souligné la tâche antécédente qui détermine la date au plus tôt de l'opération correspondant à la colonne, c'est-à-dire la tâche telle que la somme de sa durée et de sa date au plus tôt fournit la date au plus tôt de la colonne (ou opération) considérée.

Pour avoir le chemin critique, il suffit alors de partir de $Z$, puis de remonter en examinant les tâches soulignées : pour $Z, I$ est souligné; pour $I, N$ est souligné; pour $N, M$ est souligné; pour $M, L$ est souligné; pour $L, B$ est souligné; pour $B, A$ est souligné; et pour $A, Y$ est souligné.

Le chemin critique est donc constitué des tâches $ABLMNI$; c'est celui que l'on avait trouvé par la méthode PERT. On peut également calculer les dates au plus tard sur le tableau ;

- on remarquera d'abord que pour les tâches de chemin critique, dates au plus tôt et date au plus tard doivent être égales. On inscrira donc ces dates dans les colonnes correspondantes.
- ensuite, on part de $Z$ pour les tâches qui n'ont que $Z$ comme successeur (ici, $O, H, I$) on obtient les dates au plus tard en faisant la différence entre la date au plus tard de $Z$ $(33)$ et leur durée.
- pour une tâche quelconque, on repèrera les colonnes du tableau dans lequel elles apparaissent comme antécédentes : par exemple, pour $A$, c'est $C, B, J$. On ne pourra calculer la date au plus tard de la tâche que lorsqu'on disposera des dates au plus tard de toutes les tâches dont cette tâche est l'antécédente. Comme pour le Pert, on calcule les différences entre dates au plus tard des opérations suivantes, et durée de ces opérations et on prend le plus petit de ces nombres.

Compte tenu de ces précautions, le calcul donne ici les dates au plus tard suivantes :

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | A | 6 B | 11 C | 16 D | 24 E | 13 F | 20 G | 31 H | 28 I | 7 J |
| 0 | Y : 0 | 0 A:6 | 5 A:6 | 11 C:5 | 20 D:4 | 6 B:4 | 16 D:4<br>13 F:7<br>13 K:7<br>14 L :6 | 23 G:8 | 20 G:8<br>21 N:7 | 0 A:6 |

| | | | | | |
|---|---|---|---|---|---|
| 9 K | 10 L | 16 M | 21 N | 30 O | 33 Z |
| 7 J:2 | 6 B:4 | 9 k:7<br>10 L:6 | 16 M:5 | 24 E:6 | 30 O:3<br>31 H :2<br>28 I :5 |

On prendra garde aux nuances existant entre les dates au plus tard de la méthode Pert et celles de la méthode des potentiels (comparer le graphe Pert et le tableau ci-dessus).

La méthode des potentiels fournit sans ambiguïté la date au plus tard du début de chacune des tâches; la méthode PERT permet d'obtenir les dates au plus tard d'évènements plus complexes, constitués soit du début, soit de la fin d'une ou plusieurs tâches.

#### *6.4.2.4. Détermination des marges.*

Des notions autres que les dates au plus tôt, les dates au plus tard ou le chemin critique sont également fort utiles : ce sont les notions de marge. On peut en définir plusieurs sortes :

a) la marge libre

En PERT, on la définit de la façon suivante : si une tâche, représentée par un arc reliant le sommet $i$ au sommet $j$ est de durée $t_{ij}$ et si les dates au plus tôt sont désignées par $t_i$ et $t_j$ on aura la marge libre sur la tâche associée à l'arc $i\ j$ égale à :

$$m_{ij} = t_j - t_i - t_{ij}$$

La signification de cette marge est la suivante : si les dates attendues sont les dates au plus tôt et si le retard sur le début de la tâche $ij$ dépasse $m_{ij}$, alors les dates au plus tôt des tâches ultérieures s'en trouvent modifiées, et le calendrier est donc perturbé. C'est donc la marge de retard que peut prendre la tâche $ij$ sans qu'aucune autre tâche en soit affectée. On peut à ce propos définir une autre notion, à savoir la date limite de la tâche $ij$. Elle est donnée ici par :

$$\delta_{ij} = t_i + m_{ij} = t_j - t_{ij}$$

C'est la date de début de la tâche $ij$ au-dela de laquelle les dates au plus tôt d'autres tâches sont perturbées.

Dans le langage des potentiels, la marge libre est définie par

$$m_i = \min_{j \in \Gamma(i)} (t_j - t_i - \theta_{ij})$$

$m_i$ = marge libre de la tâche $i$ associée au sommet $i$.

$t_i$ = date au plus tôt de la tâche $i$

$\theta_{ij}$ = valeur de l'arc reliant le sommet $i$ au sommet $j$ (dans l'exemple pris, c'est la durée de la tâche $i$)

$\Gamma(i)$ = ensemble des tâches suivantes de la tâche $i$

Quant à la date limite $\delta_i$, du début de la tâche $i$, elle est donnée par

$$\delta_i = \min\left[t_j - \theta_{ij}\right] = t_i + m_i$$

$j \in \Gamma(i)$

Si un retard sur une tâche dépassant la marge libre perturbe le calendrier, il peut ne pas être suffisant pour entraîner une augmentation de la durée totale des travaux. C'est pourquoi on définit une autre marge.

b) La marge totale

En PERT, on définit la marge totale par :

$$M_{ij} = t_j^* - t_i - t_{ij}$$

avec

$t_{ij}$ : durée de la tâche qui joint la sommet $i$au sommet $j$;

$t_i$ : date au plus tôt associée au sommet $i$,

$t_j^*$: date au plus tard associée au sommet $j$,

$M_{ij}$ : marge totale sur la tâche associée à l'arc $ij$.

La marge totale a la signification suivante : si le retard sur le début de la tâche $ij$ dépasse $M_{ij}$, non seulement le calendrier est perturbé, mais la durée totale des travaux est augmentée.

En potentiels, on peut définir la marge totale associée à la tâche $i$ par :

$$M_i = t_i^* - t_i$$

Avec

$t_i$ : date au plus tôt du début de $i$

$t_j^*$ date au plus tard du début de $i$

**Remarque :** pour les tâches du chemin critique, les marges libres et totales sont nulles.

Sur l'exemple étudié, le calcul des marges libres et des marges totales donne :

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| marge libre | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 1 | 0 | 0 | 0 | 9 |
| marge totale | 0 | 0 | 5 | 5 | 9 | 3 | 3 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 9 |

Le tableau suivant donne le résumé des notions introduites ci-dessus.

| | PERT | Potentiels |
|---|---|---|
| Date au plus tôt | $t_i$ i : évènement | $t_i$ i : tâche |
| Date au plus tard | $t_i$* i : évènement | $t_i$* i : tâche |
| Marge libre | $m_{ij} = t_j - t_i - t_{ij}$<br>$t_{ij}$ : durée de la tâche ij | $m_i = \min (t_j - t_i - \theta_{ij})$<br>$j \in \Gamma(i)$<br>$\theta_{ij}$ : contrainte de temps entre tâche i et tâche j |
| Date limité | $\delta_{ij} = t_j - t_{ij}$ | $\delta_{ij} = t_i - m_i$ |
| Marge totale | $M_{ij} = t^*_j - t_i - t_{ij}$ | $M_i = t^*_i - t_i$ |

### *6.4.2.5. Modification des contraintes.*

Il peut se faire qu'un problème d'ordonnancement évolue dans le temps, par suite de modifications dans les données. Prenons par exemple le cas simple que nous avons étudié dans les paragraphes précédents et supposons que le planning étant établi par la méthode PERT et la méthode des potentiels, les transformations suivantes soient à intégrer :

a) l'opération F ne peut commencer que 12 unités de temps après le début des opérations,

b) une nouvelle tâche V est introduite : elle doit être précédée par G et E et précéder I. Sa durée est 7.

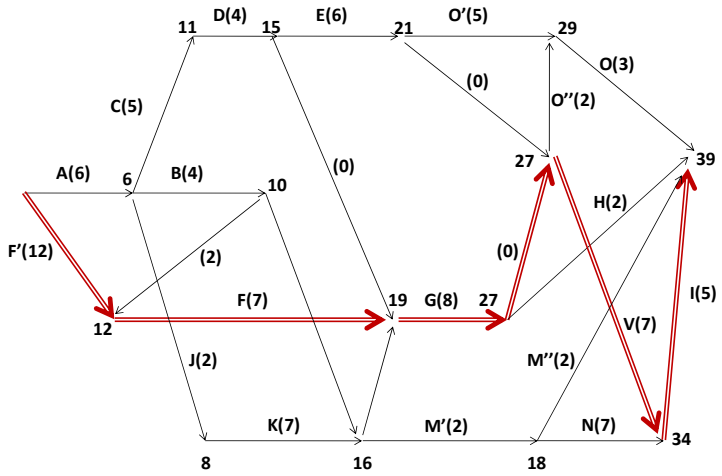
c) l'opération N peut commencer dès que les 2/5 de la tâche M sont accomplis.

d) l'opération O ne peut commencer que si 5 unités de temps se sont passées depuis la fin de l'opération E et 2 depuis la fin de G.

Examinons les transformations qu'induisent ces nouvelles contraintes sur le tracé du graphe PERT. On est obligé de créer de nouvelles opérations :

- une opération F', de durée 12 représentant l'attente entre le début du projet et la tâche F.
- une opération O' de durée 5 représentant l'attente entre la fin de E et le début de O, et une opération O" représentant l'attente entre la fin de G et le début de O.
- deux opérations M' de durée de 2 et M" de durée 3 pour représenter la contrainte c),

- enfin, il faut créer, pour représenter convenablement la contrainte b) des arcs fictifs. Finalement, le graphe PERT peut se représenter de la façon suivante :




Le chemin critique, dessiné en traits doubles sur la figure a changé : il est constitué des tâches F, G, V, et I et le délai minimum total est maintenant égal à 39.

Regardons ce que donneraient ces transformations sur le tableau des potentiels. Ce tableau se transforme de la façon suivante :

| 0 | A | 6 B | 6 C | 11 D | 15 E | 12 F | 19 G | 27 H | 34 I | 6 J |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Y : 0 | 0 A:6 | 0 A:6 | 6 C:5 | 11 D:4 | 6 B:4<br>0 Y :12 | 11 D:4<br>12 F:7<br>8 K:7<br>10 L :6 | 19 G:8 | 18 N:7<br>27 V:7 | 0 A:6 |

| 8 K | 10 L | 16 M' | 18 M'' | 18 N | 29 O | 27 V | 39 Z |
|---|---|---|---|---|---|---|---|
| 7 J:2 | 6 B:4 | 8 k:7<br>10 L:6 | 16 M':2 | 16 M':2 | 15 E :11<br>19 G :10 | 15 E:6<br>19 G :8 | 29 O:3<br>27 H:2<br>34 I:5<br>18 M'' :3 |

On notera que les chiffres à gauche des sous-colonnes ne représentent plus systématiquement les durées des tâches, mais les contraintes de temps existant entre les

tâches. Ainsi, entre le début de E et celui de O, 11 unités de temps doivent s'écouler, ce qui est consigné dans la colonne correspondant à O.

On voit que les modifications apportées au tableau des potentiels sont nettement plus aisées que celles qui ont permis de dessiner le nouveau graphe de PERT. En particulier, l'addition d'arcs fictifs dans le graphe PERT complique assez rapidement le problème. D'une façon générale, la méthode des potentiels est d'une plus grande souplesse que la méthode de PERT. Cela dit, le graphe PERT fournit une bonne visualisation du déroulement des tâches, ce que permet moins le tableau des potentiels. On peut d'ailleurs utiliser conjointement les deux méthodes, en résolvant d'abord le problème par les potentiels et en représentant ensuite la solution trouvée par un graphe PERT (on peut aussi visualiser par un GANTT).

## 6.5. LES AUTRES TYPES DE PROBLEME

Nous n'avons résolu ici que des problèmes d'ordonnancement simples, dans la mesure où ils ne comportaient que des contraintes potentielles (contraintes de succession). Ils donnent cela dit de nombreuses applications effectives, notamment au niveau de la gestion du projet, un mode de gestion qui reçoit un engouement certain actuellement. Il n'y a pas un projet de quelque importance, qu'il s'agisse de la construction d'une infrastructure ou de la conception d'une nouvelle voiture, qui ne commence pas par l'élaboration d'un PERT.

L'introduction de contraintes autres, disjonctives ou cumulatives par exemple, peut compliquer notablement le problème. Les méthodes PERT et potentiels étant de simples adaptations des algorithmes de plus court chemin sont clairement polynomiales. Il n'en est plus de même lorsque l'on introduit des contraintes disjonctives, qui, pourtant, renvoient à des problèmes très réels, comme celui d'ordonnancement d'atelier (interdiction d'usage simultané de la même ressource par des pièces différentes). Les méthodes combinatoires qui ont été proposées ici ou là et qui recherchent l'optimum ne sont plus polynomiales. On a souvent recours à des heuristiques (cf. plus loin).

Une autre complication survient lorsque l'on considère des durées de tâches qui ne sont plus définies à l'avance, mais aléatoires. Là aussi, il est difficile de traiter formellement les problèmes correspondants. La plupart du temps, on se contente de durées moyennes, mais on peut démontrer que ce faisant on biaise vers le bas la durée totale attendue du projet.

Enfin, un autre problème se pose lorsque l'on veut introduire un compromis entre la durée et le coût. Si l'on n'est pas satisfait du délai total trouvé d'effectuation des travaux, on peut penser accélérer certaines tâches, mais en dépensant plus (embauches supplémentaires par exemple). Sous certaines hypothèses (augmentation linéaire des coûts en fonction de la diminution de la durée sur chaque tâche), on dispose d'un algorithme polynomial -le PERT-COST- permettant une réduction donnée du délai au coût minimal.

# Chapitre 7 • Arbres et arborescences

Dans les chapitres précédents, nous avons défini un certain nombre de concepts fondamentaux de la théorie des graphes; puis nous avons donné les moyens de résoudre les problèmes concernant la notion de chemin dans un graphe, et en particulier les problèmes de chemin de valeur minimale.

Dans ce chapitre, nous allons examiner des classes de graphes particuliers, que l'on rencontre très fréquemment dans les applications : les arbres et les arborescences.

## 7.1. ARBRES – DEFINITION

Nous nous plaçons ici dans le cas des graphes non orientés $G = (X, E)$ où $X$ est l'ensemble des sommets et $E$ l'ensemble de arêtes. Il s'agit de multigraphes (cf. chap I) dans la mesure où deux sommets quelconques peuvent être reliés par plusieurs arêtes.

Dans ces conditions, une définition possible d'un arbre est la suivante :

**Définition I**

Un arbre est un graphe $\mathrm{G(X, E)}$ connexe, sans cycles, comportant au moins deux sommets.

En fait, dans de nombreux cas, d'autres définitions équivalentes sont susceptibles d'être utilisées :

**Définition II**

Un arbre est un graphe $\mathrm{G(X, E)}$, sans cycle, comportant n sommets avec $\mathrm{n} \geq 2$, et dont le nombre d'arêtes est égal à $\mathrm{n} - 1$.

Démontrons que la définition I entraîne la définition II. Pour cela nous introduirons les notations suivantes pour caractériser le graphe $G$ :

$n$ : nombre de sommets

$m$: nombre d'arêtes

$p$ : nombre de composantes connexes du graphe (voir chap I).

puis nous définirons un nombre remarquable, dit *nombre cyclomatique* par :

$$v(G) = m - n + p$$

La démonstration peut alors se faire en deux temps :

a) Nous allons d'abord démontrer la propriété suivante : *Si un graphe G ne comporte pas de cycles, on a* $v(G) = 0$.

Pour cela, considérons un multigraphe $G$ quelconque, et $G'$ le multigraphe obtenu en ajoutant une arête entre deux sommets a $et\ b\ de\ G$.

Deux cas se présentent :

1) $a\ et\ b$ ne sont pas reliés dans $G$ par une chaîne (on ne crée pas de cycle en ajoutant l'arête $ab$).

Si $m', n', p'$ sont respectivement le nombre d'arêtes, le nombre de sommets et le nombre de composantes connexes de $G'$, on a dans ce cas :

$$m' = m + 1$$
$$n' = n$$
$$p' = p - 1$$
$$d'où$$
$$v(G) = v(G)$$

2) $a\ et\ b$ sont reliés dans $G$ par une chaîne (on crée des cycles en ajoutant l'arête $ab$). Alors :

$$m' = m + 1$$
$$n' = n$$
$$p' = p$$
$$et$$
$$v(G') = v(G) + 1$$

Considérons alors un graphe $G$ quelconque et le graphe partiel issu de G constitué par les sommets isolés de $G$ (suppression de toutes les arêtes).

Pour ce graphe : $G_0$

$$\mathrm{v}(G_0) = 0 - n + n = 0$$

Ajoutons progressivement les arêtes pour reconstituer $G$.

D'après ce que l'on vient de voir, le nombre cyclomatique augmente d'une unité lorsque l'on relie deux sommets déjà reliés entre eux par une chaîne, c'est-à-dire lorsque l'on crée au moins un cycle. À la fin du processus, on a donc :

$v(G) \leq 0$ nombre de cycles de G.

et si G ne contient pas de cycles :

$v(G) = 0$ C.Q.F.D

**Remarque 1** : on voit également que $v(G) \geq 0$

**Remarque 2** : si G contient *un seul* cycle, on voit que $v(G) = 1$

b) si $G$ est sans cycle et connexe,

$$p = 1\ v(G) = m - n + p = 0$$

d'où $m = n - 1$

En conclusion la définition I entraîne bien la définition II.

**Définition I→ Définition II**

De la même façon, nous donnerons une troisième définition d'un arbre :

**Définition III**

*Un arbre ,est un graphe $G(X, E)$ connexe, comportant n sommets avec $n \geq 2$ et dont le nombre d arêtes est égal à $n - 1$.*

On a : **Définition II → Définition III**

En effet, si $G$ est sans cycle $v(G) = m - n + p = 0$
Si par ailleurs $m = n - 1$, on a $p = 1$, et $G$ est connexe.

**Définition IV**
*Un arbre est un graphe $G = (X, E)$ avec $n \geq 2$, sans cycle et où l'on créerait un cycle et un seul en réunissant par une arête deux sommets non adjacents.*
On a

**Définition III → Définition IV**
En effet, si $p = 1$, et $m = n - 1$ on a $m - n + p = 0$ ; si par ailleurs on ajoute une arête, ni $n$ ni $p$ ne changent; le nouveau graphe $G'$ est tel que $v(G') = v(G) + 1 = 1$
$G'$ possède donc un cycle et un seul.

**Définition V**

Un arbre est un graphe $G(X, E)$, avec $n \geq 2$, connexe, tel que la suppression d'une arête le rendrait non connexe.

**Définition IV → Définition V**
En effet, si $G$ est sans cycle, $m - n + p = 0$. Si on fait $m = m + 1$ (addition d'une arête) la définition IV donne pour le nouveau graphe $G'$ : $v(G') = m' - n + p' = 0$
Soit $m + 1 - n + p' = 1$ soit $m - n + p' = 0$. d'où $p' = p$. Mais si $G$ n'était pas connexe, on pourrait avoir dans certains cas $p' < p$.

Donc, $G$ est connexe. Par ailleurs si on supprime une arête à $G$, comme

$m - n + p = 0\ et\ p = 1\ \ m = n - 1$ ; alors $m' = n - 2\ n' = n$ et on a toujours $m' - n' + p' = 0$ (ce n'est pas en supprimant une arête qu'on créera un cycle); alors $n - 2 - n + p' = 0$ d'où $p' = 2$. Le graphe $G'$ est bien non connexe.

**Définition VI**

Un arbre est un graphe $G(X, E)$, avec $n \geq 2$, tel qu'il existe une chaîne et une seule entre toute paire de sommets.

**Définition V → Définition VI**

En effet, si G est connexe, il existe une chaîne entre deux sommets quelconques, mais il n'en existe qu'une, car dans le cas contraire, la suppression d'une arête ne créerait pas nécessairement deux composantes connexes.

Enfin, on a l'application :

**Définition VI → Définition I**
En effet, $G$ est connexe puisqu'il existe une chaîne entre chaque paire de sommets, mais comme cette chaîne est unique, il n'y a pas de cycle.
En conséquence la suite d'implications :
$\boldsymbol{Déf\ I \rightarrow Déf II \rightarrow Déf III \rightarrow Déf IV \rightarrow Déf V \rightarrow Déf VI \rightarrow Déf I}$ montre que les six définitions proposées sont équivalentes.

## 7.2. PROBLEMES SUR LES ARBRES

On rencontre souvent dans la pratique les problèmes suivants :

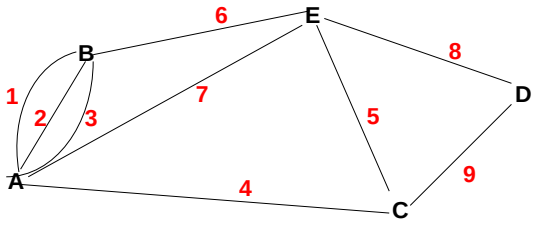
**Problème 1**
Soit un graphe $G(X, E)$. Trouver un graphe partiel de $G$ qui soit un arbre (on sait qu'un graphe partiel de G est obtenu en supprimant un certain nombre d'arêtes de $G$).

La méthode à utiliser est donnée par la démonstration du résultat suivant :

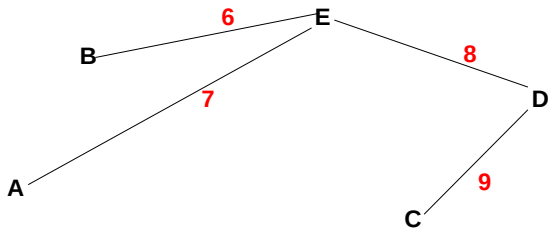
Un graphe $G(X, E)$ admet un graphe partiel qui soit un arbre si et seulement si il est connexe. En effet, s'il n'est pas connexe, aucun des graphes partiels n'est connexe, et donc G n'admet pas d'arbres partiels.

S'il est connexe, cherchons une arête dont la suppression ne donne pas un nouveau graphe non connexe ; si une telle arête n'existe pas, c'est que $G$ est déjà un arbre, d'après la définition V. Si une telle arête existe, supprimons-la. Ensuite, on cherche à éliminer une nouvelle arête, etc. Si nous ne pouvons plus supprimer d'arête, c'est qu'on a obtenu un arbre partiel, toujours d'après la définition V.

Par exemple, si on prend le graphe :



en essayant de supprimer les arêtes dans l'ordre de numérotation, on obtient :




**Problème 2**
Soit un graphe connexe $G = (X, E)$ tel que toute arête u est associée à un nombre $l(u) \geq 0$, appelé valeur de l'arête. Trouver l'arbre partiel de $G$ de valeur totale minimale.

Il existe de nombreux algorithmes pour résoudre ce problème. Nous allons en donner deux.

**1) Algorithme de Kruskal.**

Cet algorithme repose sur le résultat suivant : Si $(X, T)$ est l'arbre de valeur minimale du graphe $(X, E)$, $(X, T)$ vérifie la condition :

Pour toute arête $u \in E - T$, le cycle $\mu_u$ créé en ajoutant $u$ à $T$ a toutes ses arêtes $v \neq u$ telles que

$$l(u) \geq l(v)$$

Cette condition est évidente : en effet, en ajoutant $u \in E - T$ à $T$, on crée bien un cycle et un seul $\mu_u$ ; par ailleurs, les arêtes $v \neq u$ de $\mu_u$ appartiennent à $T$. S'il existe un $v$ tel que $l(v) \geq l(u)$, il suffit de remplacer $v$ par $u$ dans l'arbre $T$ pour obtenir un arbre de valeur moindre que $T$.

Dans ces conditions, l'arbre cherché, de valeur totale minimale contient nécessairement l'arête de plus faible valeur; de même, il contient l'arête de valeur immédiatement supérieure à celle de la précédente, à condition que ces deux arêtes ne forment pas un cycle etc. C'est ce raisonnement qui est à la base de l'algorithme.

L'arbre partiel $U = (v_1 \ldots v_{n-1})$ de valeur minimale est obtenu en prenant pour $v_1$ l'arête de plus faible valeur, pour $v_2$ l'arête de plus faible valeur différente de $v_1$ et telle que $(v_1 v_2)$ ne forment pas un cycle, pour $v_3$l'arête de plus faible valeur différente de $v_1$ et $v_2$ et telle que $(v_1, v_2, v_3)$ ne forment pas un cycle etc. Lorsque toutes les arêtes ont des valeurs différentes, l'arbre obtenu est évidemment unique.

Si certaines arêtes ont des valeurs identiques, il peut y avoir plusieurs solutions au problème. Pour trouver une de ces solutions, il suffit de différencier légèrement les arêtes de valeur identique.

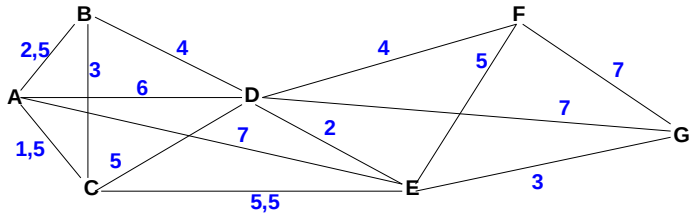
Ainsi, si nous avons trois arêtes avec $l(u_1) = l(u_2) = l(u_3)$, on posera :

$$l(u_1) = l$$
$$l(u_2) = l + \varepsilon$$
$$l(u_3) = l + 2\varepsilon$$

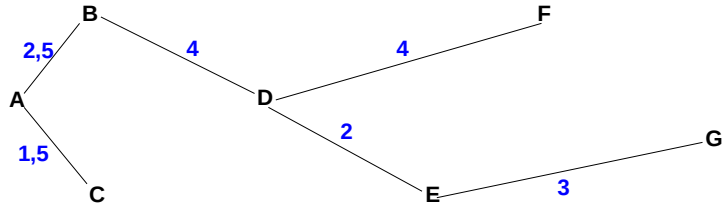
avec $\varepsilon$ suffisamment petit pour ne pas modifier l'ordre des arêtes; une fois le calcul effectué par la procédure exposée ci-dessus, on supprime le $\varepsilon$.

On démontre que l'on obtient bien ainsi un arbre partiel de valeur minimale (quel que soit l'ordre dans lequel on augmente la valeur des arêtes équivalentes).

Appliquons cet algorithme à l'exemple suivant :




On obtient facilement :



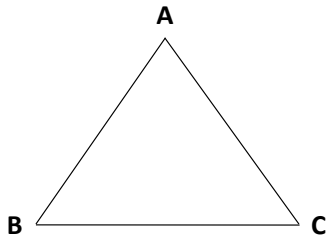

**2) Algorithme de Sollin**

Dans l'algorithme de Sollin, on part d'un sommet arbitraire $x_0$. On le joint au sommet le plus proche (c'est-à-dire au sommet adjacent tel que l'arête reliant $x_0$ à ce sommet soit de valeur minimale parmi celles qui ont $x_0$ pour sommet), soit $x_1$. On prend alors un point $x_2 \notin \{x_0, x_1\}$ on le joint au sommet le plus proche, puis un point $x_3 \notin \{x_0, x_1, x_2\}$ que l'on joint également au sommet le plus proche, etc... A la fin de cette procédure, on obtient des sous-arbres qui peuvent être disjoints, soit $v_i$ et $v_j$ deux de ces sous-arbres. Comme le graphe initial $G$ est connexe, il existe des arêtes de $G$ joignant $v_i$ à $v_j$. On les joint alors par l'arête de plus faible valeur et on continue jusqu'à ce que les sous-arbres soient tous reliés entre eux.

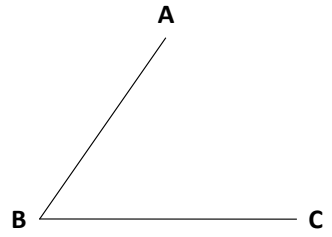
Sur l'exemple précédent, on joint d'abord $A$ à $C$, puis $B$ à $A$, puis $D$ à $E$, puis $F$ à $D$, puis $G$ à $E$. Enfin, on ajoute l'arête $BD$ pour connecter le graphe. On retrouve bien l'arbre partiel obtenu par l'algorithme de Kruskal.

Le problème de la recherche d'un arbre partiel minimal se pose souvent dans la pratique, essentiellement lorsqu'il s'agit de dessin de réseaux, tels que voies de communications entre villes, réseaux de pipelines, de galeries dans une mine etc.

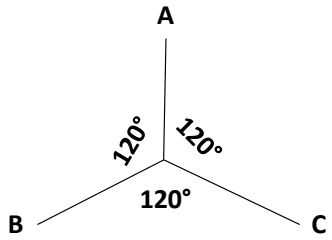
Dans ces conditions, les sommets du graphe $G$ représentent des points de passage obligatoires de l'arbre partiel recherché. Cela dit, on peut encore améliorer l'arbre partiel trouvé par une procédure type Kruskal ou Sollin en créant des sommets intermédiaires. Par exemple, pour le graphe très simple suivant :




L'arbre partiel de valeur minimale est :




mais le réseau reliant les trois points et de longueur totale minimale est :




où le sommet $D$ créé est tel que l'on voit les trois côtés du triangle $ABC$ sous un angle de $120^{\circ}$. On essaiera à chaque fois que l'on aura trouvé l'arbre partiel de valeur minimale, de l'améliorer par l'addition de tels sommets intermédiaires. Les procédures pour y

arriver ne sont pas toujours simples, et ne conduisent pas forcément à l'optimum : on aura plutôt affaire à des heuristiques.

## 7.3. ARBORESCENCES – DEFINITION

### 7.3.1. Définitions préalables

a) Graphes quasi fortement connexes

Un graphe orienté $G = (X, U)$ est quasi fortement connexe si, pour toute paire de sommets $x, y$, il existe un sommet $z$ du graphe d'où partent des chemins reliant $z$ à $x$ et $z$ à $y$.

**Remarque** : un graphe quasi fortement connexe est évidemment connexe si on le désoriente. Un graphe fortement connexe est quasi fortement connexe, la réciproque n'étant pas vraie.

b) Racine d'un graphe

Un sommet $z$ d'un graphe $G(X, U)$ est une racine si, pour tout sommet $x \neq z$ de $G$, il existe un chemin reliant $z$ à $x$.

**Théorème : la condition nécessaire et suffisante pour qu'un graphe $G(X, U)$ admette une racine est qu'il soit quasi fortement connexe.**

En effet, si $G$ admet une racine, il est évidemment quasi fortement connexe. D'autre part, supposons $G$ quasi fortement connexe. Considérons les sommets de $G$, soit $x_1, x_2, \dots, x_n$.

Il existe un sommet $z_1$ d'où l'on peut aller en $x_1$ et $x_2$ un sommet $z_2$ d'où l'on peut aller en $x_3$ et $z_1$ ; un sommet $z_3$ d'où l'on peut aller en $x_4$ et $Z_2$. Un sommet $z_{n-1}$ d'où l'on peut aller en $x_n$ et $z_{n-2}$. En reprenant en sens inverse, on voit que de $z_{n-1}$ on peut aller en tous les sommets du graphe : $z_{n-1}$ est donc une racine.

### 7.3.2. Arborescences. Définition

Comme pour les arbres, il existe de nombreuses définitions des arborescences. Nous retiendrons essentiellement les trois définitions suivantes :

**Définition I :** une arborescence est un graphe $H$ (on supposera toujours que $n \geq 2$) quasi fortement connexe et sans cycle, si on le désoriente.

**Définition II :** une arborescence est un graphe $H$ quasi fortement connexe de $n - 1$ arcs.

**Définition III :** une arborescence est un graphe $H$ qui est un arbre si on le désoriente et qui a une racine.

Il est évident que ces trois définitions sont équivalentes :

En effet, **Définition I** $\rightarrow$ **Définition** II, car si $H$ est quasi fortement connexe, il est connexe; comme il est sans cycle, c'est un arbre, qui a $n - 1$ arêtes (voir les définitions des arbres).

**Définition II → Définition III**, car si $H$ est quasi fortement connexe, c'est qu'il a une racine, d'après le théorème ci-dessus. Par ailleurs, étant connexe et ayant $n-1$ arcs, c'est un arbre.

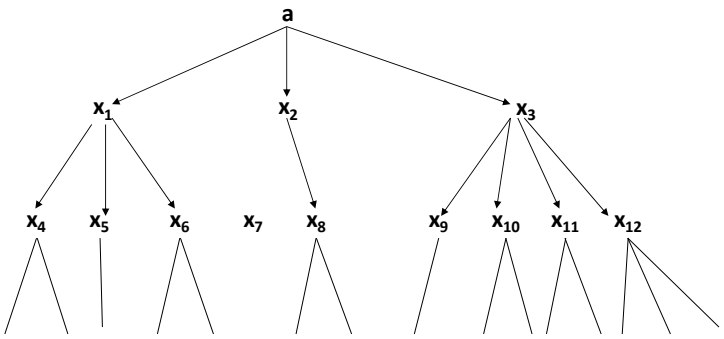
**Définition III → Définition I**, car si $H$ admet une racine il est quasi fortement connexe, et s'il est un arbre, il n'a pas de cycle.

Ces trois définitions sont donc équivalentes. Parmi les autres définitions, dont nous ne démontrerons pas l'équivalence avec les définitions précédentes, citons :

**Définition IV :** une arborescence est un graphe tel qu'un sommet $a$ est relié à tout autre sommet par un chemin unique issu de $a$ (qui est donc une racine).

**Définition V :** une arborescence est un graphe connexe tel que les demi-degrés intérieurs de tous les sommets sont égaux à 1 sauf pour un sommet $a$ pour lequel le demi-degré intérieur est nul.

Si l'on tient compte de ces définitions, le dessin d'une arborescence dans le plan est très classique :




*Exemple d'arborescence : $a$ est la racine*

Ce type de dessin apparaît dans de multiples applications : arbres généalogiques, ordres hiérarchiques, classifications, théorie des questionnaires, etc.

### 7.3.3. Ordres définis sur une arborescence

On peut définir sur une arborescence plusieurs ordres et préordres : parmi ceux-ci, citons:

a) l'ordre associé

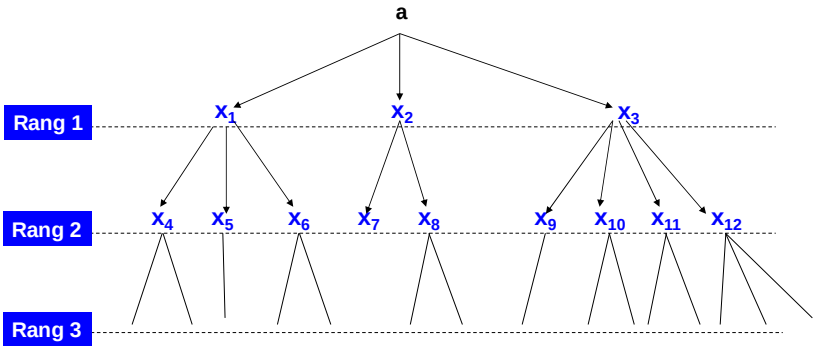
Dans un graphe quelconque $G(X,U)$, la relation $R$ définie sur $X$ par $xRy$ (s'il existe un chemin entre $x$ et $y$) constitue un préordre : en effet, cette relation est réflexive si l'on admet que $x$ est relié à lui-même et elle est évidemment transitive.

Pour une arborescence, cette relation est antisymétrique : elle définit donc un ordre, qui est d'ailleurs partiel. Pour cet ordre, $a$ est minorant de tous les sommets.

b) ordre transverse

D'après la définition IV des arborescences, le chemin reliant la racine $a$ à un sommet $x$ quelconque du graphe est unique. Appelons alors rang du sommet $x$ la longueur (nombre d'arcs) du chemin reliant $a$ à $x$.

On peut classer les sommets suivant les rangs. Exemple :




Ce classement définit encore un ordre partiel. Pour compléter l'ordre, on peut penser à ordonner les sommets de même rang, ce qui définit l'ordre transverse.

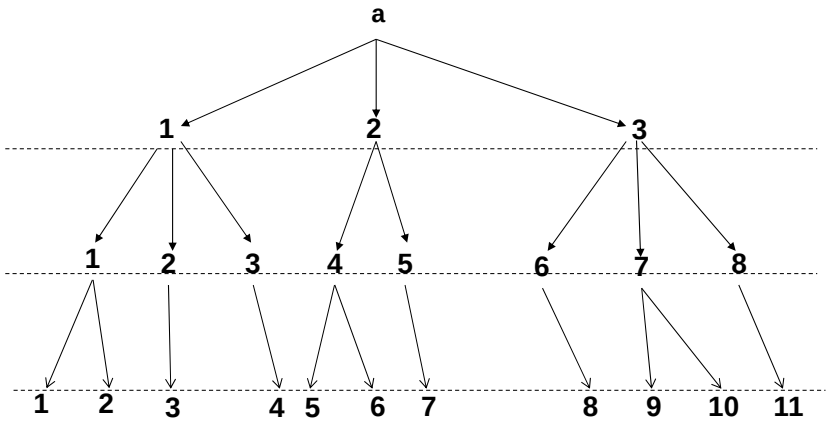
Pour un rang donné, cet ordre est donné tout simplement par la représentation que l'on donne de l'arborescence, en ordonnant les sommets de ce rang de gauche à droite.

Plus précisément, d'après la définition V des arborescences, un sommet quelconque n'a qu'un précédent. La relation "avoir même précédent immédiat" définit une relation d'équivalence, donc des classes d'équivalence. Alors, l'ordre transverse défini sur l'ensemble de sommets d'un rang donné obéit aux deux règles :

1) les sommets d'une même classe sont consécutifs

2) si deux sommets de même rang $\alpha \; et \; \beta$ appartiennent à deux classes d'équivalence différentes $C_\alpha \; et \; C_\beta$ définies par deux sommets du rang inférieur $x$ et $y$, l'ordre entre $\alpha \; et \; \beta$ est le même que celui entre $x$ et $y$.

Par exemple :




On a muni d'un ordre complet chaque ensemble de sommets d'un rang déterminé.

c) Ordre de Tarry

Supposons que nous ayons représenté l'arborescence comme sur les figures précédentes, c'est-à-dire que :

- les sommets d'un même rang sont sur une même horizontale
- pour un rang donné, les sommets sont rangés suivant l'ordre transverse précédent.

Imaginons alors un promeneur se déplaçant sur l'arborescence en appliquant les règles suivantes :

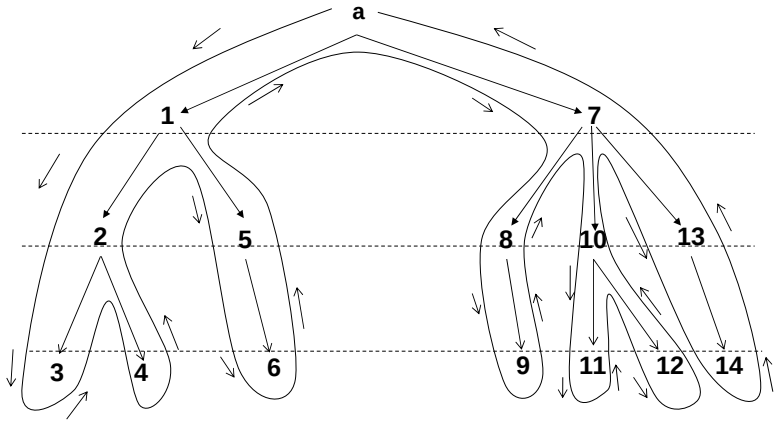
1- il part de la racine

2- il quitte le sommet où il se trouve en prenant l'arc le plus à sa droite et dans le sens de son orientation; si cette règle ne lui permet plus de progresser, il applique la règle 3.

3- il rebrousse chemin (en parcourant les arcs dans le sens contraire à leur orientation) jusqu'à ce qu'il atteigne un sommet d'où parte un arc qu'il n'a pas encore emprunté; alors, il respecte la règle 2.

L'ordre de Tarry est celui dans lequel le promeneur rencontre chaque sommet *pour la première fois.*

Ainsi, sur l'arborescence suivante, on a déterminé la marche du promeneur de Tarry (jusqu'au retour à la racine *a*) et numéroté les sommets suivant l'ordre trouvé.




Ces différents ordres ont une certaine importance dans les techniques que nous allons exposer maintenant et qui constituent une illustration particulièrement intéressante des arborescences : *les procédures de recherche arborescentes.*

## 7.4. LES PROCEDURES DE RECHERCHE ARBORESCENTS

Ces procédures tentent de répondre à la question tout à fait générale : comment trouver un élément $x_0$ d'un ensemble $E$ tel que, si $f(x)$ est une fonction à valeur réelle définie pour tout $x \in E$, l'on ait $f(x_0) \leq f(x), \forall x \in E$, soit plus simplement : trouver le minimum de la fonction $f$ sur l'ensemble $E$.

Les mathématiques nous ont fait rencontrer ce problème très souvent; en ce qui concerne la recherche opérationnelle, on peut même dire qu'elle est complètement dominée par cette question.

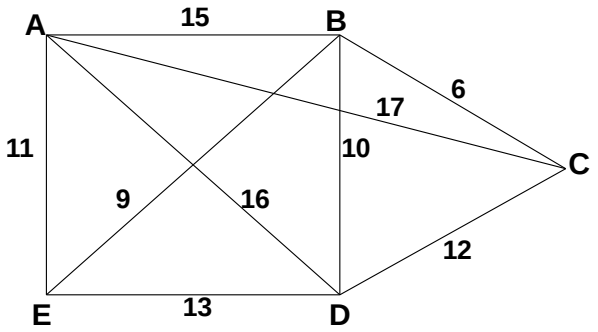
Pour certains problèmes, comme on l'a vu, il existe des procédures d'optimisation permettant d'obtenir le point $x_0$ : c'est le cas, par exemple, des programmes linéaires, où $E$ est un polyèdre convexe, et pour lesquels l'algorithme du simplexe permet de trouver l'optimum en un temps de calcul le plus souvent raisonnable.

Pour d'autres problèmes, il n'existait pas encore, jusqu'à une date récente, de procédure d'optimisation : l'exploration arborescente leur a apporté une solution. C'est le cas du problème du voyageur de commerce, que nous allons d'abord étudier sur un exemple, avant de fournir des généralités sur les procédures de recherche arborescente.

### 7.4.1. Un exemple d'exploration arborescente : le problème du voyageur de commerce

Pour faire comprendre facilement ce qu'est une exploration arborescente, nous allons prendre un exemple très simple :

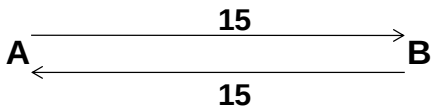
Soit 5 villes $A\ B\ C\ D\ E$ reliées entre elles par des routes à double sens, suivant le schéma ci-dessous :




Les distances entre les villes sont indiquées sur les arêtes.

Un voyageur de commerce se pose la question suivante : il voudrait passer une fois et une seule fois dans chacune des villes et revenir à son point de départ de telle façon que la distance parcourue soit la plus faible possible.

En termes de graphe, le problème se pose de la façon suivante : il s'agit de trouver un cycle hamiltonien (cycle qui passe en une fois et une seule en chacun des sommets) de valeur totale minimale. Dans le cadre des graphes orientés, on peut aussi remplacer chaque arête par deux arcs orientés en sens contraire et de même valeur (dans d'autres graphes, les valeurs peuvent évidemment être différentes) :




Il s'agit alors de trouver dans ce nouveau graphe le circuit hamiltonien de valeur minimale.

C'est sur le concept orienté que nous allons raisonner en nous basant sur un algorithme permettant de résoudre ce problème, l'algorithme de Little.

1) Evaluation par défaut minimale

Ecrivons la matrice des valeurs des arcs entre les différents sommets, en remplissant la diagonale de $+\infty$ et en posant égale à $+\infty$ la valeur d'un arc qui n'existe pas.

**Matrice 1**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | +∞ | 15 | 17 | 16 | 11 |
| B | 15 | +∞ | 6 | 10 | 9 |
| C | 17 | 6 | +∞ | 12 | +∞ |
| D | 16 | 10 | 12 | +∞ | 13 |
| E | 11 | 9 | +∞ | 13 | +∞ |

Il est évident qu'on ne change pas de circuit hamiltonien optimal lorsqu'on diminue les valeurs de tous les arcs partant de A d'une même quantité : en effet, cette opération, étant donné qu'un circuit hamiltonien quelconque emprunte un seul arc issu de A, ne fait que diminuer de cette quantité les valeurs de tous les circuits hamiltoniens existant dans le graphe considéré. Dans ces conditions, on peut diminuer tous les éléments de la première ligne de la matrice ci-dessus d'une quantité telle que l'on fasse apparaître un zéro. Cette quantité est ici égale à 11. On peut faire le même raisonnement sur les lignes relatives à B, à C, etc., si bien que l'on fait apparaître un zéro en moins sur la ligne :

**Matrice 1**

| | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| A | +∞ | 15 | 17 | 16 | 11 | - 11 |
| B | 15 | +∞ | 6 | 10 | 9 | - 6 |
| C | 17 | 6 | +∞ | 12 | +∞ | - 6 |
| D | 16 | 10 | 12 | +∞ | 13 | - 10 |
| E | 11 | 9 | +∞ | 13 | +∞ | - 9 |

**Matrice 2**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | +∞ | 4 | 6 | 5 | 0 |
| B | 9 | +∞ | 0 | 4 | 3 |
| C | 11 | 0 | +∞ | 6 | +∞ |
| D | 6 | 0 | 2 | +∞ | 3 |
| E | 2 | 0 | +∞ | 4 | +∞ |

La somme des éléments ôtés aux différentes lignes est : $11 + 6 + 6 + 10 + 9 = 42$.

Comme tous les éléments de la nouvelle matrice sont positifs ou nuls, tout circuit hamiltonien a une valeur, calculée à partir de ces éléments, également positive ou nulle, ce qui signifie que, puisqu'en diminuant les éléments d'une ligne on diminuait d'autant les valeurs de l'ensemble des circuits hamiltoniens, la valeur de ces derniers est au moins égale à 42.

Mais on peut faire le même raisonnement sur les colonnes puisqu'un circuit hamiltonien emprunte un seul des arcs arrivant à un sommet.

On voit alors qu'on peut enlever 2 aux éléments de la nouvelle A et 4 à ceux de la colonne D, ce qui donne la nouvelle matrice :

**Matrice 2**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | +∞ | 4 | 6 | 5 | 0 |
| B | 9 | +∞ | 0 | 4 | 3 |
| C | 11 | 0 | +∞ | 6 | +∞ |
| D | 6 | 0 | 2 | +∞ | 3 |
| E | 2 | 0 | +∞ | 4 | +∞ |

- 2 - 4

**Matrice 3**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | +∞ | 4 | 6 | 1 | 0 |
| B | 7 | +∞ | 0 | 0 | 3 |
| C | 9 | 0 | +∞ | 2 | +∞ |
| D | 4 | 0 | 2 | +∞ | 3 |
| E | 0 | 0 | +∞ | 0 | +∞ |

17

On a ôté encore 6 aux valeurs des circuits hamiltoniens en conservant des valeurs positives ou nulles aux arcs. Au total, on est sûr que la valeur d'un circuit hamiltonien du graphe est supérieure ou égale à 48. On dira que 48 constitue une évaluation par défaut sur l'ensemble des circuits hamiltoniens du graphe.

2) Séparation

Appelons $H$ l'ensemble des circuits hamiltoniens du graphe. Nous allons partitionner l'ensemble $H$ en deux sous-ensembles disjoints : le sous-ensemble des circuits hamiltoniens qui passent par un certain arc et ceux qui n'y passent pas. Comment choisir cet arc ? L'algorithme de Little propose le raisonnement suivant : prenons un zéro de la matrice 3, par exemple le zéro de l'arc $\overrightarrow{AE}$, et considérons les circuits hamiltoniens qui ne passent pas par $\overrightarrow{AE}$. On peut alors calculer une nouvelle évaluation par défaut de la valeur de ces circuits hamiltoniens particuliers. En effet, comme un circuit hamiltonien quelconque doit emprunter un arc issu de A, l'arc le plus court après $\overrightarrow{AE}$ qu'il puisse emprunter est $\overrightarrow{AD}$ de valeur 1; de même, puisqu'il doit emprunter un arc arrivant sur E, l'arc de ce type le plus court est $\overrightarrow{BE}$ ou $\overrightarrow{DE}$, de valeur 3. On voit que le fait d'exclure l'arc $\overrightarrow{AE}$ d'un circuit hamiltonien fait que ce circuit, calculé sur la matrice 3 a une valeur au moins égale à $3 + 1 = 4$, donc, calculé sur la matrice 1, une valeur au moins égale à $48 + 4 = 52$.

Cette quantité égale à 4 sera appelée *regret* pour l'arc $\overrightarrow{AE}$. Sur la matrice 3, on peut ainsi calculer les regrets pour tous les arcs de valeur nulle, par la formule (si $a_{ij}$ sont les éléments de la matrice).

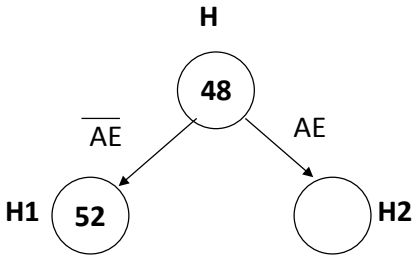
$$\theta_{ij} = \min_{k \neq j} a_{ik} + \min_{\ell \neq i} a_{\ell j}$$

On obtient :

$\theta_{AE} = 4$ $\theta_{BC} = 0 + 2 = 2$ $\theta_{BD} = 0$ $\theta_{CB} = 2$

$\theta_{DB} = 2$ $\theta_{EA} = 4$ $\theta_{EB} = 0$ $\theta_{ED} = 0$

L'algorithme de Little préconise alors de choisir l'arc de *regret maximal* (par exemple $\overrightarrow{AE}$) pour partitionner l'ensemble $H$ (il s'agit d'une règle empirique qui en principe accélère la recherche de l'optimum). Dans ces conditions, $H$ sera partitionné en deux sous-ensembles : $H_1$, ensemble de circuits hamiltoniens ne passant pas par $\overrightarrow{AE}$ et $H_2$ ensemble des circuits hamiltoniens contenant $AE$ ce que nous allons symboliser de la façon suivante :




On connait une évaluation par défaut pour $H_1$ : 52. Qu'en est-il pour $H_2$ ?

Comme un circuit hamiltonien n'emprunte qu'un seul arc partant de A et qu'un seul arc arrivant en E, les arcs autres que $\overrightarrow{AE}$ des circuits de $H_2$ seront choisis sur la matrice 4 obtenue à partir de la matrice 3 en supprimant la ligne A et la colonne E.

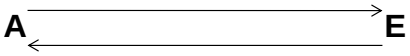
**Matrice 4**

| | A | B | C | D |
|---|---|---|---|---|
| B | 7 | +∞ | 0 | 0 |
| C | 9 | 0 | +∞ | 2 |
| D | 4 | 0 | 2 | +∞ |
| E | +∞ | 0 | +∞ | 0 |

**(-4)**

**Car on a supprimé l'arc AE**

On remarquera que l'on a fait apparaître $+\infty$ en EA pour éviter des circuits parasites du type :

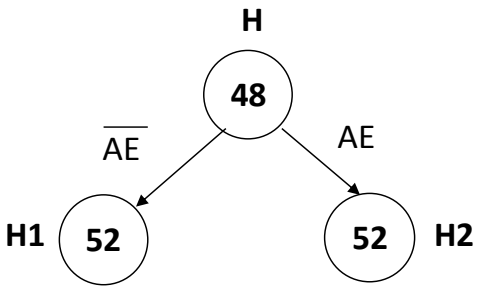



On observe alors qu'on peut faire apparaître un nouveau zéro dans la colonne A en ôtant 4 à tous les éléments de cette matrice.

**Matrice 5**

| | A | B | C | D |
|---|---|---|---|---|
| B | 3 | +∞ | 0 | 0 |
| C | 5 | 0 | +∞ | 2 |
| D | 0 | 0 | 2 | +∞ |
| E | +∞ | 0 | +∞ | 0 |

Et on obtient alors une évaluation par défaut sur $H_2$ $(AE)$ égale à $48 + 4 = 52$.




3) Création d'une arborescence et choix du sommet à séparer.

Recommençons sur $H_1$l'opération que l'on vient de faire sur $H$. Nous reprenons la matrice 3 en remplaçant le zéro de $\overrightarrow{AE}$ par $+\infty$ puisque $H_1$ est caractérisé par le fait que l'arc $\overrightarrow{AE}$ n'est pas emprunté.

**Matrice 6**

| | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| A | +∞ | 4 | 6 | 1 | +∞ | - 1 |
| B | 7 | +∞ | 0 | 0 | 3 | |
| C | 9 | 0 | +∞ | 2 | +∞ | |
| D | 4 | 0 | 2 | +∞ | 3 | |
| E | 0 | 0 | +∞ | 0 | +∞ | |
| | | | | | - 3 | |

Ce qui donne, par soustraction des éléments convenables, la matrice 7 (soustraction de 1 à la ligne A et de 3 à la colonne E).

**Matrice 7**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | $+\infty$ | 3 | 5 | 0 | $+\infty$ |
| B | 7 | $+\infty$ | 0 | 0 | 0 |
| C | 9 | 0 | $+\infty$ | 2 | $+\infty$ |
| D | 4 | 0 | 2 | $+\infty$ | 0 |
| E | 0 | 0 | $+\infty$ | 0 | $+\infty$ |

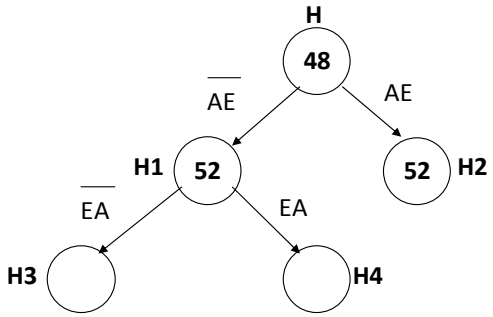
**Attention** : l'évaluation par défaut correspondant à cette matrice est toujours 52; elle correspond à l'exclusion de $\overrightarrow{AE}$ des circuits hamiltoniens et prend en compte le regret calculé précédemment.

Les nouveaux regrets de cette matrice sont :

$\theta_{AD} = 3$ $\theta_{BC} = 2$ $\theta_{BD} = 0$ $\theta_{CB} = 2$ $\theta_{DB} = 0$

$\theta_{DE} = 0$ $\theta_{EA} = 4$ $\theta_{EB} = 0$ $\theta_{ED} = 0$

On sélectionne l'arc $\overrightarrow{EA}$ pour partitionner $H_1$ en deux sous-ensembles, le sous-ensemble $H_3$ des circuits hamiltoniens qui ne contiennent ni $\overrightarrow{AE}$ ni $\overrightarrow{EA}$ et l'ensemble $H_4$ des circuits hamiltoniens qui ne contiennent pas $\overrightarrow{AE}$, mais contiennent $\overrightarrow{EA}$.



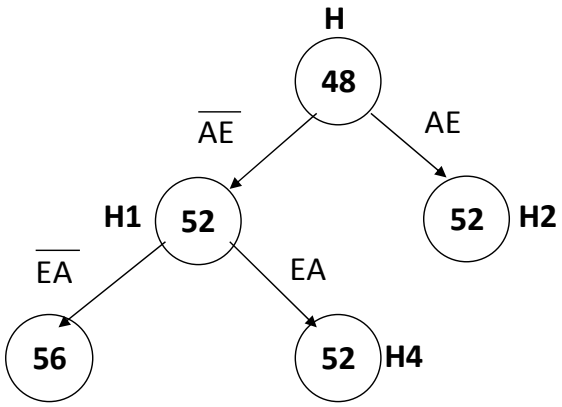

Pour $H_3$ l'évaluation par défaut est $52 + 4 = 56$. Pour $H_4$ la nouvelle matrice est (suppression de la ligne E et de la colonne A).

**Matrice 8**

| | B | C | D | E |
|---|---|---|---|---|
| A | 3 | 5 | 0 | $+\infty$ |
| B | $+\infty$ | 0 | 0 | 0 |
| C | 0 | $+\infty$ | 2 | $+\infty$ |
| D | 0 | 2 | $+\infty$ | 0 |

$(\overline{AE} + EA)$

Cette matrice comporte déjà un zéro par ligne et par colonne et ne permet donc pas de corriger l'évaluation par défaut qui reste 52.




On voit ainsi se dessiner une arborescence : chaque sommet de cette arborescence représente un sous-ensemble de l'ensemble des circuits hamiltoniens du graphe considéré.

Chaque sommet donne naissance à deux sommets représentant deux parties du sous-ensemble représenté par le sommet initial; chacune de ces parties est spécifiée par la présence ou non de tel ou tel arc dans les circuits hamiltoniens qui les constituent.

Lorsque l'on passe d'un sommet à ses deux successeurs, on dit que l'on *sépare* le sous-ensemble correspondant au sommet.

Par ailleurs, on dispose en chaque sommet d'une quantité qui est inférieure à la valeur de tous les circuits hamiltoniens appartenant au sous-ensemble correspondant à ce sommet : on dit qu'on a une *évaluation par défaut de la fonction.*

Supposons alors que l'on poursuive la procédure commencée en adoptant la règle suivante : à chaque étape, on essaie de séparer le sommet obtenu où l'évaluation par défaut calculée est la plus faible possible.

Quant à la séparation elle-même, on utilise toujours le même principe que précédemment : les deux parties du sous-ensemble séparé se distingueront par le fait que leurs éléments (circuits hamiltoniens) passeront ou ne passeront pas par l'arc de plus fort regret. On peut alors avoir trois éventualités :

a) le sommet que l'on veut séparer correspond à un sous-ensemble vide. On dira alors qu'il est *terminal vide*.

Sur l'exemple, on peut s'en apercevoir en considérant la suite d'arcs par où doivent passer les circuits hamiltoniens du sous-ensemble à séparer. Si un sommet apparaît plus de deux fois dans cette suite d'arcs, le sous-ensemble est vide (en effet tout sommet du graphe apparaît exactement deux fois dans un circuit hamiltonien). De la même façon, si le graphe partiel constitué par cette suite d'arcs contient déjà un circuit, il ne peut y avoir de circuit hamiltonien passant par ces arcs.

Alors, ce sommet n'est pas à séparer, on le raye des sommets pendants de l'arborescence, et on recherche le sommet d'évaluation minimale parmi les sommets restants : on tentera de séparer le sommet trouvé.

b) Le sommet que l'on veut trouver n'est pas vide[5] et il contient plus d'un élément (ou du moins, on ne peut décider si l'une ou l'autre de ces propositions est fausse) : alors, on sépare ce sommet, et on calcule l'évaluation par défaut des deux sommets obtenus. On continue alors la procédure.

c) le sommet que l'on veut séparer contient exactement un élément : c'est le cas d'un sommet où le nombre d'arcs par où doivent passer les circuits hamiltoniens relatifs à ce sommet est exactement égal à 5 et où le graphe partiel correspondant à ces arcs forme un circuit hamiltonien. Ce sommet sera dit *terminal non vide*. Mais en ce sommet, on a exactement la valeur du circuit hamiltonien; et non plus une valeur par défaut (c'est la somme des valeurs des 5 arcs retenues).

D'après la règle utilisée pour choisir le sommet à séparer, la valeur de ce circuit hamiltonien est inférieure aux évaluations par défaut calculées en chacun des autres sommets pendants. Mais cela signifie que les circuits hamiltoniens correspondant à ces sommets pendants sont tous de valeur supérieure à celle du circuit hamiltonien trouvé; ce dernier est donc le *circuit hamiltonien optimum*.

**Remarque** : cela est vrai si le sommet à séparer est unique, c'est-à-dire s'il n'existe pas plusieurs sommets d'évaluations par défaut minimales et égales. Dans ce cas, il peut y avoir des solutions optimales équivalentes.

Pour illustrer ces constatations, reprenons l'exemple proposé et appliquons la procédure à partir de calculs déjà effectués.

Séparons le sous-ensemble $H4$ les regrets calculés sur la matrice 8 sont :

$$\theta_{AD} = 3 \qquad \theta_{BC} = 2 \qquad \theta_{BD} = 0 \qquad \theta_{BE} = 0$$

---

[5] Par commodité de langage, on confond ici le sommet et le sous-ensemble qui lui correspond.

$\theta_{CB} = 2$ $\theta_{DB} = 0$ $\theta_{DE} = 0$

On sélectionne donc l'arc $\overrightarrow{AD}$ pour partitionner en $H5$ et $H6$ (cf. l'arborescence complète page suivante). Pour $H6$ ($\overrightarrow{AE}$, $\overrightarrow{EA}$, $\overrightarrow{AD}$), la matrice correspondante est :

**Matrice 9**

| | B | C | E |
|---|---|---|---|
| B | +∞ | 0 | 0 |
| C | 0 | +∞ | +∞ |
| D | 0 | 2 | 0 |

qui donne toujours une évaluation par défaut égale à 52.

Sur la matrice 9, le regret maximal est $\theta_{CB} = \infty$ d'où *H7* et *H8*. *H7* d'évaluation infinie est vide; quant à *H8* il correspond à la nouvelle matrice :

**Matrice 10**

| | C | E |
|---|---|---|
| B | 0 | 0 |
| D | 2 | 0 |

Regret maximal $\theta_{BE} = \infty$ d'où $H9$ (vide) et $H10$ où il n'y a plus qu'un circuit possible : $\overrightarrow{EA}$, $\overrightarrow{AD}$, $\overrightarrow{DC}$, $\overrightarrow{CB}$, $\overrightarrow{BE}$, de valeur 54, et qui est donc terminal non vide.

Mais, du coup, le sommet pendant $H2$ a une évaluation par défaut inférieure à celle du sommet terminal non vide. On doit donc reprendre les calculs en $H2$.

Les regrets calculés sur la matrice 5 donnent :

$\theta_{BC} = 2$ $\theta_{BD} = 0$ $\theta_{CB} = 2$ $\theta_{DA} = 3$

$\theta_{DB} = 0$ $\theta_{EB} = 0$ $\theta_{ED} = 0$

D'où sélection de $\overrightarrow{DA}$ pour séparer $H2$ en $H11$ et $H12$ avec la matrice 11 pour $H12$.

**Matrice 11**

| | B | C | D |
|---|---|---|---|
| B | +∞ | 0 | 0 |
| C | 0 | +∞ | 2 |
| E | 0 | +∞ | 0 |

qui ne change pas l'évaluation par défaut 52. Regret maximal sur cette matrice $\theta_{BC} = \infty$ d'où $H13$ et $H14$, $H13$ est vide et $H14$ correspond à la matrice 12.

**Matrice 12**

| | B | D | |
|---|---|---|---|
| C | +∞ | 2 | - 2 |
| E | 0 | 0 | |

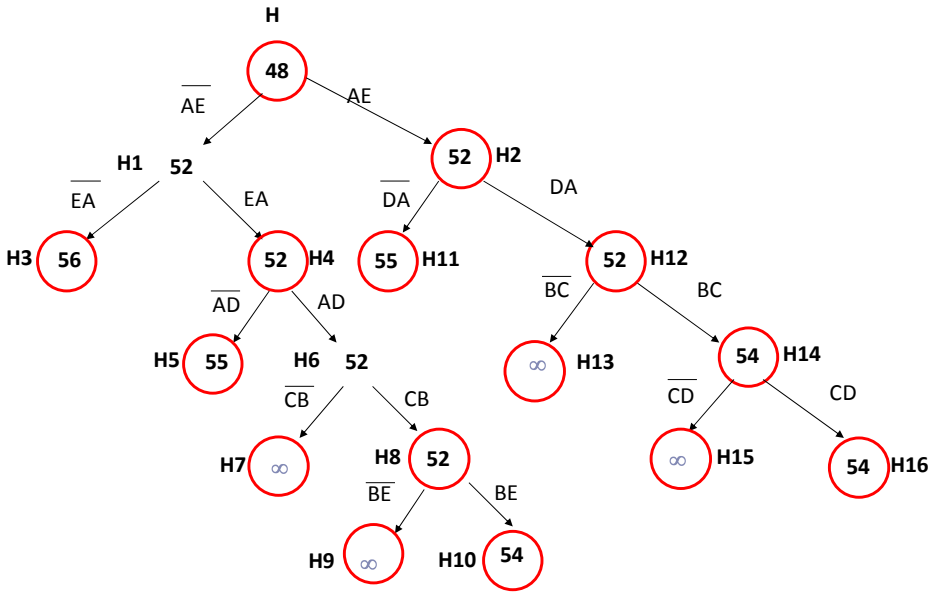
qui donne la matrice 13 par soustraction de 2 à la ligne C

**Matrice 13**

| | B | D | |
|---|---|---|---|
| C | +∞ | 0 | - 2 |
| E | 0 | 0 | |

Regrets maximaux sur cette matrice : $\theta_{CD} = \infty$ $\theta_{EB} = \infty$. Deux nouveaux sommets $H15$ (vide) et $H16$ qui correspond également à un circuit hamiltonien unique :

$\overrightarrow{AE}$, $\overrightarrow{EB}$, $\overrightarrow{BC}$, $\overrightarrow{CD}$, $\overrightarrow{DA}$, de valeur égale à 54 et qui est en fait le même cycle que celui trouvé précédemment. L'algorithme est alors terminé; il n'est pas utile en effet de séparer les ensembles non vides correspondant aux sommets pendants d'évaluation finie, cette évaluation étant supérieure à celle des deux sommets terminaux non vides.




La procédure sur ce graphe simple a été quelque peu longue, à cause de l'équivalence de deux circuits[6].

Cet exemple ayant été traité, nous pouvons passer à un court exposé sur les procédures d'exploration arborescente en général.

## 7.4.2. Les procédures de recherche arborescente

### *7.4.2.1. Principes généraux.*

Comme nous l'avons déjà dit, ces procédures sont censées répondre au problème très général suivant :

soit un ensemble $E$, fini ou non, et une fonction $f$ qui, à tout élément $x \in E$, fait correspondre $f(x) \in R$.

Il s'agit de trouver un élément $x_0$ tel que

$$f(x_0) \leq f(x)\ \forall x \neq x_0\ et\ x \in E$$

ou encore plus simplement : trouver le minimum de $f$ sur $E$.

---

[6] On peut éviter cet inconvénient en ajoutant au début des calculs une quantité $\varepsilon$ très petite sur l'un des deux arcs joignant deux sommets.

Le principe d'une procédure arborescente consiste à restreindre le problème posé à l'examen, non de $E$ mais d'une suite $E_1, E_2, \ldots E_q \ldots$ de sous-ensembles de $E$, en profitant à chaque fois de l'information disponible pour chacun de ces sous-ensembles.

Pour construire cette suite d'ensembles, on utilise deux grands principes :

- un principe de séparation :
  lorsque l'on examine un sous-ensemble $E_q$, on doit disposer d'une méthode permettant de spécifier et d'examiner ensuite des sous-ensembles de $E_q$. On sépare $E_q$ en plusieurs sous-ensembles qui sont inclus dans la suite $E_1 \ldots E_q \ldots$
- un principe d'évaluation : il faut être capable de calculer sur chaque sous-ensemble $E_q$ une borne inférieure de la fonction $f$.

Grâce au principe de séparation on peut, comme on l'a fait pour l'exemple choisi, associer à la suite $E_1 \ldots E_q \ldots$ une arborescence telle que chaque sommet de cette arborescence représente un sous-ensemble $E_q$.

La procédure consiste à explorer cette arborescence en partant de $E_0 = E$ et en respectant certaines règles permettant de trouver la solution optimale.

L'intérêt d'une telle procédure porte sur la faculté que l'on a, à une étape du calcul, d'exclure de la recherche de l'optimum certains sous-ensembles.

En effet, lorsque l'on considère un sommet de l'arborescence, il se peut que :

a) le sous-ensemble correspondant à ce sommet soit vide, auquel cas on n'aura pas à séparer ce sommet.

b) on puisse résoudre le problème posé sur le sous-ensemble $E_q$ correspondant au sommet; c'est-à-dire que l'on trouve l'élément $X_q$ de $E_q$ tel que :

$$f(x_q) \leq f(x) \qquad \forall x \in E_q$$

alors, le sommet est terminal non vide et l'on dispose de l'évaluation $f(x_q)$ qui est le minimum de $f$ sur $E_q$. On ne le séparera pas dans la suite.

c) un sommet ne soit ni terminal vide ni terminal non vide, mais soit affecté d'une évaluation par défaut supérieure à l'évaluation d'un sommet terminal non vide; on est alors sûr que l'optimum ne peut se trouver dans le sous-ensemble correspondant à ce sommet et on peut l'exclure de la recherche.

Cela dit, les différentes procédures diffèrent entre elles essentiellement par les règles utilisées pour choisir à chaque étape le sommet que l'on va séparer.

À ce titre, nous citerons deux grands types de procédures.

- les PSES (Procédures de Séparation et d'Evaluations Séquentielles) où le choix du sommet à séparer se fait indépendamment de la fonction d'évaluation, mais est intrinsèquement lié au principe de séparation : un ordre transverse étant donné (cf. le paragraphe précédent) pour ranger les sommets de l'arborescence

de même niveau, on décrit l'arborescence suivant l'ordre de Tarry qui lui est associé, en considérant que les sommets en cul-de-sacs sont les sommets terminaux vides et les sommets terminaux non vides.
La procédure s'arrête lorsque toutes les évaluations de sommets non encore séparés sont supérieures à la meilleure évaluation trouvée pour un sommet terminal non vide : pour appliquer ce type de procédure, il faut donc déjà savoir à peu près où se situera l'optimum pour organiser l'arborescence de façon à exhumer assez vite les meilleures solutions.

- les PSEP (Procédures de Séparation et d'Evaluation Progressive) où le choix du sommet à séparer se fonde sur la considération des évaluations des sommets pendants (sur lesquels on a le choix). C'est le cas de la procédure que nous avons utilisée pour résoudre notre exemple. Ces procédures sont souvent plus pénibles à programmer que les précédentes mais, par les possibilités d'orientation qu'une bonne fonction d'évaluation permet, elles sont plus souvent appliquées. Nous fournirons donc dans le paragraphe suivant des précisions sur les règles qu'elles mettent en jeu.

#### *7.4.2.2. Les méthodes PSEP*

**Principe de base**
Les conditions auxquelles doivent répondre les principes de séparation et d'évaluation des méthodes PSEP sont les suivantes :

1) Principe de séparation :

*Condition de finitude :*
Si l'on prend un sous-ensemble, qu'on le sépare en sous-ensembles et que l'on sépare encore ses sous-ensembles etc ... la suite obtenue est finie.

*Condition de conservation*
Soit $E_q$ un sous-ensemble; la méthode de séparation donne des $E_j \subset E_q$ avec $j \in S$ où S(q) désigne l'ensemble des indices des sous-ensembles obtenus à partir de $E_q$ (sous-ensembles "descendants" de $E_q$); on doit avoir :

$$\bigcup_{j \in S(q)} E_j = E_q$$

*Condition d'arrêt :*
On n'aura pas à séparer un sous-ensemble $E_q$ dans deux cas :
- il est vide,
- on peut calculer le minimum $f$ sur $E_q$et l'élément ou les éléments correspondants. $E_q$ est terminal non vide.

2) Principe d'évaluation

pour chaque $E_q$ on peut calculer une quantité $v_{(q)}$ telle que :

- $v(q) \leq \min f(x)$

$x \in E_q$

- $v(q) = \min f(x)$ pour un sommet terminal non vide

$x \in E_q$

*Règles de sélection*

La règle de sélection du sommet à séparer est très simple : on choisit le sommet d'évaluation par défaut minimale.

Dans ces conditions l'algorithme général d'une PSEP est le suivant :

*Algorithme*

L'algorithme est itératif : nous désignerons par $p$ le numéro d'une itération quelconque, et par $Z^p$ l'ensemble des sommets candidats à une séparation pour l'itération $p$. Toute itération se décompose en trois phases :

Phase I Sélection

a) $p = 0$ ; on sélectionne $Z^0 = E_0 = E$

b) $p > 0$ on sélectionne le sous-ensemble $E_q$ avec$E_q \in Z^p$ tel que [7]

$v(q) \leq v(j)\ \forall E_j \in Z^p$ c'est la règle annoncée.

on va alors tenter de séparer $E_q$ :

Phase 2 Séparation

Comme on l'a vu, il peut se produire 3 cas, suivant l'état de l'ensemble $E_q$ sélectionné.

a) $E_q$vide; revenir alors à la case sélection en posant $Z^{p+1} = Z^p - \{E_q\}$

b) $E_q$est un sommet terminal non vide ; on a

$$v(q) = \min f(x)$$
$$x \in E_q$$

Comme pour tout $E_j \in Z^p, v(q) \leq v(j)$, d'après la règle de sélection, et que $Z^p$ représente, comme on le verra, l'ensemble des sous-ensembles encore séparables à l'itération $p$, c'est qu'on a trouvé un minimum de la fonction $f$ sur $E$, et ce minimum est

[7] Comme précédemment, on convient de confondre sommet de l'arborescence et sous-ensemble représenté par ce sommet.

dans $E_q$ On arrête la procédure, sauf s'il existe des $E_j \in Z^p$ tels que $v(q) = v(j)$, auquel cas, il pourrait y avoir des solutions équivalentes que l'on peut rechercher en séparant les $E_j$ répondant à $v(q) = v(j)$.

c) $E_q$ est non terminal non vide, on le sépare et on examine ses suivants en calculant l'évaluation par défaut pour chacune d'entre eux.

Phase 3 Examen

Le calcul de l'évaluation par défaut en chacun des suivants de $E_q$ peut faire apparaître deux circonstances intéressantes :

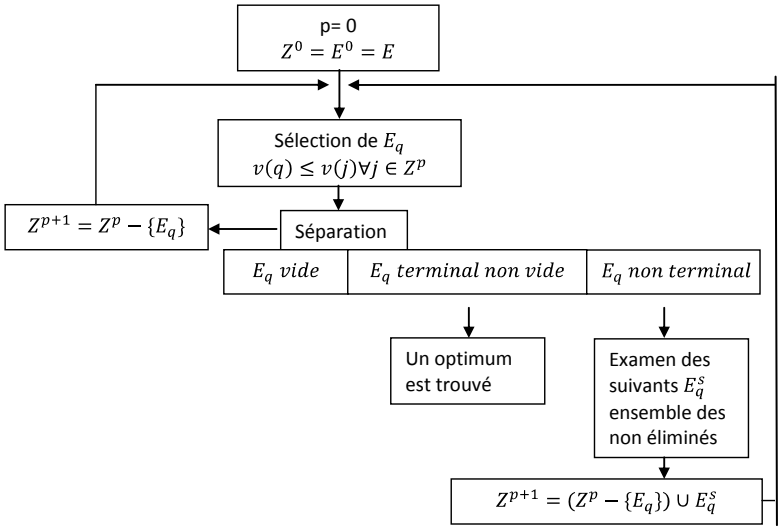
- un suivant $E_j$ de $E_q$ est vide et l'on s'en aperçoit lors de cette phase; on peut alors le supprimer tout de suite de $Z^{p+1}$ (sur l'exemple traité, nous ne procédions pas à cet examen systématique des suivants, mais on aurait pu évidemment le faire).
- un suivant $E_j$ de $E_q$ est tel qu'il existe un sommet *terminal non vide* $E_k$ avec $k \neq j$ et tel que $v(k) < v(j)$; alors, le minimum cherché ne peut être dans $E_j$ et on élimine également $E_j$ de $Z^{p+1}$.

Soit $E_s^q$ l'ensemble des suivants de $E_q$non éliminés, on posera :

$$Z^{p+1} = (Z^p - \{E_q\} \cup E_s^q$$

c'est-à-dire que l'on élimine $E_q$qui n'est plus sommet pendant, puis on conserve l'ensemble des sous-ensembles séparables et intéressants, et on revient à la phase de sélection. Pour l'itération $p+1$, $Z^{p+1}$ représente bien l'ensemble des sommets candidats à la séparation.

Cet algorithme peut être résumé par l'ordinogramme général suivant :



**Remarque** : pendant l'exposé, nous n'avons parlé que de problèmes de minimisation. La procédure exposée reste valable lorsqu'il s'agit de maximiser $j$ sur $E$ à condition de :

a) remplacer "évaluation par défaut" par "évaluation par excès"

b) sélectionner à chaque itération le sommet d'évaluation maximale

## 7.4.3. Quelques remarques sur les procédures de recherche arborescente

### *7.4.3.1. Avantages de ce type de procédure*

On constate en général deux défauts concernant les procédures classiques d'optimisation, tel l'algorithme du simplexe : elles fournissent la solution optimale, mais sont incapables en général d'en explorer le voisinage; par ailleurs, elles sont liées à une structure définie de la fonction économique.

Ces défauts apparaissent moins dans les procédures de recherche arborescentes, qui sont d'une part aptes à explorer certains voisinages, et qui d'autre part peuvent s'adapter à des formes de fonction très diverses ; en particulier, ces procédures peuvent prendre en charge mieux que les autres, plusieurs critères économiques, et non plus un seul : on cherchera par exemple une solution telle que pour chaque critère la valeur pour cette solution n'est pas trop éloignée à un seuil donné de la valeur optimale de ce critère sur *E.*

### *7.4.3.2. Utilisation des procédures de recherche arborescente*

Actuellement ces procédures sont fréquemment utilisées, essentiellement pour des problèmes de graphes (voyageurs de commerce, tournées, typologie) et également pour la programmation linéaire en nombres entiers, qui n'a pas été étudiée dans la partie traitant de la programmation linéaire.

Prenons par exemple le problème très simple suivant, mais très connu en recherche opérationnelle sous la dénomination du "problème du sac à dos" : un randonneur doit partir et $a$ le choix entre $n$ objets à placer dans son sac à dos, sachant qu'il ne veut pas dépasser un poids total maximal P. Il attribue à chaque objet une utilité $u_j$. Par ailleurs, chaque objet a un poids $p_j$. Quels objets notre randonneur doit-il emporter, sachant qu'il veut maximiser l'utilité totale, somme des utilités des objets emportés?

Ce problème s'exprime de la façon suivante : si on pose $x_i = 1$ si le randonneur choisit l'objet $j$ et $x_j = 0$ s'il ne le choisit pas, il s'agit de résoudre le PL en variables bivalentes suivant :

$$\sum_{j=1}^{n} p_j x_j \leq P$$

$$x_j = 0 \; ou \; 1$$

$$Max \sum_{j=1}^{n} u_j x_j$$

**Remarque** : le problème du choix des investissements évoqué à la fin de la partie sur la programmation linéaire obéit à la même formulation, ce qui montre que cet exemple n'a pas seulement une valeur anecdotique.

On peut fort bien imaginer de résoudre ce problème par une méthode arborescente, en construisant des sous-ensembles de solutions, où certaines des variables $x_j$ sont spécifiées et pas les autres. Une évaluation par excès (on est ici dans une perspective de maximisation) est obtenue en chacun de sous-ensembles en résolvant le programme linéaire en variables réelles correspondant, compte tenu des variables spécifiées.

Pour fixer les idées, prenons le petit exemple numérique suivant, à quatre variables :

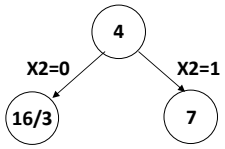
$$3x_1 + x_2 + 2x_3 + x_4 \leq 4$$

$$x_1, x_2, x_3, x_4 = 0 \; ou \; 1$$

$$Max \; 4x_1 + 3x_2 + 2x_3 + x_4 \leq 4$$

Indépendamment du fait que des considérations très simples aboutissent ici à l'optimum (une énumération exhaustive pour $n$ variables conduit à $2^n$ calculs), appliquons les principes précédents.

Le PL en variables réelles donne $x_2 = 4$ , les autres variables nulles, et $z = 12$ (ici le simplexe est réduit à sa plus simple expression) et ne donne donc pas une solution réalisable. Mais c'est bien une évaluation par excès sur l'ensemble des solutions.
Prenons alors le sous-ensemble des solutions tel que $x_2 = 1$ et le sous-ensemble des solutions tel que $x_2 = 0$ (pourquoi le choix de $x_2$? parce que le rapport $\frac{u_j}{p_j}$ est le plus grand pour cette variable, ce qui correspond à une règle intuitive de choix). On obtient le début d'une arborescence, en calculant à chaque fois une évaluation par excès par résolution du PL en variables réelles, avec d'une part $x_2 = 1$ et d'autre part $x_2 = 0$.




Le problème est résolu : en effet, le sous-ensemble
$x_2 = 1$, donne $x_1 = 1$ , $x_3 = 0$, $x_4 = 0$.
C'est donc un sous-ensemble terminal non vide. Comme il a une évaluation par excès supérieure à celle relative au seul autre sommet pendant, on a trouvé l'optimum.

Il va sans dire que sur des problèmes plus riches en données, la procédure peut être plus longue. Encore une fois, on n'a pas l'assurance de ne pas explorer l'ensemble des combinaisons possibles ou en tout cas, une partie non négligeable de cet ensemble.

#### *7.4.3.3. Choix des principes de séparation et d'évaluation*

L'exposé d'une méthode telle que PSEP laisse une grande latitude pour le choix des principes de séparation et d'évaluation. Pour arriver le plus possible à l'optimum, il convient de procéder à ces choix avec circonspection : d'une part, il ne faut pas que le principe de séparation "éparpille" trop les solutions voisines de l'optimum dans les sous-ensembles qu'il génère; d'autre part, il faut que le principe d'évaluation conduise rapidement à l'élimination d'une proportion importante de sous-ensembles, c'est-à-dire qu'apparaissent des sous-ensembles dont l'évaluation est supérieure au minimum de $j$ sur $E$ (ces sous-ensembles ne seront en effet jamais séparés); enfin, la règle de sélection utilisée dans les PSEP est convenable si le biais qu'introduit l'évaluation par défaut est à peu près le même sur tous les sous-ensembles candidats.

#### *7.4.3.4. Problèmes de temps et de place en informatique*

Les problèmes les plus importants relatifs aux procédures de recherche arborescente sont d'ordre informatique; avant d'engager les calculs, on ne sait pas en effet combien d'itérations vont être nécessaires; par ailleurs, le stockage en mémoire de toutes les informations utiles concernant les sommets candidats peut poser des problèmes de temps de gestion des périphériques.

Aussi, au lieu de rechercher l'optimum strict de $f$ sur $E$, se contente-t-on souvent de solutions non optimales, mais dont on a lieu de penser qu'elles ne sont pas trop éloignées de l'optimum : les méthodes utilisées pour obtenir ces solutions satisfaisantes sont dites heuristiques.

Nous les évoquons dans le chapitre suivant en commençant par un retour sur la notion de complexité des algorithmes, notion qui justifie l'existence et l'usage de ces méthodes approchées.

# Chapitre 8 • Complexité des problèmes et heuristiques

## 8.1. RETOUR SUR LA COMPLEXITE DES ALGORITHMES

On a vu plus haut ce qu'on entendait par algorithme polynomial : c'est un algorithme dont on peut démontrer que le temps de calcul est borné par une fonction polynomiale de la taille du problème, la taille étant elle-même définie par le nombre de bits nécessaire pour entrer les données en mémoire (dans un problème de graphe, cette fonction polynomiale s'exprimera par l'intermédiaire du nombre de sommets $n$ et du nombre d'arcs ou d'arêtes $m$).

Les algorithmes de chemin ou d'arbres que nous avons décrits sont polynomiaux. Il n'en est pas de même pour l'algorithme du simplexe ou, comme on vient de le souligner, pour la procédure arborescente susceptible de résoudre le problème du voyageur de commerce. Cela dit, on sait qu'il existe des algorithmes polynomiaux pour résoudre un programme linéaire ; pour l'instant, il n'existe pas de tels algorithmes pour résoudre le problème du voyageur de commerce. Cette différence nous conduit à introduire quelques notions sur la complexité des algorithmes et des problèmes, utiles si l'on veut connaître quel est l'état de l'art actuel en matière d'algorithmique et pour comprendre la nécessité, dans beaucoup de cas, d'introduire des méthodes moins satisfaisantes que les algorithmes d'optimisation, méthodes que nous évoquons dans le chapitre suivant.

**a) Problème polynomial**

C'est un problème combinatoire pour lequel existe un algorithme polynomial (exemple : chemin de valeur minimale, arbre de valeur minimale). Nous appellerons $P$ la classe des problèmes polynomiaux.

**b) Problèmes d'existence et problèmes d'optimisation**

Il est utile de distinguer deux types de problèmes : ceux pour lesquels on cherche une solution qui satisfasse à des contraintes (exemple : existe-t-il un circuit hamiltonien dans un graphe orienté donné ? Ou encore : existe-t-il un circuit hamiltonien de valeur totale inférieure à un nombre $k$ donné ? Ou encore : existe-t-il une solution à un programme linéaire en nombre entier donné ?) Et ceux pour lesquels on cherche la ou les solutions qui minimisent ou maximisent une certaine fonction (c'est le cas des problèmes que nous avons abordés : celui du voyageur de commerce exige que non seulement on trouve un circuit hamiltonien mais que par ailleurs ce circuit soit le plus court possible).

**Remarque** : on peut dans beaucoup de cas ramener la complexité d'un problème d'optimisation à celle du problème correspondant d'existence. Ainsi, il est évident que si on trouvait un algorithme polynomial pour le voyageur de commerce (auquel cas ce problème lui-même est dit polynomial) alors on aurait un algorithme polynomial pour le

problème : existe-t-il un circuit hamiltonien de valeur totale inférieure à $k$ ? Inversement, si ce dernier algorithme existait alors il suffirait de faire varier $k$ pour trouver de façon polynomiale (ou presque : il faut ajouter à la représentation de la taille du problème dans l'ordinateur celle de $k$) l'optimum du problème d'optimisation. Cette équivalence est cela dit de faible apport pour la résolution concrète des problèmes combinatoires. En revanche la distinction que nous opérons ainsi permet d'avancer dans l'exploration de la notion de complexité des problèmes combinatoires.

**c) La classe des problèmes NP**

On ne dispose pas, on l'a dit, d'algorithme polynomial permettant de résoudre le problème du voyageur de commerce. Pour autant, on ne sait pas (ou on ne sait pas encore) si ce problème est polynomial ou non (pour affirmer qu'il n'est pas polynomial, il faudrait démontrer qu'il ne peut pas exister d'algorithme polynomial le résolvant, ce qui n'a pas encore été fait). En revanche, supposons que quelqu'un, à la vue du graphe symbolisant le problème, exhibe un circuit entre les villes et affirme : ce circuit est hamiltonien. Il n'est pas très difficile de montrer qu'on peut vérifier dans un temps polynomial s'il a raison ou pas. De même, si, confronté à un programme linéaire en nombres entiers, on donne des valeurs positives entières aux variables, on peut vérifier dans un temps polynomial si cette solution " devinée " est réalisable ou non.

Les problèmes d'existence pour lesquels existent des algorithmes polynomiaux permettant de vérifier qu'une solution est satisfaisante sont appelés problèmes NP. Une autre façon de dire les choses est d'introduire la notion d'algorithme non déterministe : ce sont des algorithmes qui, en plus des instructions usuelles, comportent des instructions de choix : si de bons choix sont effectués un problème NP peut être résolu en un temps polynomial. C'est le cas du voyageur de commerce (j'ajoute des arcs les uns après les autres de façon à constituer un circuit et si je suis astucieux je peux " tomber " sur un circuit hamiltonien ou mieux sur un circuit hamiltonien qui a une valeur inférieure à $k$. De même, je donne des valeurs entières aux variables du programme linéaire, et avec un peu de chance j'obtiens une solution réalisable, tout cela dans un temps qui est borné par une fonction polynomiale de la taille du problème).

**Attention** : NP ne veut pas dire non polynomial. D'ailleurs, il est facile de voir que les problèmes polynomiaux sont NP. On a donc :

$$P \subset NP$$

**d) Relations d'ordre entre problèmes**

Soit deux problèmes $P_1$ $et$ $P_2$ d'existence. On écrira $P_1 < P_2$ si les conditions suivantes sont respectées :

a) Il existe une transformation $f$ qui est telle que toute solution$s$ $x$ de $P_1$ soit transformée en une solution $f(x)$ $de$ $P_2$.

b) La fonction $f$ est calculable en un temps polynomial

Cela signifie que l'on peut résoudre le problème $P_1$ en résolvant le problème $P_2$. $P_1$ sera dit faiblement réductible à $P_2$.

Par exemple, on peut transformer le problème de l'existence d'un circuit hamiltonien dans un graphe orienté en un problème d'existence d'une solution dans un programme linéaire en variables bivalentes (on pose $x_{ij} = 0\ ou\ 1$ ou suivant que l'arc $ij$ appartient au circuit recherché et on exprime ensuite le fait que l'on a affaire à un circuit hamiltonien)

Si $P_1 < P_2\ et\ P_2 < P_1, alors\ P_1 et\ P_2$ sont dits équivalents.

On peut proposer une relation un peu plus complexe pour les problèmes d'optimisation :

Soit un problème $P_1$ de ce type :

$$Min\ z_1(x)$$
$$x \in D_1$$

et le problème $P_2$ :

$$Min\ z_2(x)$$
$$x \in D_2$$

S'il existe une bijection $g$ de $D_1$ sur $D_2' \subset D_2$, calculable en un temps polynomial, et telle que :

$$z_1(x) = z_2(g(x)) \qquad \forall x \in D_1$$

et

$$z_2(u) > z_2(v) \qquad \forall v \in D'_2 \ \ \forall u \in D_2 - D'_2$$

alors $P_1$ est dit fortement réductible à $P_2$ .

La première condition exprime que toute solution optimale de $P_1$ se conserve dans la bijection ; la seconde permet d'éliminer les solutions de $P_2$ qui ne sont pas solution de $P_1$.

Deux problèmes $P_1\ et\ P_2$ qui sont tels que $P_1$ est fortement réductible à $P_2$ et $P_2$ fortement réductible à $P_1$ sont fortement équivalents.

Ces relations sont importantes à cause des résultats suivants :

**e) Problèmes NP-complets et NP-difficiles**

Un problème d'existence NP est dit NP-complet si tout problème de NP lui est faiblement réductible. En d'autres termes, s'il existe des problèmes NP-complets, et si on démontre qu'un quelconque de ces problèmes se résout de façon polynomiale, alors tous les problèmes NP sont polynomiaux ! Une telle possibilité est confirmée par le théorème de Cook (1970), qui a trouvé de tels problèmes NP-complets en travaillant sur la logique des propositions : soit $n$ variables booleiennes $u_i$ (prenant les valeurs $0\ et\ 1$) et leurs conjuguées $\overline{u_i}$ ($\overline{u_i} = 0$ si $u_i = 1$ et $\overline{u_i} = 1$ si $u_i = 0$), et soit $m$ sous-ensembles des variables $u_i$ et $\overline{u_i}$. Ces sous-ensembles sont appelés des clauses. Si une des clauses comporte au moins une variable égale à 1, elle est dite satisfaite. Le problème suivant :

existe-t-il une fonction *f* sur l'ensemble des $u_i$ avec $f(u_i) = 0\ ou\ 1$ telle que toutes les clauses soient satisfaites ? est un problème NP-complet. Si on trouve un algorithme polynomial pour ce problème de logique, alors on trouve des algorithmes polynomiaux pour de nombreux problèmes combinatoires de la recherche opérationnelle.

Cela dit, la présomption la plus couramment admise est qu'il n'existe pas d'algorithmes polynomiaux pour ces problèmes NP-complets et que l'on risque fort de ne pas trouver de méthode efficace (polynomiale) pour résoudre les problèmes que nous avons évoqués (voyageur de commerce, programme linéaire en nombres entiers, par exemple, le premier étant fortement réductible au second).

Dans la pratique, il peut être utile de démontrer qu'un problème auquel on est confronté est NP-complet : cela signifie que ce n'est sans doute pas la peine de se fatiguer pour trouver un algorithme polynomial, à moins de concourir pour la médaille Fields.

La notion de problème NP-difficile étend celle de problème NP-complet aux problèmes d'optimisation. On démontre que si un problème d'existence est NP-complet, alors le problème d'optimisation correspondant est NP-difficile.

Les programmes linéaires en nombres entiers et le problème du voyageur de commerce sont NP-difficiles.

Finalement, les procédures arborescentes, malgré leur élégance, n'assurent pas que l'on puisse trouver l'optimum de ces problèmes dans un temps raisonnable, en tout cas évidemment pour des exemples d'une certaine taille. C'est pourquoi de nombreux auteurs ont proposé pour ce type de problème des heuristiques : méthodes d'exploration des solutions qui conduisent à des calculs rapides et à des résultats dont on a lieu de penser (mais sans en être jamais totalement certain) qu'ils sont proches de la solution optimale. C'est l'objet du développement suivant.

## 8.2. METAHEURISTIQUES

Pour résoudre des problèmes combinatoires particulièrement difficiles (NP-difficiles pour être précis), du type :

$$Min\ f(x)$$

$$x \in A$$

($A$ représentant l'espace des contraintes ; par souci de simplification, on appellera solution tout $x$ satisfaisant aux contraintes).

Plusieurs techniques ont été élaborées, souvent astucieuses, susceptibles de parvenir à des solutions " vraisemblablement " proches de la solution optimale. On peut être légitimement déçu par le fait qu'on ne recherche plus une solution rigoureusement optimale ; il convient cela dit de souligner que ces " métaheuristiques ", une fois imaginées, sont testées systématiquement sur des cas de grande taille dont on connaît la solution exacte si bien qu'on peut ne conserver, parmi les nombreuses idées explorées par les chercheurs, que celles qui sont en quelque sorte confirmées par l'expérience.

Ce sont quelques une de ces techniques, les plus connues et les plus appliquées en matière d'optimisation combinatoire, que nous exposons ci-dessous rapidement, en nous limitant pour chacune d'elle aux grands principes qui la guident.

On exposera trois grands types de méthode : les algorithmes d'Optimisation par Colonies de Fourmis, les algorithmes gloutons, les méthodes de voisinage, avec deux variantes : le recuit simulé et la méthode Tabou, et les algorithmes génétiques, cette classification n'étant pas sans arbitraire (les algorithmes génétiques, notamment, utilisent la notion de voisinage).

### 8.2.1. Ant colony optimisation

Les algorithmes d'Optimisation par Colonies de Fourmis sont de nouvelles méthodes d'optimisation qui s'inspirent du comportement des fourmis qui sont capables de trouver le chemin le plus court du nid à une source de nourriture et de s'adapter aux changements de l'environnement et ceci grâce à une substance leur permettant de laisser une trace sur leur chemin : la phéromone.

Le problème du voyageur de commerce a fait l'objet du premier algorithme de colonies de fourmis : *Ant System (*Dorigo et Gambardella, 1997). Dans ce problème, il s'agit de trouver le chemin le plus court reliant $n$ villes données ; chaque ville ne devant être visitée qu'une et une seule fois. Formellement, soit un graphe connexe, $(G, U)$, où les sommets représentent les villes et les trajets représentent les arêtes. A chaque itération $t$ $(1 \leq t \leq t_{max})$, chaque fourmi $k$ $(k = 1, \dots, m)$ parcourt au hasard le graphe et construit une chaîne de *n* sommets ($n = |G|$). A chaque étape, la fourmi se déplace du sommet $i$ vers le sommet $j$ en fonction des règles suivantes :

- Sa prochaine destination est choisie dans un ensemble d'arêtes possibles, notée $j_i^k$. Il s'agit de la liste des trajets possibles pour une fourmi $k$ lorsqu'elle se trouve dans une ville $i$,
- la visibilité entre les sommets : $\eta_{ij} = \frac{1}{d_{ij}}$. Ce paramètre est employé pour diriger le déplacement des fourmis vers des sommets proches : plus une ville est proche, plus elle a de chance d'être choisie.
- La quantité des phéromones déposée sur l'arête reliant deux sommets est appelée l'intensité de la piste. Il s'agit de l'attractivité du trajet : plus l'intensité de phéromone disposée sur l'arête entre deux sommets est grande, plus le trajet aura de chance d'être choisi par les fourmis. La règle aléatoire de transition proportionnelle proposée par (Bonabeau et al. 99) est la suivante :

$$p_{ij}^k(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} . \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in J_i^k} \left[\tau_{il}(t)\right]^{\alpha} . \left[\eta_{il}\right]^{\beta}} & si\ j \in J_i^k \\ 0 \quad sinon \end{cases}$$

où les deux principaux paramètres contrôlant le système (*Ant System*) sont $\alpha \; et \; \beta$. Ils représentent l'importance relative de l'intensité de la piste, $\tau_{ij}(t)$, et de visibilité, $\eta_{ij}$ .

Si $\alpha = 0$, seule la visibilité de la ville est prise en compte ; la ville la plus proche est donc choisie à chaque pas. En revanche, si $\beta = 0$, seules les pistes de phéromone jouent (les fourmis sont aveugles).

- Règle (1) de distribution des phéromones. Après un tour complet, les fourmis déposent, sur l'ensemble des arêtes parcourues, une quantité de phéromone $\Delta\tau_{ij}^{k}(t)$. Cette quantité est une fonction de la qualité de la solution trouvée :

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^{k}(t)} & si(i,j)\epsilon T^{k}(t) \\ 0 & sinon \end{cases}$$

où $T^{k}(t)$ est la chaîne parcourue par la fourmi $k$ à l'itération $t$, $L^{k}(t)$ la longueur de la chaîne, $Q$ un paramètre de réglage (souvent du même ordre que la longueur).

- La règle (2) de disparition des pistes de phéromone permet de ne pas tomber dans des solutions optimales locales, les mauvaises solutions disparaissent ainsi pendant l'exécution de l'algorithme. A la fin de chaque itération, la règle de mise à jour des pistes de phéromone est la suivante :

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

où $\Delta\tau_{ij}^{k}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}(t) \; et \; \rho$ un paramètre de réglage. La quantité initiale de phéromone sur les arêtes est une distribution uniforme de petites quantités $\tau_0 \geq 0$.

**Algorithme Ant System – Traveling Salesman Problem**

**Données :**
$|G|$ : Nombre de sommets (villes) dans le graphe ;
$U$, Ensemble d'arêtes (trajets) reliant les sommets (villes) ;
$k$ : ($k \; \epsilon \; \{1, \dots, m\}$ le nombre de fourmis ;
$\tau_0$ : Quantité de phéromone initiale ;

**Initialisation :**
$\tau_0 \rightarrow \tau_{ij}, i, j = 1, \dots, N$ ;
Placer arbitrairement chaque fourmi sur une ville
Pour chaque $t \; \epsilon \; \{1, \dots, t_{max}\}$
    Pour chaque $k \; \epsilon \; \{1, \dots, m\}$
        Construire un chemin $T^{k}(t)$ selon la règle de transition (1)
        Calculer la longueur $L^{k}(t)$de ce chemin
    Soit $T^{+}$ le meilleur chemin trouvé et $L^{+}$ la longueur correspondante ;
    Mettre à jour les traces de phéromones suivant la règle (2) ;
Retourner $T^{+}$et $L^{+}$

Plusieurs développements ont été proposés ensuite pour améliorer cette première version de l'algorithme. D'abord par Dorigo, Maniezzo et Colormi en 1991, consistant à donner à la meilleure chaîne un poids additionnel sous forme d'une quantité supplémentaire de phéromone. Bullnheimern, Hartl et Strauss, 1997 ont proposé de ne retenir que les meilleures fourmis pour la mise à jour des phéromones. D'autres auteurs, Stützle et Hoos, 1997, ont proposé une mise à jour de phéromone réservée uniquement à la meilleure fourmi ayant générée la meilleure solution depuis la première itération de l'algorithme (Best iteration Ant).

En 1997, Dorigo et Gambadella, proposent une règle de transition différente appelée Pseudo-random Proportional Rule pour compléter les développements précédents. Ainsi, à partir d'un sommet $i$, la fourmi $k$ choisit de se déplacer vers le sommet $j$ comme suit :

$$j = \left\{ Arg\ max_{l \in J_i^k} \left[\tau_{ij}(t)\right]^{\alpha} . \left[\eta_{ij}\right]^{\beta} \right\}\ si\ q \leq q_0$$

Depuis, d'autres algorithmes ont été développés pour résoudre de nouveaux problèmes combinatoires tels que celui du routage de véhicule [Bul, 99], du problème du sac à dos multidimensionnel [Ala, 04], des problèmes d'ordonnancement [Shy04] [Lio07].

### 8.2.2. Algorithmes gloutons

Ce sont les plus simples, et ils reposent sur une considération élémentaire : dans un problème combinatoire, une solution est composée d'un certain nombre d'éléments (des variables bivalentes pour le problème du sac à dos, des arcs pour le problème du voyageur de commerce, des arêtes pour le problème de l'arbre de valeur minimale). Une solution incomplète sera constituée par des éléments spécifiés (des variables égales à $0\ ou\ 1$ pour le sac à dos), les autres ne l'étant pas.

A chaque itération, on ajoute un ou plusieurs éléments spécifiés, sans remettre en cause les spécifications précédentes. On complète ainsi progressivement la solution incomplète de façon à obtenir à la fin (tous les éléments sont alors spécifiés) une solution réalisable (mais pas obligatoirement optimale).

Nous avons vu déjà deux exemples de tels algorithmes : pour le problème du chemin de valeur minimale, l'algorithme de Moore est un algorithme glouton : à chaque itération on ajoute un sommet à l'ensemble des sommets pour lesquels on connaît la solution, sans remettre en cause les sommets précédemment trouvés. L'algorithme de Kruskal pour l'arbre de valeur minimale est aussi glouton : on ajoute à chaque fois une arête aux arêtes déjà introduites pour constituer progressivement l'arbre optimal.

La caractéristique de ces deux exemples est que l'on obtient ainsi la solution optimale.

Ce n'est évidemment pas le cas des procédures gloutonnes que l'on peut imaginer pour d'autres problèmes. Par exemple, pour le sac à dos, on peut penser introduire successivement des variables $x_j = 1$, en vérifiant à chaque fois que la contrainte reste satisfaite (si elle ne l'est pas, on pose la variable sous examen égale à $0$ et on essaye une autre). Un ordre intuitif (que nous avons utilisé ci-dessus pour la procédure arborescente) par lequel on introduit les variables peut être donné en prenant les ratios utilité/poids par ordre décroissant. Cette méthode ne conduit pas sur de nombreux

exemples à l'optimum, mais compte tenu du caractère de bon sens du critère utilisé pour spécifier les variables, on peut penser qu'il ne s'agit pas d'une trop mauvaise solution (dans la plupart des cas).

De même, pour le problème du voyageur de commerce, des procédures gloutonnes simple peuvent être utilisées : ainsi, on peut partir d'un sommet et chercher le sommet le plus proche (arc de valeur minimale), puis faire la même chose à partir du nouveau sommet, et ainsi de suite (en ne choisissant jamais un sommet déjà rencontré) jusqu'à « reboucler » sur les sommets de départ. Cela dit, sur de nombreux cas, une telle méthode intuitive peut aboutir à une solution très éloignée de l'optimum.

Ce sont des algorithmes qui, par définition, sont rapides. Cependant, ils ne sont pas « sûrs » : la solution obtenue risque fort d'être médiocre. C'est pourquoi ils sont surtout utilisés pour initialiser les techniques plus sophistiquées que nous allons maintenant aborder.

### 8.2.3. Les méthodes de voisinages

Définissons d'abord ce que l'on entend par voisinage d'une solution :

C'est un sous-ensemble $V(x)$ associé à une solution $x$ grâce à une fonction de voisinage $g$.

$$x \xrightarrow{g} V(x)$$

La fonction a un caractère « local », c'est-à-dire elle change marginalement la solution $x$.

Pour bien faire comprendre cette notion, le mieux est de prendre des exemples :

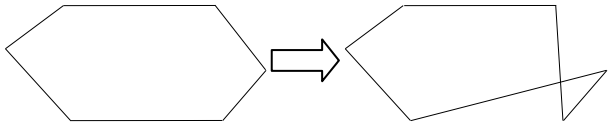
1) Dans le problème de sac à dos ou plus généralement de programmation linéaire en variables bivalentes, la solution peut être considérée comme codée par une suite de $n$ nombres égaux à $0\ ou\ 1$. Une fonction $g$ possible consiste par exemple à remplacer un de ces éléments par son complément à 1.

Ainsi, la solution $(1100010001)$ devient $(110010001)$, le cinquième élément ayant été complémenté.

On voit que cette fonction g simple (un élément quelconque complémenté) fournit un voisinage de $x$ comportant $n$ éléments.

On peut imaginer évidemment d'autres fonctions $g$ (comme celle consistant à intervertir deux éléments de la liste)

2) Pour le problème du voyageur de commerce non orienté, une transformation connue consiste à prendre deux arêtes non adjacentes et à les remplacer par deux autres permettant de reconstituer un cycle hamiltonien. Ainsi :



Cette transformation est appelée 2-opt.

Pour qu'une fonction de voisinage soit intéressante, il faut qu'elle observe deux conditions :

- elle doit être susceptible de fournir de proche en proche toutes les solutions du problème,
- elle doit être réversible : si $x_2 \in V(x_1), alors\ x_1 \in V(x_2)$.

Cette notion de voisinage introduit une méthode générale de recherche d'une bonne solution : il s'agit de la méthode d'amélioration itérative :

a) On part d'une solution initiale $x_0$.

b) On cherche dans le voisinage de $x_i$ une solution qui améliore la fonction $f$ (cette recherche peut prendre plusieurs formes : *exhaustive* - on cherche tous les voisins et on prend le meilleur, *systématique* - on ne cherche pas tous les voisins mais ceux donnés par une règle définie à l'avance, aléatoire (on tire un voisin au hasard) .

c) On remplace $x_i\ par\ x_{i+1}$ si $x_{i+1}$ est en effet meilleur et on recommence en $b$.

d) On s'arrête lorsque l'on ne trouve pas dans le voisinage de $x_i$ de solutions améliorantes.

Il est clair que cette méthode (qui peut être fort longue) a l'inconvénient d'aboutir dans un certain nombre de cas à un minimum local (l'algorithme du simplexe, qui appartient bien à cette classe de méthode, avec une fonction de voisinage consistant à échanger une variable de base et une variable hors base ne comporte pas cet inconvénient : il permet d'obtenir à coup sûr l'optimum).

C'est pourquoi des techniques ont été imaginées pour « sortir » d'un optimum local obtenu par amélioration itérative. La plus connue est constituée par le recuit simulé.

#### *8.2.3.1. Le recuit simulé*

Cette méthode, mise au point au début des années quatre-vingt aux Etats Unis et en Slovaquie, est fondée tout d'abord sur une idée intéressante et simple à la fois, qui mixe en quelque sorte les principes d'exploration combinatoire et la simulation aléatoire : on adopte la méthode générale d'amélioration itérative, mais pour éviter de se faire « piéger » par un minimum local, on accepte des itérations qui dégradent - provisoirement espère-t-on - la fonction économique, mais non systématiquement : on ne le fait qu'en fonction de certaines probabilités.

Quant au terme de recuit simulé, il vient d'une analogie avec la thermodynamique et la métallurgie. Pour obtenir un cristal, état d'énergie minimale du matériau, on le porte à une température élevée et on diminue cette dernière. Le résultat obtenu dépend de la vitesse de refroidissement : si cette dernière est très élevée, on obtient un état métastable, correspondant à un minimum local d'énergie ; c'est le phénomène de la trempe, qui conduit à des matériaux très résistants. Si on veut au contraire obtenir la structure cristalline, il faut faire baisser la température lentement.

Mais comme la décroissance de la température est un processus difficile à maîtriser complètement, il peut se faire qu'elle soit trop rapide et qu'elle conduise à des états métastables. La technique dite du recuit consiste alors à réchauffer le matériau afin que les molécules quittent leur état d'énergie locale minimale, puis de reprendre la baisse de la température.

La méthode du recuit simulé, en optimisation combinatoire, va reprendre à sa manière ces différentes notions : les spécifications des variables seront assimilées aux coordonnées des molécules, la valeur de la fonction économique à l'énergie ; quant à la température, elle va donner l'idée d'une fonction de contrôle du protocole d'amélioration itérative, permettant notamment de calculer des probabilités de choix de solutions « dégradantes » (analogie avec l'augmentation provisoire de température).

Plus précisément, le principe général est le suivant :

Supposons que nous soyons en une solution $x_i$ et que la fonction de voisinage $g$ nous donne une solution $x_j = g(x_i)$.

On retiendra $x_i$ avec une probabilité égale à 1 si $f(x_j) < f(x_i)$ et égale à une certaine fonction $P$ si $f(x_j) > f(x_i)$.

Cette fonction $P$ est de la forme :

$$P = e^{\frac{-((f(x_j)-f(x_i)}{t}}$$

$t$ étant un paramètre entièrement contrôlé que l'on fait décroître tout au long du processus, et qui joue le rôle de la température dans le processus du recuit métallurgique (on continue d'ailleurs à appeler ce paramètre température dans la méthode du recuit simulé).

Au début du processus, $t$ étant élevé, $P$ est voisin de 1 et on aura tendance à accepter de nombreuses solutions qui dégradent $j$. À la fin du processus, $P$est voisin de 0 et on n'accepte plus que rarement cette circonstance.

La technique permettant de savoir si on retient $x_j$ à partir du moment où $f(x_j) > f(x_i)$ est très simple : à partir d'un programme de génération de nombre aléatoire (il en existe dans toutes les bibliothèques de programmes informatiques), on extrait une valeur $u$, réalisation de la variable uniforme sur $0 - 1$.

Si $u < P$ on accepte $x_j$ ; on ne l'accepte pas dans le cas contraire.

La technique du recuit simulé consiste à faire baisser la température par paliers, un certain nombre d'itérations ayant donc lieu à température constante.

La méthode du recuit simulé peut donc se décliner, d'une façon générale, de la façon suivante :

a) Initialisation

Partir d'une solution initiale $x_0 x_{min} = x_0, f_{min} = f(x_0)$. On se donne par ailleurs une fonction de voisinage $g$ (ensemble $V$), une température initiale $t_0$, une loi de

décroissance de la température $t_{i+1} = d(t_i)$, un nombre d'itérations $k$ à effectuer à température constante, un test d'arrêt.

b) Pour une température $t$ donnée :

On explore une suite de $x_j, x_j = g(x_i)$.

Si $f(x_j) < f(x_i))$, on retient $x_j$.

Si par ailleurs $f(x_j) < f_{min}$ alors $x_{min} = x_j \; et \; f_{min} = f(x_j)$

Si $f(x_j) > f(x_i)$ on calcule
$P = e^{\frac{-(f(x_j)-f(x_i))}{t}}$

On tire $u$ dans la distribution de la loi uniforme.

Si $u < P$ on retient $x_j$; sinon on ne le retient pas.

c) Lorsque l'on a atteint le nombre d'itérations $k$ d'une température donnée, on fait décroître cette dernière par la fonction $d(t)$ et on reprend en $b$).

La loi de décroissance de la température est souvent prise de la forme $h^t$, avec

$0 < h < 1$ (suite géométrique tendant vers 0).

Comme on le voit, la méthode du recuit simulé est gourmande en choix initiaux. Il s'agit en effet de choisir de façon efficace un certain nombre d'ingrédients :

- la fonction de voisinage
- la valeur initiale de $t$
- la fonction de décroissance de $t$
- le nombre d'itérations à $t$ constant
- la solution initiale
- le test d'arrêt

Un certain nombre de règles empiriques ont été préconisées par les spécialistes de ces méthodes, mais on ne peut pas dire qu'existe à l'heure actuelle un algorithme unique s'imposant à l'ensemble des problèmes que l'on cherche à résoudre. Il convient donc d'ajouter une condition nouvelle pour que ce type de technique se révèle efficace, à savoir l'expérience du modélisateur, seul susceptible, comme un grand chef dans sa cuisine, de doser ces divers ingrédients en fonction de la configuration du problème.

Cela dit, la méthode du recuit simulé a fait ses preuves sur des problèmes combinatoires de grande dimension, notamment des problèmes du type voyageur de commerce ou encore des problèmes d'ordonnancement d'atelier.

Un de ses avantages est qu'elle est facile à programmer.

Un inconvénient de la méthode est constitué par le nombre important d'ingrédients qu'il faut fixer au départ et qui, pour le moment, relèvent d'une expertise « pointue ».

La méthode Tabou, qui suit, n'a pas cet inconvénient.

### *8.2.3.2. La méthode Tabou*

La méthode Tabou, qui date de la fin des années soixante-dix, est, elle aussi, fondée sur un principe très simple : comme la méthode du recuit simulé, elle procède par itérations en allant de solution en solution, mais de son côté elle s'interdit de revenir à une solution déjà trouvée (d'où sa dénomination). Supposons alors qu'elle " tombe " en une solution qui est un minimum local. L'exploration du voisinage va donner des solutions moins bonnes ; si on accepte néanmoins l'une ce ces solutions, la moins mauvaise, le fait d'interdire la solution optimum local va nous permettre de sortir du " trou " constitué par le minimum local. On voit donc que, comme dans le cadre du recuit simulé, on accepte de dégrader la fonction économique provisoirement. Mais le principe adopté conduit à mettre en œuvre beaucoup moins d'ingrédients d'expertise que le recuit simulé. Si on traduit en effet le principe dans un algorithme, alors on voit qu'il suffit :

- de se donner une fonction de voisinage,
- de partir d'une solution initiale,
- à partir de la dernière solution trouvée $x_i$ à l'itération $i$, d'explorer le voisinage de cette solution, de prendre la meilleure solution dans ce voisinage (éventuellement moins bonne que la solution dont on part),
- d'ajouter à la liste des solutions interdites (dite liste Tabou $T$) la solution $x_i$,
- de continuer ainsi jusqu'à un test d'arrêt (en général portant sur le nombre d'itérations).

En fait, on n'applique pas directement cette procédure en général : en effet conserver en mémoire la liste $T$ des solutions rencontrées, trouver dans $V(x_i)$ la meilleure solution, comparer cette solution à toutes celles de la liste T prend beaucoup de place en mémoire et beaucoup de temps de calcul. Pour éviter ces inconvénients, on retient plutôt la transformation élémentaire qui permet de passer de $x_i$ à $x_j$ (par exemple changement d'un $0$ $en$ $1$ à une certaine place dans un codage en à $0-1$ d'une solution). Ce sont alors les transformations élémentaires inverses (les « mouvements ») qui vont être interdites. Mais cela comporte alors un nouvel inconvénient. Si je passe de $x_i$ à $x_j$ par une certaine transformation élémentaire $t_{ij}$ et de $x_j$ à $x_k$ par une autre transformation élémentaire $t_{jk}$, alors j'interdis, peut-être à tort de revenir de $x_k$ vers $x_i$ par un mouvement $t_{jk}^{-1}$ . On risque d'appauvrir ainsi considérablement le voisinage des solutions explorées. Pour pallier ce défaut, on limite la liste Tabou à une certaine taille, et on la gère par une procédure « first in - first out » : quand la liste $T$ est pleine, on supprime de $T$ le mouvement qui est le plus ancien et on le remplace par celui que l'on vient d'interdire.

Comme on le voit une autre différence de la méthode Tabou avec le recuit simulé est qu'elle est déterministe et qu'elle ne fait pas intervenir de module de recherche aléatoire. Cela dit, elle demande *a priori* que l'on explore le voisinage d'une solution $x_i$ de façon exhaustive. Cela peut prendre du temps de calcul ; aussi certaines adaptations de la méthode préconisent-elles de considérer un échantillon aléatoire du voisinage de $x_i$, auquel cas on retrouve des considérations qui ne sont pas très éloignées de la méthode précédente.

La taille de la liste Tabou est en général de l'ordre de la dizaine, mais là aussi d'autres choix sont possibles.

La méthode Tabou a été appliquée à de nombreux problèmes combinatoires difficiles. Citons par exemple le planning de cours ou le planning d'examens dans les universités, problème hautement combinatoire (affectation de cours ou d'examens à la fois à des salles et à des horaires) qui représente classiquement des " casse-tête " terrifiants pour les directions des études.

Un autre type d'heuristique actuellement en vogue se fonde sur des principes assez différents que ceux qui sont à la base des méthodes d'itération d'une solution à une autre par voisinage : il s'agit des algorithmes génétiques.

### 8.2.4. Les algorithmes génétiques

Les algorithmes génétiques portent leur nom à cause d'une analogie (les heuristiques sont friandes en métaphores) avec les phénomènes d'évolution des espèces, et plus précisément des processus de mutation et de reproduction des individus. Comme on le sait, ces processus vont mettre en cause les gènes et les chromosomes de ces individus, agissant soit par mutation (changement d'un gène dans un chromosome) soit par substitution de séquences de gènes à d'autres (croisement) lors de la reproduction. Les algorithmes génétiques sont alors fondés sur l'idée suivante : dans la mesure où les processus biologiques conduisent à une certaine adaptation des espèces à leur milieu, la descendance d'individus adaptés va probablement être également adaptée, et peut-être en mieux. Si on fait l'analogie avec les solutions d'un problème combinatoire et si l'on peut engendrer des solutions nouvelles à partir de solutions existantes, conservant les « bonnes » caractéristiques des premières, alors on disposera de solutions progressivement meilleures.

Par différence avec les méthodes que nous venons de voir, les algorithmes génétiques vont examiner non pas une solution l'une après l'autre mais un ensemble de solutions $x_1 \dots x_j, x_N$, appelée population, de taille $N$ en général fixée à l'avance. Chaque solution est codée dans un alphabet donné de caractères, qui sont souvent les caractères $0\ ou\ 1$ (programme linéaire en nombres entiers par exemple) mais qui peuvent être d'une autre nature (liste des arcs par exemple dans le problème du voyageur de commerce ou encore numéros des machines sur lesquelles doivent passer des pièces dans un problème d'ordonnancement). La séquence des caractères d'une solution, par analogie avec les processus biologiques, sera appelée chromosome de la solution et un caractère donné gène.

Puisque l'on veut sélectionner des individus " forts " afin qu'ils aient une descendance également ou encore plus forte, il faut un indicateur de cette force : on prend tout simplement la valeur de la fonction économique pour l'individu - la solution - considéré. Plus cette valeur est faible (on a choisi de minimiser $j$), plus la solution mérite d'être sélectionnée dans le processus de reproduction.

On peut énoncer alors la forme générale d'un algorithme génétique (il existe cela dit de nombreuses variantes et on peut toujours en inventer de nouvelles) :

1) Créer une population initiale $P(1)(t = 1)$ de $N$ solutions codées chacune par un chromosome.

2) Calculer la valeur de la fonction économique $f(x_i)$ pour chaque solution $x_i$ de $P(t)$. La force est du type $K - f$ , où $K$ est une constante.

3) On sélectionne $N$ solutions dans $P(t)$ en utilisant une loi aléatoire. Une solution peut donc apparaître plusieurs fois. On fait en sorte que plus elle est forte, plus elle a de chances d'apparaître. On obtient un ensemble $S(t)$ de solutions sélectionnées.

4) Croisement : on apparie deux à deux les solutions de $S(t)$. Pour chaque paire, avec une probabilité $p_c$ , on applique une fonction de croisement : les deux solutions vont donner deux solutions « filles » dont les chromosomes seront des combinaisons des chromosomes des « parents ». Les deux filles sont placées dans un nouveau sous ensemble $R(t)$. Avec la probabilité $1 - p_c$, ce sont les parents qui sont placés dans $R(t)$.

5) Mutation : pour chaque individu (solution) de $R(t)$, on applique une fonction de mutation, avec la probabilité $p_m$. On le laisse intact avec la probabilité $1 - p_m$. On recopie tous les individus, mutés ou non, dans $P(t + 1)$.

6) On retourne en 2) en passant à l'itération $t + 1$ et on continue jusqu'à un critère d'arrêt (en général le nombre d'itérations).

Comme on le voit, il y a ici un nombre non négligeable d'ingrédients à choisir. Donnons quelques précisions sur chacun d'eux :

- **Codage**
  On a déjà évoqué ce point. Dans beaucoup de cas (mais pas dans tous) on trouvera un codage en $0 - 1$. Par exemple, les solutions du problème de sac à dos se codent spontanément de cette façon. **Remarque** : quand le problème comporte des variables à valeurs réelles, ayant une certaine précision, on peut toujours se ramener à un codage en $0 - 1$, mais il s'agit d'opérations souvent très lourdes. De même, quand les variables sont entières, on peut prendre leur codage binaire ; le chromosome sera constitué par la séquence des codages binaires.

- **Sélection**
  La « force » d'une solution, on l'a dit, est du type $F(x) = K - f(x)$ dans le cas d'une minimisation. En général, on prend pour $K$ le maximum observé pour $f(x)$, soit dans la population considérée pour la sélection, soit depuis le début de la procédure (de telle sorte que $F$ est positive).

La sélection prend alors la forme d'une loterie ; chaque solution $x_i$ va avoir la probabilité:

$$p(x_i) = \frac{F(x_i)}{\sum_{j=1}^{N} F(x_j)}$$

d'être choisie.

Le procédé pour ce faire est très simple :
On définit les bornes de la fonction de répartition liée à la distribution $p$ :

$$l(x_i) = \sum_{k=1}^{i} p(x_k)$$

On tire (cf ci-dessus pour le recuit simulé) un nombre aléatoire $u$ compris entre 0 $et$ 1. Si $l(x_{i-1} < u < l(x_i)$
alors on sélectionne la solution $x_i$.
On répète $N$ fois cette procédure de façon à obtenir toujours $N$ individus. Un individu fort sera en principe (mais le hasard peut s'y opposer) sélectionné plusieurs fois.

- **Croisement**
Dans l'étape 4 introduite ci-dessus, on apparie deux à deux les solutions sélectionnées par la procédure précédente. Soit une telle paire de solutions, avec leurs chromosomes respectifs $a\ et\ b$. Cette paire de parents va engendrer une paire d'enfants, par l'intermédiaire d'une certaine combinaison des chromosomes (opérateur de croisement).

Il existe plusieurs opérateurs de croisement (et à la limite on peut en imaginer plein d'autres). En voici quelques uns :

a) croisement « un point » : supposons que les chromosomes soient constitués par une suite de $p$ caractères, les gènes, pris dans un même vocabulaire. On tire un nombre aléatoire entier compris entre 1 $et\ p$. Soit $l$ ce nombre.

Avant le croisement les deux chromosomes s'écrivent :
$a = (a_1 \ldots\ldots\ldots\ldots a_l, a_{l+1}, \ldots a_p)$
$b = (b_1 \ldots\ldots\ldots\ldots b_l, b_{l+1}, \ldots b_p)$

Après le croisement, les chromosomes des enfants sont :
$a' = (a_1 \ldots\ldots\ldots\ldots a_l, b_{l+1} \ldots b_p)$
$b' = (b_1 \ldots b_l, a_{l+1} \ldots a_p)$

b) croisement $k\ points$ : on génère aléatoirement non un nombre compris entre 1 $et\ p$ mais $k$, créant ainsi $k+1$ sous-chromosomes. Le premier enfant aura

pour chromosome la concaténation, dans l'ordre des sous-chromosomes, des sous-chromosomes de numéro impair du premier parent et des sous-chromosomes de numéro pair du second (situation inverse pour le second enfant). Par exemple, pour un croisement « 3 points » :

Parents : $a = (00/1010/10/10)$

$b = (10/1001/01/01)$

Enfants : $a' = (00/1001/10/01)$

$b' = (10/1010/01/10)$

c) croisement uniforme : on prend un des deux parents, et on répète $p$ fois l'opération suivante : on tire au hasard un deux enfants (avec la probabilité $\frac{1}{2}$) et on place à la jème position de l'enfant le $j^{ème}$ gène du parent, cela pour $j = 1,2 \ldots p$. Pour l'enfant non tiré au sort, on place en $j^{ème}$e position le $j^{ème}$ gène de l'autre parent.

Ces trois opérateurs de croisement ont la propriété suivante : si les deux parents ont le même gène à la même position, cette caractéristique va se reproduire sur les enfants. On a là la traduction de l'idée intuitive qui guide ce type de méthode, à savoir que l'on essaye de garder ce qui fait la force des individus (des solutions).

- **Mutation**

  L'opérateur de mutation, appliqué sur les enfants obtenus à l'étape précédente, doit permettre de créer une certaine diversité dans les populations engendrées. En effet, la propriété des croisements que l'on vient d'évoquer (conservation des gènes identiques des parents pour leurs descendants) a malgré tout l'inconvénient potentiel de conduire à des individus " faussement forts " (minimum local dans le langage des solutions) et de ne pas pouvoir en sortir. L'opérateur de mutation le plus banal consiste à choisir aléatoirement un gène et à changer sa valeur, mais cette opération est surtout valable pour des codages du type $0 - 1$. Pour d'autres problèmes, il faudra adapter l'opérateur (par exemple pour le problème du voyageur de commerce on pourra prendre la transformation 2-opt décrite plus haut).

Par ailleurs, on peut considérer que la mutation suit un objectif contradictoire avec celui du croisement, d'après ce que l'on vient de dire. Donc, comme on vise avant tout le perfectionnement des descendants par le mariage des solutions, on appliquera l'opérateur de mutation avec une probabilité faible.

De nombreuses variantes, encore une fois, ont été testées sur ces algorithmes génétiques; on peut notamment mixer les heuristiques que l'on vient de voir, par exemple en combinant algorithme génétique et recuit simulé (modification aléatoire d'une solution). De même, les opérateurs de croisement et de mutation doivent être adaptés au type de

problème abordé : ce ne seront pas les mêmes pour un problème d'ordonnancement d'atelier et pour un problème de voyageur de commerce.

Compte tenu de la nature de ces méthodes, il n'est pas très étonnant que peu de résultats généraux existent sur leurs performances. Citons cependant la théorie des schémas, qui a donné lieu à quelques résultats de ce type. Mais là encore, la mise en route de ces heuristiques nécessite une solide expérience en la matière (fixation de nombreux ingrédients : codage, taille de la population, probabilités et opérateurs de croisement et de mutation, nombre d'itérations etc.).

Les algorithmes génétiques, comme les autres, se sont attaqués non sans succès à de très difficiles problèmes combinatoires, comme ceux d'ordonnancement d'atelier. La recherche est très active dans le domaine et consiste notamment, sur des problèmes de taille importante, à comparer les performances des différentes méthodes que nous venons d'exposer. Il y en a d'autres, d'ailleurs (citons la technique également connue des « réseaux neuronaux », qui nous semble moins centrale pour les problèmes qui nous occupent ici).

Un des avantages de ce type de méthode est le peu de savoir mathématique qu'elles présupposent. Alors que les méthodes d'optimisation connues en mathématiques reposent sur des connaissances étendues (lagrangiens, hessiens, conditions de Kuhn et Tucker, etc.) demandant un certain effort de compréhension et des bases solides en mathématiques (tout en étant en général peu adaptées aux problèmes particuliers mobilisés par la recherche opérationnelle, sauf exception -on a évoqué un exemple d'importation de ces méthodes classiques avec les algorithmes de point intérieur pour la programmation linéaire), les algorithmes génétiques exigent très peu de « back ground » de ce type. Mais c'est là une des caractéristiques générales des méthodes de la recherche opérationnelle, comme on l'a souligné en introduction.

Après ce survol des problèmes combinatoires difficiles, nous allons revenir à des problèmes moins « chevelus » car relevant d'algorithmes polynomiaux : les problèmes de flots.

# Chapitre 9 • Problèmes de flots

## 9.1. FLOTS – DEFINTION

### 9.1.1. Flots

Soit un graphe $G(X, U)$ donné par ses sommets $x_1, x_2 \dots x_n \in X$ et ses arcs $u_1, u_2 \dots u_m \in U$

On appelle flot dans ce graphe un vecteur $\emptyset = (\emptyset_1, \emptyset_2, \dots \emptyset_m)$ tel que :

1) toute composante $\emptyset_i$ est un nombre entier que l'on appelle flux dans l'arc $v_i$

2) la loi de Kirchhoff (loi aux nœuds) est vérifiée en tout sommet $x$, c'est-à-dire que : si $U_x^-$ représente l'ensemble des arcs incidents intérieurement à $x$ (ayant $x$ comme extrémité terminale) et $U_x^+$ l'ensemble des arcs incidents extérieurement à $x$ (ayant $x$ comme extrémité initiale), on a :

$$i/u_i \in U_x^- \qquad i/u_i \in U_x^+$$

### 9.1.2. Réseaux de transport

Pour relier la notion de flot à des phénomènes réels, il convient auparavant de définir ce que l'on entend par réseau de transport.
Un réseau de transport est un $1 - graphe$ sans boucle où à chaque arc $u$ est associé un nombre entier $c(u) \geq 0$ (que l'on appellera capacité de l'arc $u$) et où :

1) il existe un sommet $x_0$ et un seul tel que $U^-(x_0) = 0$ ($x_0$ n'a pas de précédent) qu'on appellera entrée du réseau.

2) il existe un sommet $z$ et un seul tel que $U^+(z) = 0$ ($z$ n'a pas de suivant) qu'on appellera sortie de réseau.

À un réseau de transport est donc associé le type de représentation suivant :

La notion de flots peut être associée, on le voit, à de multiples problèmes qui consistent à faire passer des quantités de matières sur les arcs d'un réseau de transport tout en essayant de faire respecter au mieux certains critères. Remarquons qu'elle est aussi très utilisée dans d'autres disciplines que la recherche opérationnelle (hydraulique et électricité par exemple).

Nous allons examiner le problème essentiel relatif à cette notion : celui du flot maximal.

## 9.2. PROBLEME DU FLOT MAXIMAL

Considérons un réseau de transport de sommet $x_0, x_1, \dots x_n, z$ et les arcs $u_1, u_2, \dots u_m$ $de$ de capacités $c(u_1), c(u_2), \dots c(u_m)$ entières et positives. Nous adjoindrons à ce réseau un arc fictif $\overrightarrow{x_0}$, de capacité infinie.

Le problème consiste à établir un flot sur le graphe de telle sorte que le flux total arrivant en $z$ soit le plus grand possible avec pour tout $U$ :

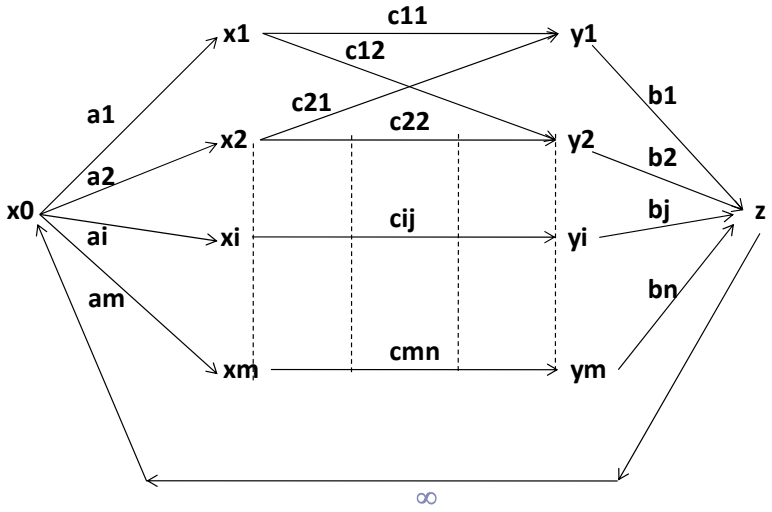
$$0 \leq \phi(u) \leq c(u)$$

si Ø est le flux sur l'arc $u$. Un flot obéissant à ces contraintes est appelé *flot compatible*.

On voit que l'addition de l'arc $zx_0$ permet d'avoir la loi des nœuds vérifiée en tous les sommets.

**Exemple de problème de flot maximal**

On a un certain nombre d'entrepôts $x_1, x_2 \dots x_m$, avec des quantités d'une certaine marchandise $a_1, a_2 \dots a_m$. Il s'agit de transporter cette marchandise en des points $y_1, y_2, \dots y_n$ où existent des demandes $b_1, b_2, \dots b_n$. De l'entrepôt $i$ au demandeur $j$, la quantité maximale que l'on peut transporter est désignée par $c_{ij}$. On veut satisfaire globalement la demande au maximum, c'est-à-dire que l'on cherche la quantité maximale de marchandises à transporter, compte tenu des disponibilités, des demandes et des limitations $c_{ij}$. On voit que le problème consiste à chercher le flot qui maximise le flot arrivant en $z$ dans le réseau de transport ci-dessous, où à chaque entrepôt correspond un sommet $x_i$ et à chaque destination un sommet $y_j$ ; le sommet $x_i$ est relié au sommet $y_j$ par arc de capacité $c_{ij}$ : $x_0$ est relié à $x_i$ par un arc de capacité $a_i$ et $y_j$ est relié à $z$ par un arc de capacité $b_j$.



Formellement, on peut écrire le problème du flot maximal ainsi : posons, pour simplifier $z = x_{n+1}$ et $c_{ij} = 0$ si l'arc $x_i x_j$ n'existe pas. Nous appellerons maintenant $x_{ij}$ le flux sur l'arc reliant $x_i$ à $y_j$. on doit avoir :

$0 \leq x_{ij} \leq c_{ij} \quad \forall i \; et \; \forall j$

et pour respecter la loi des noeuds :

$$\sum_{j=1}^{n} x_{ij} \leq a_i \qquad \forall_i = 1 \ldots . m$$

$$\sum_{i=1}^{m} x_{ij} \leq b_j \qquad \forall_j = 1 \ldots . n$$

$x_{ij} \geq 0 \qquad i = 1 \ldots m, \qquad j = 1 \ldots n$

Par ailleurs, il s'agit de maximiser sous ces contraintes la quantité :

$\phi_z = x_{n+1,0}$

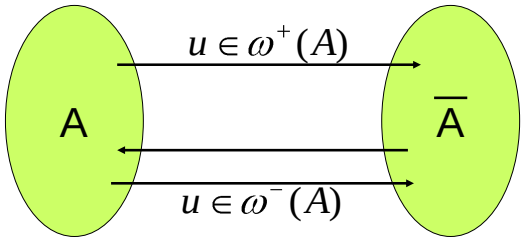
(flux sur l'arc de retour $zx_0$)

On voit donc que l'on a affaire ici à un programme linéaire, mais qui risque d'être très lourd à résoudre : en effet, si le réseau possède 20 sommets de part et d'autre par exemple, le programme linéaire portera sur $20 \times 20 = 400$ inconnus et aura $400 + 40 = 440$ contraintes. Or, un réseau de cette dimension est petit comparé à la majorité de ceux que l'on traite dans les applications courantes. On peut évidemment le simplifier en ne prenant qu'une variable - flux par arc existant, ce que nous n'avons pas fait pour simplifier les notations, mais de toute façon, l'algorithme du simplexe est beaucoup trop pénible pour ce type de problèmes, et nous allons donner une méthode de résolution plus élégante : l'algorithme de **Ford-Fulkerson.**

Auparavant, mettons en évidence un certain nombre de résultats qui nous seront utiles.

a) Définition d'un cocycle.

Considérons un graphe orienté $G(X, U)$. Soit $A$ un sous-ensemble de $X$, on appellera $\omega^+(A)$ l'ensemble des arcs ayant leur extrémité initiale dans $A \in X$ et leur extrémité terminale hors de $A$. De même $\omega(A)$ sera l'ensemble des arcs ayant leur extrémité initiale hors de A et leur extrémité terminale dans A. L'ensemble d'arcs $\omega(A) = \omega^+(A) \cup \omega^-(A)$ sera appelé un *cocycle du graphe* $G(X, U)$.




Les arcs $u \in \omega^+(A)$ seront dits orientés dans le sens « sortie ». Les arcs $\omega^-(A)$ seront dits orientés dans le sens « entrée ».

Quant aux cycles, on les a déjà définis et utilisés pour des graphes non orientés. Mais on peut aussi définir des cycles sur un graphe orienté : il suffit de désorienter les arcs. Un cycle sur un graphe orienté est donc constitué d'une suite d'arcs qui ne sont pas forcément orientés dans le même sens.

Démontrons le lemme suivant :

b) Lemme de MINTY

Ce lemme, qui *a priori* n'a rien à voir avec les flots, mais nous sera très utile par la suite, s'énonce de la façon suivante :

Soit un graphe $G(X, U)$, dont on colorie arbitrairement certains arcs soit en noir, soit en rouge, soit en vert (les autres restant incolores). On suppose qu'il existe un arc noir que nous appellerons $u_0$. Alors une seule des propositions suivantes est vérifiée :

a) il passe par $u_0$ un cycle ne contenant pas d'arcs incolores, avec tous les arcs noirs orientés dans le même sens, tous les arcs verts orientés dans le sens contraire et les arcs rouges orientés dans un sens quelconque.

b) $u_0$ appartient à un cocycle ne contenant pas d'arcs rouge avec tous les arcs noirs orientés dans le même sens et tous les arcs verts orientés dans le sens contraire.

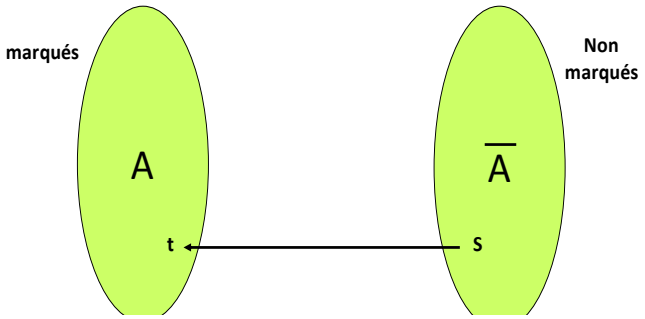
**Démontrons ce lemme :**
Appelons respectivement $s$ et $t$ l'extrémité initiale et l'extrémité terminale de $u_0$.

1. on marque $t$
2. si le sommet $i$ est marqué, et si l'arc $ij$ est noir, alors on marque $j$
3. si le sommet $j$ est marqué, et si l'arc $ij$ est vert, alors on marque $i$
4. si $i$ ou $j$ est marqué, et si l'arc $ij$ est rouge, alors on marque $j$ ou $i$.

On réitère cette procédure jusqu'à ce qu'on ne puisse plus marquer de sommet. Deux cas alors se produisent :

a) Par cette procédure de marquage, on arrive à marquer $s$. Ceci veut dire qu'il existe au moins une chaîne de sommets marqués entre $t$ et $s$ (donc un cycle partant de $t$ si on ajoute l'arc $u_0$), telle que tout sommet sur cette chaîne est marqué grâce au sommet précédent. Il n'y a donc pas d'arc incolore sur ce cycle. Par ailleurs, les arcs noirs sont tous dans le même sens (sens $s\ t$) et les arcs verts dans le sens contraire. Les arcs rouges sont dans un sens quelconque.

b) On ne parvient pas à marquer $s$ par cette procédure de marquage appliquée systématiquement. Appelons alors $A$ l'ensemble des sommets du graphe marqués et $\overline{A}$ l'ensemble des sommets non marqués.




Prenons alors le cocycle joignant $A$ $et$ $\overline{A}$. Tous les arcs noirs de ce cocycle sont dans le même sens $(\overrightarrow{s,t})$ ; sinon, s'ils allaient de $A$ dans $\overline{A}$, on aurait pu marquer les sommets correspondants de $\overline{A}$, ce qui n'a pas été le cas. De même, tous les arcs verts du cocycle sont également dans le même sens $(\overrightarrow{t,s})$ sinon, s'ils étaient dans le sens contraire $(\overrightarrow{s,t})$ on pourrait marquer leur extrémité initiale dans $\overline{A}$.

Enfin, il ne peut y avoir d'arc rouge joignant $A$ $et$ $\overline{A}$ ou $\overline{A}$ à $A$, car il entraînerait de toute façon un marquage dans $A$. Il peut y avoir en revanche dans le cocycle des arcs incolores.

**Le lemme de Minty est ainsi démontré.**

c) Flot maximal et coupe de capacité minimale

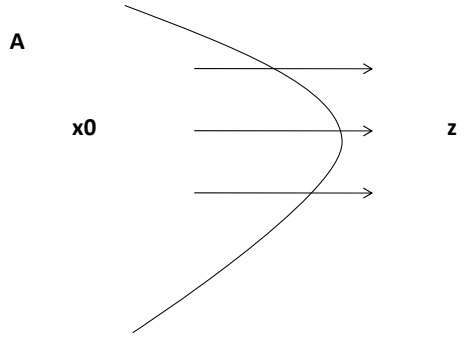
Revenons maintenant aux flots et aux réseaux de transport. Soit un réseau $G(X,U)$, d'entrée $x_0$, de sortie $z$, et d'arc de retour $\overrightarrow{zx_0}$ et de capacités des arcs $uc(u)$.

Prenons un sous-ensemble $A$ de sommet $G$ tel que $x_0 \in$A et $z \notin A$. Soient, suivant les notations précédentes $\omega^+(A)$ l'ensemble des arcs partant de $A$ et $\omega^-(A)$ l'ensemble des arcs arrivant dans $A$.

On appellera coupe dans le réseau $G(X,U)$ l'ensemble d'arcs $\omega^+(A)$. On appellera capacité de la coupe la quantité

$$C(A) = \sum_{u \in \omega^+(A)} c(u)$$

C(u)étant la capacité de l'arc u




Soit maintenant un flot compatible avec les contraintes $0 \leq \emptyset(u) \leq c(u)\ \forall u$

Prenons un sommet $x$ quelconque de $A$. Sur ce sommet, la loi aux noeuds s'inscrit :

$$\sum_{u \in U_x^-} \phi(u) - \sum_{u \in U_x^+} \phi(u) = 0$$

Sommons ces égalités sur tous les sommets $x \in A$. Les flux sur les arcs ayant leurs deux extrémités dans $A$ vont s'éliminer; il restera, si $\emptyset_z$ est le flot sur l'arc retour :

$$\sum_{u \in \omega^-(A)} \phi(u) + \phi_z - \sum_{u \in \omega^+(A)} \phi(u) = 0$$

Soit

$$\phi_z = \sum_{u \in \omega^+(A)} \phi(u) - \sum_{u \omega^-(A)} \phi(u)$$

Comme $\emptyset(u) \geq 0\ \forall u$, on peut écrire :

$$\phi_z \leq \sum_{u \in \omega^+(A)} \phi(u)$$

et encore, puisque $\varphi(u) \leq c(u) \forall u$

$$\phi_z \leq \sum_{u \omega^+(A)} c(u)$$

$\varphi_z \leq C(A)\ \forall \emptyset\ et\ \forall A$

Quels que soient le flot et l'ensemble $A$ choisis, on a donc toujours cette inégalité. On est alors certain que si l'on trouve un flot $\varphi$ et un ensemble $A$ tel que :

$$\phi_z = C(A)$$

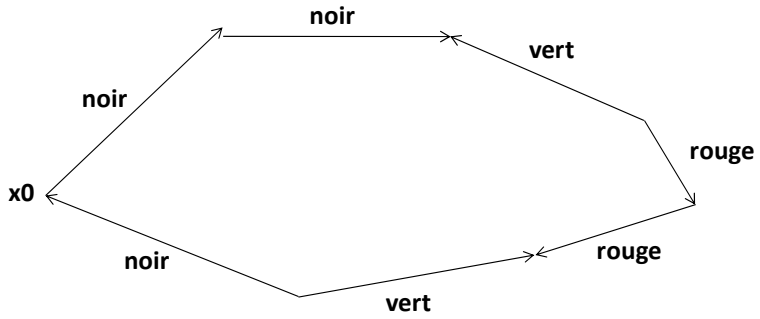
alors on a trouvé à la fois le flot maximal et la coupe de capacité minimale.

Précisons ce point en démontrant le résultat suivant qui va nous conduire en même temps à l'algorithme permettant de trouver le flot optimal :

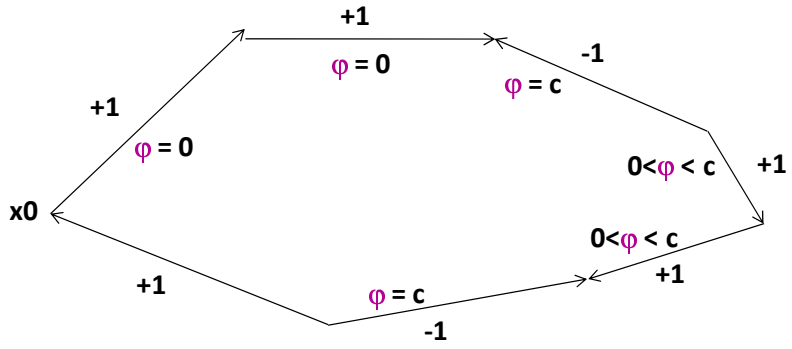
**<u>Théorème</u> : La valeur maximale d'un flot de $z$ à $x_0$ dans $G(X,U)$, muni de capacités $c(u)$ est égale à la capacité d'une coupe de capacité minimale.**

Supposons en effet, un flot $\emptyset$ tel que $\emptyset_z$ soit maximal. Colorions les arcs de $G$ de la façon suivante :
- l'arc de retour $zx_0$ est colorié en noir,
- un arc $u$ tel que $\emptyset(u) = 0$ est colorié en noir,
- un arc $u$ tel que $0 < \emptyset(u) < c(u)$ est colorié en rouge,
- un arc $u$ tel que $\emptyset(u) = c(u)$ est colorié en vert.

Appliquons le lemme de Minty. Montrons que nous ne sommes pas dans le cas a) de ce lemme. En effet, supposons que l'on soit dans le cas a) : il existe alors un cycle partant de $x_0$ avec tous les arcs noirs dans le sens $\overrightarrow{zx_0}$ tous les arcs verts dans le sens contraire et les arcs rouges dans un sens quelconque. Par exemple :

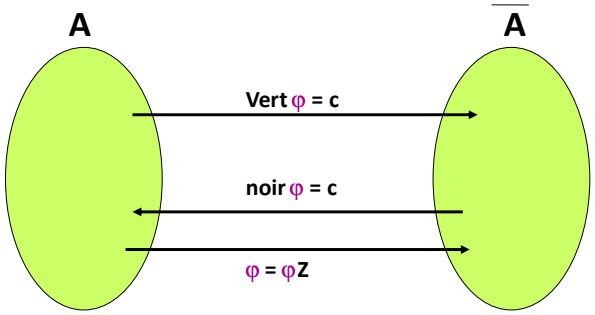



Mais d'après le coloriage choisi sur ce cycle, on peut (en conservant la loi des nœuds) augmenter les flux sur les arcs noirs d'une unité, diminuer d'une unité les flux sur les arcs verts, augmenter ou diminuer d'une unité les flux sur les arcs rouges suivant le sens de ces derniers. On vérifie aisément que les lois des nœuds restent effectivement vérifiées et les contraintes de capacité également (remarquons ici le caractère nécessaire d'intégrité des capacités).




Ainsi $\emptyset_z$ augmenterait d'une unité et le flot ne serait pas optimal.

On se trouve donc forcément dans le cas b) du lemme de Minty : $zx_0$ appartient à un cocycle séparant deux ensembles de sommets $A$ $et$ $\overline{A}$.




Les arcs noirs sont tous dans le même sens ($\overrightarrow{zx_0}$) et les arcs verts dans le sens contraire ($\overrightarrow{x_0z}$); il n'y a pas d'arc rouge appartenant au cocycle. Mais les arcs noirs (sauf $\overrightarrow{x_0z}$) ont un flux nul, les arcs verts un flux égal à la capacité de l'arc, et les équations de conservation aux nœuds donnent :

$$\phi_z = \sum_{u\in\omega^+(A)} \phi(u) - \sum_{u\in\omega^-(A)} \phi(u)$$

$$\phi_z = \sum_{u\in\omega^+(A)} \phi(u) = \sum_{u\in\omega^+(A)} c(u) = C(A)$$

Le théorème est donc démontré. Il fournit en même temps l'algorithme de détermination du flot maximal, algorithme de Ford Fulkerson.

Ce dernier consiste en effet à faire d'abord passer dans le graphe un flot quelconque. Puis on effectue le coloriage du théorème précédent. Si on arrive à trouver un cycle partant de

$x_0$ avec tous les arcs noirs dans le même sens et tous les arcs verts dans le sens contraire, on améliore le flot en changeant les flux sur les arcs du cycle. Si l'on ne trouve pas de tel cycle, on est arrivé au flot optimal.
Plus précisément, la recherche du cycle améliorant se fait grâce au marquage de Minty.

$i$ marqué, $\overrightarrow{ij}$ noir ($\emptyset(i,j) = 0$) on marque $j$,
$j$ marqué, $\overrightarrow{ij}$ vert ($\emptyset(i,j) = c(i,j)$) on marque $i$,
$i$ ou $j$ marqué, $\overrightarrow{ij}$ rouge ($0 < \emptyset(i,j) < c(i,j)$), on marque $j$ ou $i$,
Si l'on marque $z$, c'est qu'on a obtenu un cycle améliorant.

Remarquons qu'on peut simplifier la procédure de marquage :
$i$ marqué, ($\emptyset(\overrightarrow{ij}) < c(\overrightarrow{ij})$) on marque $j$ (arc $\overrightarrow{ij}$ noir ou rouge),
$j$ marqué, (Ø(ij)>0), on marque $i$ (arc $\overrightarrow{ij}$ vert ou rouge).

L'algorithme de Ford-Fulkerson est alors :

a) Faire passer un flot au jugé (c'est toujours possible : le flot nul est un flot réalisable)

b) Appliquer sur ce flot le marquage de Ford-Fulkerson : $x_0$ marqué.

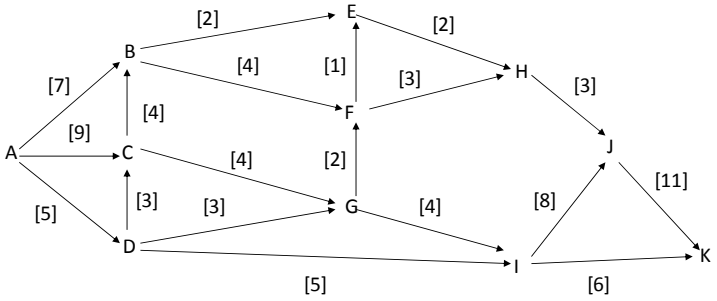
Si $i$ marqué, ($\emptyset(\overrightarrow{ij}) < c(\overrightarrow{ij})$) on marque $j$.

Si $j$ marqué, et ($\emptyset(\overrightarrow{ij}) > 0$) on marque $i$.

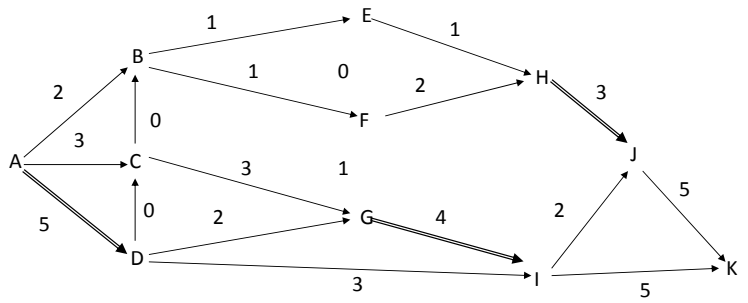
c) si on trouve un cycle de sommets marqués on peut améliorer le flot en augmentant le flot sur les arcs parcourus dans le sens ($\overrightarrow{x_0 z}$) et en diminuant le flot sur les arcs parcourus dans le sens contraire, et cela d'une même quantité. On retourne alors en b).

d) si on ne trouve pas de tel cycle, le flot est optimal.

Appliquons cet algorithme sur le graphe suivant, où on a omis l'arc de retour. La première figure (figure 1) donne les capacités, la figure suivante (figure 2) un flot au jugé.




*Les capacités*



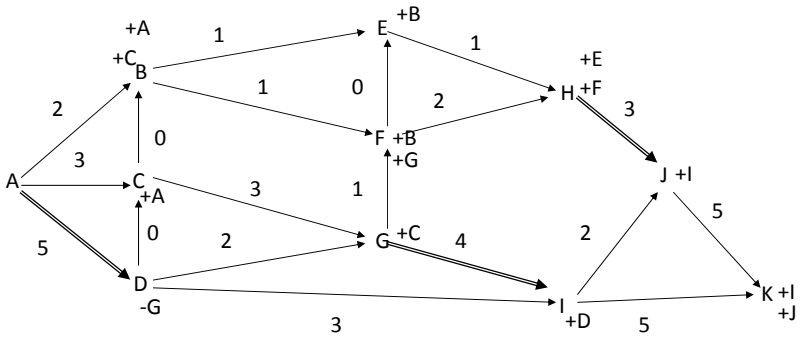

*Le flot au jugé (les traits doubles représentent les arcs satures)*

Appliquons maintenant le marquage de Ford-Fulkerson en précisant ce marquage de la façon suivante (pour savoir à partir de quel sommet un sommet donné a été marqué).

Si $j$ est marqué à partir de $i$ parceque $\emptyset(\overrightarrow{ij}) < c(\overrightarrow{ij})$ alors $j$ sera marqué par $+i$

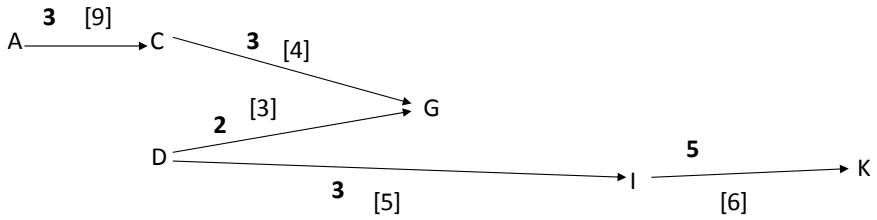
Si $i$ est marqué à partir de $j$ parceque : $\emptyset(\overrightarrow{ij}) > 0$, alors $i$ sera marqué par $-j$

On obtient alors le marquage suivant :




(on omet ici les marquages inutiles, c'est-à-dire ceux consistant à marquer $j$ parce que $i$ est marqué avec $\emptyset(\overrightarrow{ij}) < c(\overrightarrow{ij})$ puis à remarquer $i$ parce que $\emptyset(\overrightarrow{ij}) > 0$.

On voit que l'on marque $K$, donc que le flot n'est pas optimal; on marque $K$ à partir de plusieurs chaînes possibles. Prenons-en une :

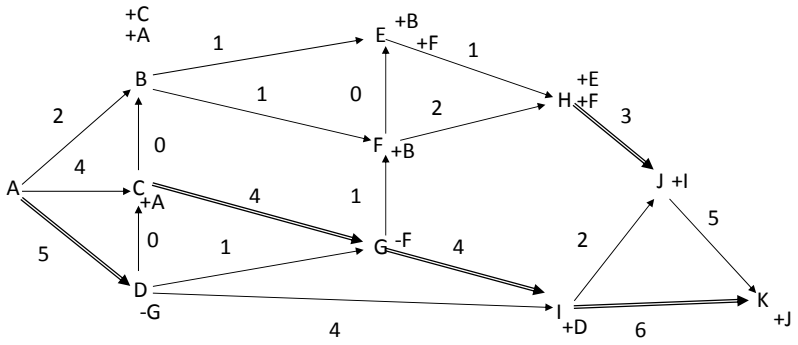


On a indiqué en les entourant les capacités des arcs.

On peut améliorer le flot en jouant uniquement sur cette chaîne. Sur l'exemple, on voit bien que l'on peut ajouter un flot égal à +1 sur les arcs $\overrightarrow{AC}, \overrightarrow{CG}, \overrightarrow{DI}, \overrightarrow{IK}$, et retrancher une unité également à l'arc $\overrightarrow{DG}$ (loi des nœuds). Le flot arrivant en $K$ augmente d'une unité.

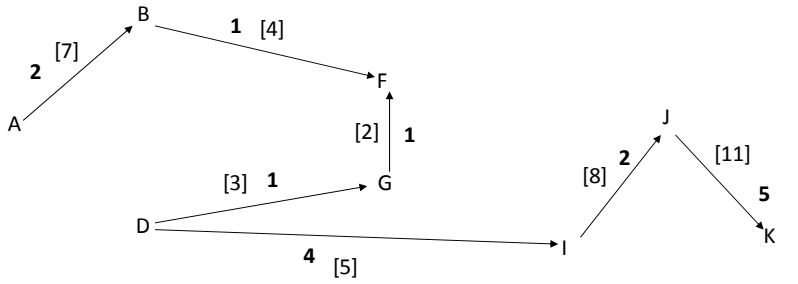
Cela dit, on peut essayer d'augmenter davantage le flot, en ajoutant et en retranchant des flots aux arcs jusqu'à ce que, soit sur un arc parcouru dans le sens $AK$ le flot atteigne la capacité, soit sur un arc parcouru dans le sens contraire, le flot devienne nul.

On voit qu'ici en fait on est tout de suite limité par la capacité de l'arc $\overrightarrow{CG}$, égale à 4 alors que le flot existant est 3. On ne peut donc, à partir de cette chaîne qu'augmenter le flot d'une unité. On obtient ainsi le nouveau flot.

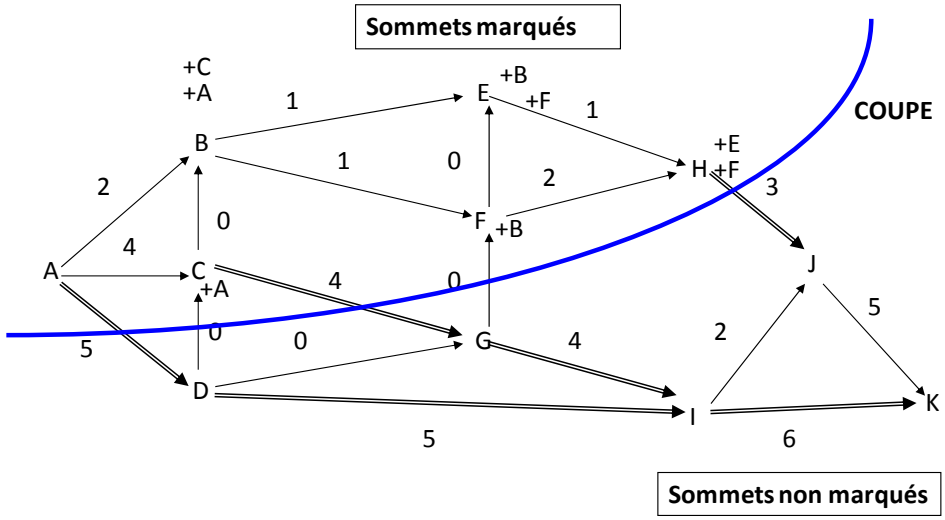



Sur ce même graphe, on a indiqué le nouveau marquage, qui permet de marquer $K$, donc d augmenter encore le flot total.

Prenons une nouvelle chaîne.




Sur cette chaîne également, on ne peut augmenter le flot arrivant en $K$ que d'une unité, à cause de l'arc $\overrightarrow{GF}$, parcouru dans le sens $\overrightarrow{KA}$ et qui porte un flot d'une unité. On obtient le flot :




Cette fois-ci, on ne peut plus marquer le sommet $K$. On a donc obtenu le flot optimal, avec un flot arrivant en $K$ égal à 12. En isolant les sommets marqués des sommets non marqués, on fait apparaître la coupe de capacité minimale (cf. figure 5), qui est constituée par les arcs $\overrightarrow{AD}$, $\overrightarrow{AG}$, $\overrightarrow{HJ}$. On vérifie que les arcs sortants sont saturés et que les arcs entrants $\overrightarrow{DC}$ et $\overrightarrow{GF}$ portent un flot nul.

On démontre facilement que la complexité de l'algorithme de Ford-Fulkerson est du type $O(m \times cmax), m$ étant le nombre d'arcs et $cmax$ la capacité maximale des arcs du réseau. En ce sens, on n'a pas affaire à strictement parler à un algorithme polynomial, puisque la complexité ne dépend pas de la taille du problème (spécifiée par $m$ $et$ $n$) mais également d'une donnée du problème, heureusement entière. On dit que l'on a affaire à un algorithme pseudo-polynomial. Il existe cela dit d'autres algorithmes voisins polynomiaux.

## 9.3. PROBLEME DU FLOT COMPATIBLE

Soit maintenant un réseau $G(X, U)$ avec des contraintes de capacité différentes de celles prises en charge auparavant, c'est-à-dire qu'on affecte à présent deux nombres entiers positifs ou nuls $b(u)$ et $c(u)$ à chaque arc $u$, et on impose aux flots $\emptyset$ sur le graphe $G(X, U)$ d'avoir toutes ses composantes $\emptyset(u)$ sur les arcs $u$ telles que

$$b(u) \leq \phi(u) \leq c(u)$$

Ce type de contrainte intervient par exemple lorsqu'on doit impérativement satisfaire certaines demandes. On peut sur ce type de graphe résoudre le problème du flot maximal, mais l'initialisation de l'algorithme de Ford-Fulkerson ne se fait pas aussi facilement qu'auparavant : il faut trouver un flot compatible avec les contraintes ci-

dessus. En particulier, le flot nul n'est pas compatible. Comment trouver un flot compatible ?

On n'est pas certain d'abord qu'un tel flot existe. Etablissons tout de suite une condition nécessaire d'existence. Pour tout sous-ensemble de sommets $A \subset X$, on a d'après les équations de conservation aux nœuds :

$$\sum_{u \in \omega^+(A)} \phi(u) - \sum_{u \in \omega^-(A)} \phi(u) = 0$$

pour tout flot compatible. Ce qui impose :

$$\sum_{u \in \omega^+(A)} c(u) - \sum_{u \in \omega^-(A)} b(u) \geq 0 \quad \forall A \in X$$

Examinons maintenant l'algorithme permettant d'obtenir un flot compatible.

On part d'un flot quelconque $\emptyset$. Appelons pour chaque arc $u$ la distance $d(u, \emptyset)$ la quantité suivante :

$$d(u, \emptyset) = 0 \; si \; b(u) \leq \emptyset(u) \leq c(u)$$
$$d(u, \emptyset) = b(u) - \emptyset(u) \; si \; \emptyset(u) \leq b(u)$$
$$d(u, \emptyset) = \emptyset(u) - c(u) si \; \emptyset(u) > c(u)$$

On pose :

$$d(\phi) = \sum_u d(u, \phi) > 0$$

Si $\emptyset$ est tel que $d(\emptyset) = 0$ alors $\emptyset$ est un flot compatible, si $\emptyset$ est tel que $d(\emptyset) > 0$ alors $\emptyset$ n'est pas compatible et il existe au moins un arc $u$ tel que $d(u, \emptyset) > 0$.

Supposons qu'il s'agisse d'un arc $u_0$ tel que $\emptyset(u_0) < b(u_0)$ (la démonstration serait tout à fait analogue si l'on prenait un arc $u_0$ tel que $\emptyset(u_0) > c(u_0)$).

Colorions $u_0$ en noir, et les autres arcs $u$ de la façon suivante :

- si $\emptyset(u) \leq b(u)$ en noir,

- si $b(u) < \emptyset(u) < c(u)$ en rouge,

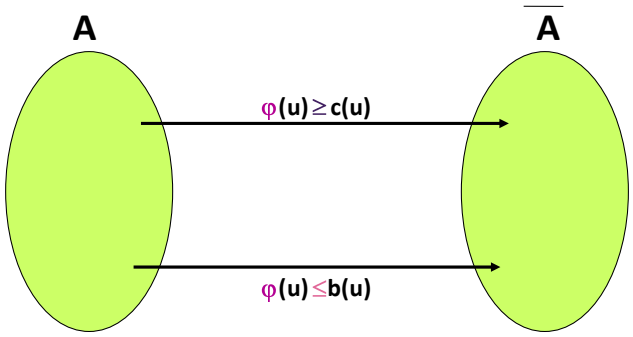
- si $\emptyset(u) \geq c(u)$en vert.

Appliquons alors le lemme de Minty.

Soit on se trouve dans le cas a) du lemme : il existe un cycle élémentaire, avec les arcs noirs parcourus dans le sens de $u_0$ , et les arcs verts dans le sens contraire. On peut alors ajouter aux arcs dans le sens $u_0$ un flot d'au moins une unité, retrancher ce même flot aux arcs parcourus dans le sens contraire. On voit qu'on diminue à chaque fois les

quantités $d(u, \varphi)$ sur ces arcs, sans changer ces distances sur les autres arcs. On diminue donc la distance totale d($\emptyset$).

Supposons maintenant qu'on se trouve dans le cas b) $u_0$ appartient alors à un cocycle où tous les arcs noirs sont dans le même sens, où les arcs verts sont dans le sens contraire et où il n'y a pas d'arc rouge.




Mais si on applique les équations de conservation aux nœuds, on a nécessairement

$$\sum_{u \in \omega^+(A)} \emptyset(u) - \sum_{u \in \omega^-(A)} \emptyset(u) = 0$$

Mais comme $c(u) \leq \emptyset(u)$ pour $u \in \ \omega^+(A)\ et\ b(u) \geq \emptyset(u)\ pour\ u\omega^-(A)$

avec l'inégalité stricte au moins pour $u_0$, on a alors :

$$\sum_{u \in \omega^+(A)} c(u) - \sum_{u \in \omega^-(A)} b(u) < 0$$

C'est à dire que la condition nécessaire d'existence dans un flot compatible n'est pas vérifiée. Si donc on se trouve dans ce cas b), c'est qu'il n'y a pas de flot compatible.

S'il y a des flots compatibles, on se trouve systématiquement dans le cas a) et l'algorithme consiste, grâce à des marquages de Ford-Fulkerson, à diminuer la quantité $d(\emptyset)$ jusqu'à ce que $d(\emptyset) = 0$.

On voit alors que la condition nécessaire d'existence du flot compatible est en même temps suffisante. En effet, si elle est vérifiée pour tout $A \subset X$, on ne peut jamais être dans le cas b) du lemme de Minty, et on se trouve dans le cas a) où l'on peut diminuer $d(\emptyset)$à chaque fois qu'on trouve cette quantité positive. On aboutira donc nécessairement à $d(\emptyset) = 0$ et $\emptyset$ sera alors compatible.

Retenons donc également, au delà de l'algorithme, ci-dessus le théorème suivant, dû à Hoffmann :

La condition nécessaire et suffisante pour qu'un flot compatible existe est que pour tout $A \subset X$, on ait

$$\sum_{u \in \omega^+(A)} c(u) - \sum_{u \in \omega^-(A)} b(u) \geq 0$$

Par ailleurs, sur des graphes munis de ces contraintes de capacité, on peut également résoudre le problème du flot maximal, c'est-à-dire $Max\ \Phi_z$ (le graphe étant muni d'un arc de retour), avec $b(u) \leq \emptyset(u) \leq c(u)$ pour tout $u$.

L'algorithme devient

a) Trouver un flot compatible, grâce à l'algorithme précédent.

b) Marquer les sommets par le marquage suivant :

si $i$ est marqué et si $b(\overrightarrow{ij}) \leq \emptyset(\overrightarrow{ij}) \leq c(\overrightarrow{ij})$, on marque $j$,

si $j$ est marqué et si $\emptyset(\overrightarrow{ij}) > b(\overrightarrow{ij})$ alors on marque $i$,

si $z$ ne peut être marqué, le flot est maximal.

c) Si $z$ peut être marqué, on trouve un cycle sur lequel on peut améliorer le flot. On retourne en b).

## 9.4. PROBLEME DU FLOT DE COUT MINIMAL

Sur un réseau de transport muni des contraintes précédentes (pour un flot Φ, on doit avoir $b(u) \leq \emptyset(u) \leq c(u)$, on définit par ailleurs des coûts unitaires $\gamma_{(u)}$ de transport par arc. Le problème est de trouver le flot Φ tel que :

$$\Phi \geq 0$$

$$b(u) \leq \emptyset(u) \leq c(u)$$

$$et\ Min \sum_{u,\emptyset} \gamma(u)\emptyset(u)$$

On voit qu'il s'agit là d'un problème plus général que les deux précédentes (le flot maximal et le flot compatible) qui n'en sont que des sous-ensembles.

Il existe à présent plusieurs algorithmes permettant de traiter ce problème et utilisant directement la théorie des graphes (ce problème est par ailleurs de toute évidence un programme linéaire que l'on pourrait résoudre en tant que tel, ce qui ne fournirait pas dans la plupart des cas la procédure la plus économique).

Nous allons, sans le détailler, donner l'algorithme le plus proche de ceux que nous venons d'utiliser. Ce n'est pas nécessairement l'algorithme le plus performant. Par la suite, nous détaillerons deux cas particuliers importants de ce problème : le programme de transport et le programme d'affectation.

**Algorithme de KLEIN**

Les étapes sont les suivantes :

a) Trouver un flot compatible sur $G$. On peut utiliser à ce titre l'algorithme précédent. Si l'on ne trouve pas de flot compatible, on arrête : le problème est impossible.

b) Soit Φ un flot compatible. On pratique un marquage classique type Ford-Fulkerson.

1) on marque $x_0$

2) si $i$ est marqué et si $\emptyset(\overrightarrow{i,j}) < c(\overrightarrow{i,j})$, on marque $j$

3) si $j$ est marqué et si $\emptyset(\overrightarrow{i,j}) > b(\overrightarrow{i,j})$, on marque $i$

c) On cherche un cycle élémentaire de sommets marqués, tels que si on appelle $U^+$ les arcs tels que

$\emptyset(\overrightarrow{i,j}) < c(\overrightarrow{i,j})$ et $U^-$ les arcs tels que $\emptyset(\overrightarrow{i,j}) > b(\overrightarrow{i,j})$, on ait

$$\sum_{u \in U^+} \gamma(u) - \sum_{u \in U^-} \gamma(u) < 0$$

On peut alors améliorer le coût total en augmentant d'une certaine quantité les flots sur les arcs $U^+$et en diminuant de la même quantité les flots sur les arcs $U^-$ (de telle façon que le flot soit compatible).

d) On revient en b) jusqu'à ce qu'on ne puisse plus trouver de tels cycles améliorants.

Le principe de cet algorithme est donc simple : il est en effet tout à fait évident que les procédures b) et c) améliorent le flot total. On démontre, de façon moins simple, qu'une condition suffisante pour que le flot soit optimal est qu'il n'existe pas de tels cycles, ce qui finit de justifier l'ensemble de la procédure.

Nous allons voir maintenant un cas particulier du problème du flot à coût minimal, qui est constitué par le programme de transport.

## 9.5. LE PROGRAMME DE TRANSPORT

### 9.5.1. Position du problème

Il s'agit de transporter entre $m$ origines $(i = 1, 2, \ldots m)$ et $n$ destinations certaines quantités (de marchandises par exemple ). En chaque origine $i$ existe une disponibilité $a_i$ et en chaque destination une demande $b_j$. Par ailleurs, entre une origine $i$ et une destination $j$ le coût de transport d'une unité est une constante égale à $c_{ij}$ (qui peut être infinie s'il n'y a pas de liaison entre $i$ et $j$ ). On supposera dans la suite que

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

(la disponibilité totale est égale à la demande totale, contrainte qui ne diminue pas, on le verra, la généralité du problème).

Le problème consiste à transporter toutes les disponibilités (donc à satisfaire toutes les demandes) de façon à minimiser le coût de transport total.

### 9.5.2. Programmation linéaire

Le problème tel qu'il vient d'être posé, se met facilement sous la forme d'un programme linéaire.

Appelons en effet $x_{ij}$ la quantité transportée entre $i$ et $j$. On a les équations :

$$\sum_{i=1}^{m} x_{ij} = b_j \; pour \; j = 1, \dots n$$

$$\sum_{j=1}^{m} x_{ij} = a_i \; pour \; i = 1, \dots m$$

ce qui fait $m + n$ relations linéaires pour les $x_{ij}$, en fait $m + n - 1$ relations indépendantes, du fait de l'égalité

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

On a évidemment par ailleurs $x_{ij} \geq 0, \forall i \, , \forall j$

Enfin, il s'agit de minimiser la fonctionnelle linéaire

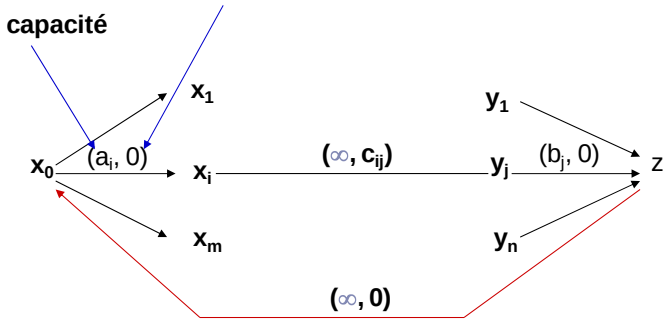
$$C = \sum_{i,j} c_{ij} x_{ij}$$

On est donc en présence d'un programme linéaire à $mn$ variables et $m + n - 1$ contraintes. On pourrait résoudre ce problème par l'algorithme du simplexe ( le fait que les $x_{ij}$ doivent être des entiers n'est pas gênant : on démontre que si les $a_i$, $b_j$, $c_{ij}$ sont entiers, ce qui n'est pas restrictif en recherche opérationnelle, alors la solution du programme linéaire est entière).

Mais l'algorithme du simplexe n'est pas le plus performant pour ce type de problème. On préfère utiliser des procédures fondées sur les concepts de flots.

### 9.5.3. Utilisation des flots

Dessinons pour le problème ci-dessus le réseau de transport :




- d'un somme $x_0$t entrée partent des arcs de capacités $a_i$ et de coût nul vers autant de sommets $x_i (i = 1 \dots m)$ qu'il y a d'origines dans le programme de transport.
- des arcs relient chaque sommet origine aux sommets destination $y_j (j = i, \dots n)$. Ces arcs sont de capacité infinie et de coût $c_{ij}$.
- de chaque sommet-destination $y_j$ part un arc vers un sommet de sortie $z$ de capacité $b_j$ et de coût 0.

On a enfin ajouté l'arc-retour de capacité infinie et de coût nul.

On voit alors que résoudre le problème de transport revient à trouver un flot saturant les arcs issus de $x_0$ et les arcs arrivant en $z$ (un tel flot sera appelé saturant) et de coût minimal.

Nous appellerons $\emptyset_{ij}$ le flot entre $x_i$ et $y_j$, et

$$c(\emptyset) = \sum_{i,j} c_{ij} \emptyset_{ij}$$

le coût associé au flot $\Phi$.

On pourrait appliquer à ce problème particulier l'algorithme de KLEIN, avec les contraintes :

$$a_i \leq \emptyset(\overrightarrow{x_0, x_i}) \leq a_i$$

$$b_j \leq \emptyset(\overrightarrow{y_j, z}) \leq b_j$$

On aboutirait à l'algorithme le plus ancien permettant de résoudre le programme de transport (algorithme dit du Stepping-Stone) mais on préférera ici une autre procédure, fondée sur la notion de potentiels.

### 9.5.4. Flots et potentiels

Introduisons deux vecteurs $\alpha = (\alpha_i)(i = 1 \dots m)\ \beta(\beta_j)(j = 1 \dots n)\ \ avec$

$$\alpha_i + \beta_j \leq c_{ij}\ \forall i, \forall j$$

Appelons $\gamma$ l'ensemble $\alpha\ et\ \beta$ ; $\gamma$ sera appelé fonction potentielle. Soit la quantité

$$b(\gamma) = \sum_{i=1}^{m} a_i \alpha_i + \sum_{j=1}^{n} b_j \beta_j$$

On a le lemme suivant :

**Lemme 1**

On a toujours : $c(\emptyset) \geq b(\gamma)$ pour tout flot saturant et toute fonction potentielle.

En effet

$$\sum_{ij} c_{ij}\phi_{ij} \geq \sum_{ij} (\alpha_i + \beta_j)\phi_{ij}$$

(puisque $\alpha_i + \beta_j \leq c_{ij}\ \forall i, \forall j$)

$$\sum_{ij} c_{ij}\phi_{ij} \geq \sum_{i} \alpha_i \sum_{j} \phi_{ij} + \sum_{j} \beta_j \sum_{i} \phi_{ij}$$

mais

$$\sum_{j} \phi_{ij} = a_i \quad \text{et} \quad \sum_{i} \phi_{ij} = b_j$$

puisque $\emptyset$ est saturant, soit

$$\sum_{ij} c_{ij}\phi_{ij} \geq \sum_{i} a_i \alpha_i + \sum_{j} b_j\ \beta_j$$

d'où l'inégalité

$c(\emptyset) \geq b(\gamma)$

Si donc on trouve un flot saturant $\emptyset$ et une fonction potentielle $\gamma$, tels que $c\Phi) = b(\gamma)$, on a trouvé le flot $\Phi$ de coût minimal.

Démontrons alors un autre résultat.

**Lemme 2**

Soit une fonction potentielle $\gamma$.

Si on supprime les arcs $\overrightarrow{x_i, y_j}$tels que $\alpha_i + \beta_j < c_{ij}$ (on garde donc ceux tels que $\alpha_i + \beta_j = c_{ij}$), on obtient un graphe partiel $\overline{G}_\gamma$.

Si $G_\gamma$ admet un flot $\Phi$ saturant ce flot est de coût minimum. En effet :

a) $\Phi$ est un flot de $G$

b) On a

$$c(\emptyset) = \sum_{i,j/u_{ij} \in \bar{G}_\gamma} c_{ij}\emptyset_{ij}$$

$(u_{ij}\ est\ l'arc\ x_i, y_j)$

Par ailleurs, sur $G_\gamma$, on a $\alpha_i + \beta_j = c_{ij}$ et

$$c(\emptyset) = \sum_{i,j/u_{ij} \in \bar{G}_\gamma} (\alpha_i + \beta_j)\emptyset_{ij}$$

Tous les sommets $x_i$ et les sommets $y_j$ sont utilisés puisque $\emptyset$ est saturant.

$$c(\emptyset) = \sum_{i=1}^{m} \alpha_i \sum_{j/u_{ij} \in \bar{G}_\gamma} \emptyset_{ij} + \sum_{j=1}^{n} \beta_j \sum_{i/u_{ij} \in \bar{G}_\gamma} \emptyset_{ij} = \sum_{i=1}^{m} a_i\alpha_i + \sum_{j=1}^{n} b_j\beta_j$$

Dans ce cas, on a alors :

$c(\Phi) = b(\gamma)$, et donc d'après le lemme 1, $\Phi$ est de coût minimal. L'algorithme va alors consister à partir d'une fonction potentielle quelconque $\gamma$ à définir $\overline{G}_\gamma$, à déterminer le flot maximal sur $\overline{G}_\gamma$. Si ce flot est saturant, l'algorithme est terminé : on a trouvé le flot de coût minimal. Si ce n'est pas le cas, il faut changer $\gamma$ de telle façon qu'au bout d'un certain nombre d'itérations, le flot sur $\overline{G}_\gamma$ devienne saturant.

Remarquons que la première opération, consistant à se donner une fonction potentielle n'a rien de difficile. Par exemple, si tous les $c_{ij}$ sont positifs, ce qui est en général le cas $\alpha_i = 0\ \forall i\ \ et\ \beta_j = 0\ \forall j$ donnent une fonction potentielle possible. Ce n'est certainement pas la meilleure, et on a intérêt à bien choisir le $\gamma$ de départ (cf. ci-dessous l'application numérique).

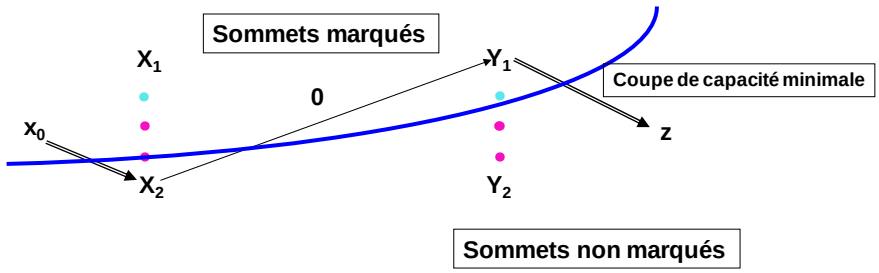
Plaçons nous donc dans le cas où on a une fonction potentielle $\gamma$, un graphe $G_\gamma$, sur lequel on a déterminé le flot maximal $\Phi$ ; mais ce flot n'est pas saturant. Comment changer $\gamma$ ?
Si le flot $\Phi$ est maximal pour $G_\gamma$, c'est qu'à la fin de l'algorithme de Ford-Fulkerson, un certain nombre de sommets sont marqués, dont $x_0$, les autres étant non marqués, dont $z$. Appelons :

$X_1$: ensemble des sommets origines marqués,

$X_2$ : ensemble des sommets origines non marqués,

$Y_1$ : ensemble des sommets destinations marqués,

$Y_2$: ensemble des sommets destinations non marqués.




On a symbolisé ci-dessus la coupe de capacité minimale. On a alors les propriétés suivantes qui découlent de l'algorithme de Ford-Fulkerson :

Les arcs $x_0X_2$ sont saturés,

Les arcs $Y_1z$ sont saturés,

Les arcs $X_2Y_1$ sont de flot nul.

Quant aux arcs $X_1Y_2$, ils ne peuvent pas exister, puisqu'ils sont de capacité infinie et qu'ils appartiennent à $\omega^+(A)$, $A$ étant la coupe ( ou encore si un arc $X_1Y_2$ existait, comme il est de capacité infinie et que le sommet de $X_1$ est marqué, alors le sommet de $Y_2$ serait marqué.)

Opérons alors sur la fonction potentiel $\gamma$ la transformation suivante :

$$\alpha_i' = \alpha_i + 1 \; pour \; x_i \in X_1$$
$$\alpha_i' = \alpha_i \; pour \; x_i \in X_2$$
$$\beta_j' = \beta_j - 1 \; pour \; y_j \in Y_1$$
$$\beta_j' = \beta_j \; pour \; y_j \in Y_2$$

On obtient une nouvelle fonction potentiel $\gamma'$. En effet, on a évidemment $(\alpha_i + \beta_j \leq c_{ij})$.

$$x_i \in X_1,\ y_j \in Y_1 \qquad \alpha'_i + \beta'_j = \alpha_i + \beta_j \leq c_{ij}$$
$$x_i \in X_2,\ y_j \in Y_2 \qquad \alpha'_i + \beta'_j = \alpha_i + \beta_j \leq c_{ij}$$
$$x_i \in X_2,\ y_j \in Y_1 \qquad \alpha'_i + \beta'_j = \alpha_i + \beta_j - 1 \leq c_{ij}$$
$$x_i \in X_1,\ y_j \in Y_2 \qquad \alpha'_i + \beta'_j = \alpha_i + \beta_j + 1 \leq c_{ij}$$

(puisque, pour ce dernier cas, la non-existence des arcs $X_1Y_2$ signifie $\alpha_i + \beta_j < c_{ij}$.

Calculons alors la fonction $b(\gamma')$.

$$b(\gamma') = \sum_i \alpha'_i\, a_i + \sum_j \beta'_j\, b_j = \sum_{X_1} (\alpha_i + 1) a_i + \sum_{X_2} \alpha_i\, a_i + \sum_{Y_1} (\beta_j - 1) b_j + \sum_{Y_2} \beta_j\, b_j$$

Soit

$$b(\gamma') = b(\gamma) + \sum_{X_1} a_i - \sum_{Y_1} b_j$$

Soit encore

$$b(\gamma') - b(\gamma) = \sum_{X} a_i - \left( \sum_{X_2} a_i + \sum_{Y_1} b_j \right)$$

Mais on a également

$$\emptyset_Z = \sum_{X_2} a_i + \sum_{Y_1} b_j$$

(flot sortant de la coupe)

Donc :

$b(\gamma') - b(\gamma) > 0$ car $\Phi$ est non saturant.

Cette opération a donc entraîné une augmentation de $b(\gamma)$.

À chaque fois donc que l'on trouve un flot $\Phi$ non saturant, mais maximal sur $\overline{G_\gamma}$, on peut changer $\gamma$de telle façon que $b(\gamma)$ augmente. Or $b(\gamma)$ est borné supérieurement par l'ensemble des valeurs $c(\Phi)$, $\Phi$ étant saturé. Au bout d'un certain temps donc, on obtiendra un flot saturant sur $\overline{G_\gamma}$, et donc le flot de coût minimal.

Par ailleurs, lorsqu'on a un flot $\Phi$ non saturant sur $\overline{G_\gamma}$ , on continue l'algorithme de la façon suivante.

On augmente les $\alpha_i$ pour les $x_i \in X_1$ et on diminue les $\beta_j$ pour les $y_j \in Y_1$ d'une même quantité, de telle façon qu'une quantité $\alpha_i + \beta_j$ devienne égale à $c_{ij}$. Cette opération consiste donc à ajouter un arc $x_i y_j$ :

- Soit le sommet $y_j$ n'était pas saturé, et on va, en partant du flot précédent, pouvoir saturer ce sommet et donc augmenter le flot.
- Soit le sommet $y_j$ était saturé, alors on n'augmente pas nécessairement le flot, mais on augmente par contre le nombre de sommets marqués $Y_1$, puisque qu'on marque $y_j$ (les autres sommets marqués de $Y_1$restent marqués, puisqu'ils n'étaient pas marqués par les arcs $X_2 Y_1$, les seuls qui risquent d'être supprimés par la modification de $\gamma$.)

$Y_1$ ne peut alors augmenter indéfiniment sans que le flot Ø n'augmente à son tour, puisque Ø n'étant pas saturant, certains arcs $\overrightarrow{y_j z}$ sont non saturés : au bout d'un certain temps, on pourra marquer $z$ et augmenter le flot.

**Récapitulons l'algorithme :**

a) On se donne une fonction potentiel $\gamma$. (ensemble de $\alpha_i$ et $\beta_j$ tels que $\alpha_i + \beta_j \leq c_{ij}$ ).

b) On détermine le graphe partiel $\overline{G_\gamma}$ (obtenu en supprimant les arcs $x_i y_j$ tels que $\alpha_i + \beta_j < c_{ij}$ ).

c) On calcule le flot maximal sur $\overline{G_\gamma}$.

Pour cela on opère de la manière suivante :
i) On détermine les ensembles $X_1$, $X_2$, $Y_1$, $Y_2$

$X_1$ : ensemble des sommets origines marqués
$X_2$ : ensemble des sommets origines non marqués
$Y_1$ : ensemble des sommets destinations marqués
$Y_2$ : ensemble des sommets destinations non marqués

ii) On calcule la quantité $\lambda = \text{Min}\,[c_{ij} - (\alpha_i + \beta_j)] = \lambda_{ij}$

**i/ $x_i \in X_1$, j/ $y_j \in Y_2$**

iii) On effectue sur $\gamma$ les modifications suivantes :

$\alpha'_i = \alpha_i + \lambda$ pour $x_i \in X_1$
$\alpha'_i = \alpha_i$ pour $x_i \in X_2$
$\beta'_i = \beta_i - \lambda$ pour $y_j \in Y_1$
$\beta'_j = \beta_j$ pour $y_j \in Y_2$

On revient à l'étape b)

Si ce flot sature les sommets origines et les sommets destination, ce flot est le flot de coût minimal. L'algorithme s'arrête.

d) Si le flot maximal sur $\overline{G_\gamma}$ n'est pas saturant on change $\gamma$, en augmentant d'une même quantité les $\alpha_i$ des sommets origines marqués et diminuant les $\beta_j$ des sommets destinations marqués de façon à créer un nouvel arc sur $\overline{G_\gamma}$. On retourne en c).

Appliquons cet algorithme sur l'exemple suivant, représenté par la matrice des coûts $c_{ij}$, les disponibilités $a_i$ et les demandes $b_j$.

Demandes

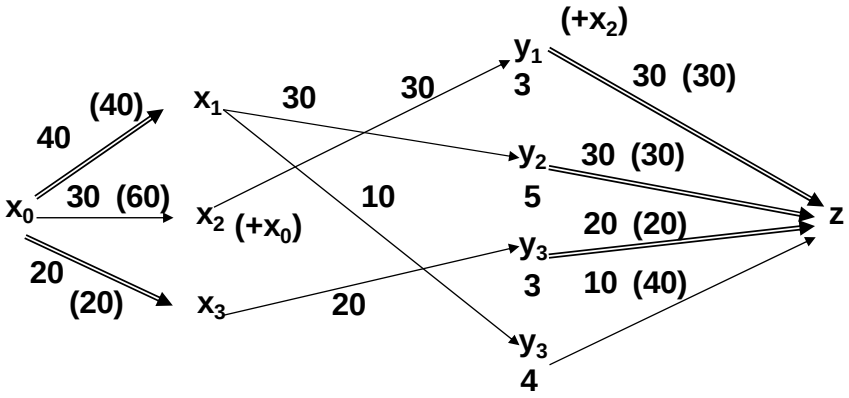
| | bj | 30 | 30 | 20 | 40 |
|---|---|---|---|---|---|
| Disponibilités ai | 40 | 10 | 5 | 8 | 4 |
| | 60 | 3 | 7 | 9 | 8 |
| | 20 | 4 | 11 | 3 | 9 |

Coûts

a) Détermination de la fonction $\gamma$ de départ

Pour partir d'une fonction potentielle convenable, on peut partir du coût unitaire minimal, soit $c_{21} = 3$.
On peut poser $\alpha_2 = 0$, $\beta_1 = 3$ ;
Ensuite on sélectionne l'autre coût égal à 3 soit $c_{33}$ ; posons $\alpha_3 = 0$, $\beta_3 = 3$ ;
Puis le coût immédiatement supérieur $c_{14} = 4$. on pose $\alpha_1 = 0$, $\beta_4 = 4$.
Enfin $c_{12} = 5$ donne $\beta_2 = 5$. On a ainsi une fonction potentielle possible.

On a alors le graphe $\overline{G_\gamma}$ suivant dans lequel on détermine facilement le flot maximal.




(Les capacités non infinies sont indiquées entre parenthèses. Les flots sont indiqués hors des parenthèses).

Les seuls sommets marqués sont $x_2 et\ y_1$.

On augmente donc $\alpha_2$ et on diminue $\beta_1$d'une même quantité. On doit toujours avoir :

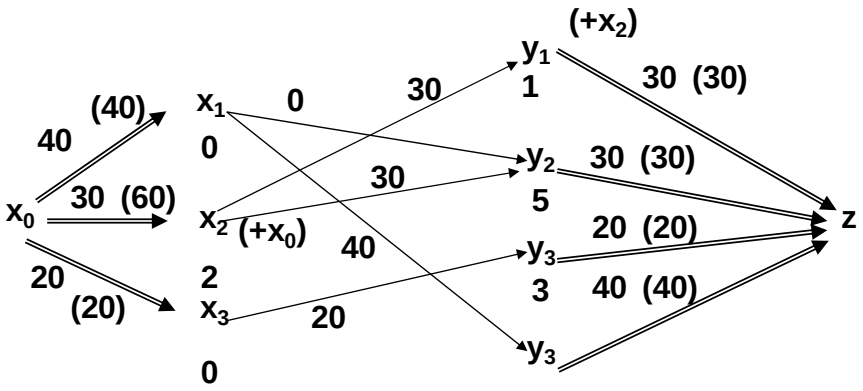
$$\alpha_i + \beta_j \leq c_{2j}\ )\ avec\ y_j \in Y_2$$

c'est à dire $j = 2, 3\ ou\ 4$. On voit que l'on est limité par l'inégalité :

$$\alpha_2 + \beta_2 \leq c_{22}\ ou\ \alpha_2 + 5 \leq 7$$

ce qui donne $\alpha_2 = 2\ et\ donc\ \beta_1 = 1$.

Le nouveau graphe $\overline{G_\gamma}$ relatif à la nouvelle fonction potentielle est le suivant :




On obtient un flot saturant ; on est donc à l'optimum (obtenu donc très rapidement par cet algorithme) avec les flux transportés suivants :

$\emptyset_{14} = 40 \quad \emptyset_{21} = 30 \quad \emptyset_{22} = 30 \ \emptyset_{33} = 20$, tous les autres flux étant nuls.

**Remarque** : lorsque l'on n'a pas l'égalité $\sum a_i = \sum b_j$, il est très facile de se ramener au cas précédent. Il suffit de créer une origine (ou une destination) fictive de disponibilité (ou demande) $\sum a_i - \sum b_j (\sum b_j - \sum a_i)$ et en ajoutant des coûts quelconques, mais égaux à une même quantité entre toutes les destinations (ou origines) avec cette origine (ou destination) fictive.

### 9.5.5 Algorithme de Stepping-Stone

Parmi les nombreuses méthodes existantes, la méthode de la « différence maximale » ou méthode de Balas-Hammer permet d'obtenir une solution de base initiale. Elle repose sur le critère économique du coût marginal et introduit la notion de manque à gagner. Elle donne une solution de base souvent très proche de l'optimum et est avérée d'une efficacité bien supérieure aux méthodes du « coin Nord-Ouest » ou de Houthakker.

**Exemple :**

*Recherche d'une solution de base.* Dans chaque ligne et dans chaque colonne, on recherche le coût minimal et on fait la différence entre ce coût et le coût immédiatement supérieur dans la même ligne ou dans la même colonne.

Par exemple, dans la colonne (1) du tableau ci-dessous, le coût minimal est 3 et le coût immédiatement supérieur est 7, la différence est 4.

| | | | | |
|---|---|---|---|---|
| 7 | 8 | 5 | 4 | 2 |
| 3 | 7 | 2 | 3 | 1 |
| 9 | 5 | 10 | 3 | 4 |
| 2 | 2 | 6 | 10 | |
| 4 | 2 | 3 | | |

On prend ensuite le maximum des différences, ici 4, et l'on attribue à la case contenant le coût minimum de la ligne ou de la colonne correspondante (ici la colonne (1)) le minimum entre disponibilité et demande. Dans le cas présent, on attribuera $min(2,3) = 2$ à la case $(2,1)$.

On raye alors la colonne ou la ligne saturée et l'on recommence les opérations sur le tableau restant.

La solution ainsi obtenue est :

| | | |
|---|---|---|
| | | 4 |
| 2 | | 1 |
| | 2 | 1 |

Cette solution est bien une solution de base puisqu'à chaque pas une ligne ou une colonne est saturée, sauf au dernier pas où l'on sature une ligne et une colonne.

*Optimisation de la solution de base. Algorithme du Stepping-Stone.*

Nous allons maintenant, à partir de la solution de base obtenue, chercher une nouvelle solution qui corresponde à un coût global moins élevé. Pour obtenir une solution plus intéressante, supposons que l'on affecte une unité dans la case (1,1). Il faut alors en retirer une de la case (1,3), en ajouter une dans la case (2,3) et en retirer une dans la case (2,1).

| | | |
|---|---|---|
| (+) | | (-) |
| (-) | | (+) |
| | | |

Cet échange circulaire d'une unité fait varier le coût total d'une quantité $\delta_{11} = 7 - 5 + 2 - 3 = 1$

Les quantités $\delta_{ij}$ représentant les écarts unitaires quand on passe d'une base à une autre, elles constituent les coûts marginaux unitaires.

De la même manière évaluons de tels échanges pour les autres cases :

$\delta_{12} = 8 - 5 + 10 - 5 = 8$

$\delta_{22} = 7 - 2 + 10 - 5 = 10$

$\delta_{31} = 9 - 3 + 2 - 10 = -2$

Pour diminuer le coût total, il faut que l'échange corresponde à un $\delta_{ij}$ négatif. On prend le plus négatif ; ici, il n'y a qu'un seul $\delta_{ij}$ négatif : $\delta_{31} = -2$

Pour passer d'une solution de base à une autre, il faut « vider » l'une des cases et en remplir une vide de la quantité correspondante.

| | | |
|---|---|---|
| (-) | | (+) |
| (+) | | (-) |

Ici, nous ne pouvons enlever que la plus petite quantité dans ces cases et enlèverons donc 1 aux cases $(2,1)\ et\ (3,3)$ pour l'ajouter aux cases $(3,1)\ et\ (2,3)$.

Le coût total de transport a diminué de $1 \times 2 = 2$

La nouvelle solution de base est optimale. En effet, tous les $\delta_{ij}$ sont maintenant positifs :

$\delta_{11} = 1\ ;\ \delta_{12} = 6\ ;\ \delta_{22} = 8\ ;\ \delta_{33} = 2$

**Remarque :** On constate que les méthodes proposées de tiennent pas compte des contraintes de la forme $\xi_{ij} \leq c_{ij}$. On suppose donc les arcs de capacité infinie ou, sous une autre forme, qu'il n'ya pas de limitations quant aux quantités transportées d'un point à un autre.

Nous allons clore maintenant ce chapitre par un cas particulier du programme de transport, qui est le problème de l'affectation optimale.

## 9.6. LES PROBLEMES D'AFFECTATION

Supposons que nous ayons à résoudre le problème suivant : nous disposons de $n$ tâches auxquelles on peut affecter $n$ moyens, chaque moyen étant affecté à une tâche et à une seule.

Le coût de l'affectation du moyen $i$ à la tâche $j$ est évalué à une constante $c_{ij}$.

Nous appellerons $C$ la matrice constituée par les $c_{ij}$ ($i$ indice de la ligne, $j$ indice de la colonne).

Le problème est : *comment affecter les moyens pour que le coût total soit minimal ?*

### 9.6.1. Possibilités de résolutions

Constatons tout d'abord que si l'on voulait énumérer tous les cas possibles, il y en aurait $n!$, ce qui exclue que l' on utilise cette méthode pour des nombres $n$ importants.

Par ailleurs, si nous voulons formaliser le problème, nous pouvons le faire de la façon suivante :

Posons

$x_{ij} = 1$ si le moyen $i$ est affecté à la tâche $j$.

$x_{ij} = 0$ si le moyen $i$ n'est pas affecté à la tâche $j$.

Comme un moyen n'est affecté qu'à une tâche et une seule, on doit avoir :

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

Par ailleurs, il s'agit de minimiser la fonction $C$.

$$C = \sum_{i,j} c_{ij}\, x_{ij}$$

Là encore, nous obtenons un programme linéaire : sa particularité est que les variables $x_{ij}$ sont astreintes à ne prendre que les valeurs $0\ ou\ 1$. Ces programmes particuliers, on l'a vu, sont appelés programmes en variables bivalentes.

On pourrait alors par exemple utiliser une procédure de recherche arborescente (cf. chapitre précédent) en prenant pour sous-ensemble de la procédure les sous-ensembles d'affectations possibles telles qu'une variable particulière soit égale à $0\ ou\ 1$. L'expérience montre que l'exploration risque dans beaucoup de cas d'être assez longue.

On reconnaîtra également sur la formalisation ci-dessus un programme de transport. On pourrait donc appliquer l'algorithme précédent. En fait c'est une légère variante de cet algorithme que nous allons exposer, l'algorithme hongrois utilisant lui aussi la procédure de Ford-Fulkerson.

### 9.6.2. Théorème fondamental

On ne change pas la solution du problème en retranchant ou en ajoutant à une ligne ou à une colonne de la matrice des coûts une même quantité.

En effet, prenons une ligne k quelconque.

La fonction $C$ peut s'écrire :

$$C = \sum_{i \neq k, j} c_{ij}\ x_{ij} + \sum_{j} c_{kj}\ x_{kj}$$

Ajoutons aux $C_{kj}$ une quantité algébrique $d$ quelconque, la fonction de coût devient $C'$:

$$C' = \sum_{i \neq k, j} c_{ij}\ x_{ij} + \sum_{j} c_{kj}\ x_{kj} + d \sum_{j} x_{kj}$$

Mais comme :

$$\sum_{j} x_{kj} = 1$$

On en déduit : $C' = C + d$

En conséquence, minimiser $C$ revient à minimiser $C'$

Le même raisonnement s'applique sur les colonnes de la matrice $C$.

### 9.6.3. Algorithme hongrois

Cet algorithme consiste essentiellement à utiliser le résultat précédent pour transformer progressivement la matrice $C$ afin d'y faire apparaître l'affectation optimale.

Nous allons l'expliquer sur un exemple, pour simplifier l'exposé.

Soit alors le problème d'affectation suivant : $n = 6$ ; les moyens seront notés $ABCDEF$ et les tâches $abcdef$. La matrice $C$ sera :

**Matrice 1**

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 25 | 19 | 24 | 15 | 19 | 30 |
| B | 40 | 27 | 28 | 20 | 31 | 29 |
| C | 17 | 14 | 11 | 12 | 18 | 16 |
| D | 32 | 28 | 27 | 20 | 31 | 30 |
| E | 29 | 25 | 28 | 18 | 24 | 27 |
| F | 30 | 32 | 35 | 22 | 30 | 28 |

*Phase 1- Apparition d'un zéro par ligne et par colonne*

Soustrayons à chaque ligne le plus petit élément de cette ligne ; on en a le droit d'après le théorème ci-dessus ; on obtient la nouvelle matrice.

**Matrice 2**

| | a | b | c | d | e | f | |
|---|---|---|---|---|---|---|---|
| A | 25 | 19 | 24 | 15 | 19 | 30 | -15 |
| B | 40 | 27 | 28 | 20 | 31 | 29 | -20 |
| C | 17 | 14 | 11 | 12 | 18 | 16 | -11 |
| D | 32 | 28 | 27 | 20 | 31 | 30 | -20 |
| E | 29 | 25 | 28 | 18 | 24 | 27 | -18 |
| F | 30 | 32 | 35 | 22 | 30 | 28 | -22 |

**Matrice 3**

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 10 | 4 | 9 | 0 | 4 | 15 |
| B | 20 | 7 | 8 | 0 | 11 | 9 |
| C | 6 | 3 | 0 | 1 | 7 | 5 |
| D | 12 | 8 | 7 | 0 | 11 | 10 |
| E | 11 | 7 | 10 | 0 | 6 | 9 |
| F | 8 | 10 | 13 | 0 | 8 | 6 |

Si sur cette matrice, nous recommençons la même opération pour les colonnes; nous obtenons une nouvelle matrice où apparaît au moins un zéro dans chaque ligne et dans chaque colonne.

**Matrice 4**

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 4 | 1 | 9 | 0 | 0 | 10 |
| B | 14 | 4 | 8 | 0 | 7 | 4 |
| C | 0 | 0 | 0 | 1 | 3 | 0 |
| D | 6 | 5 | 7 | 0 | 7 | 5 |
| E | 5 | 4 | 10 | 0 | 2 | 4 |
| F | 2 | 7 | 13 | 0 | 4 | 1 |

Si l'on pouvait alors affecter sur cette matrice un zéro par ligne et par colonne, le problème serait résolu; en effet la fonction économique $C$, avec ses nouveaux coûts tous positifs ou nuls, serait alors nulle, donc la plus faible possible.

Une telle affectation existe-t-elle ? Pour en décider nous allons appeler *couplage* un ensemble de zéros de la matrice des coûts tel que deux zéros quelconques du couplage correspondent à deux lignes différentes et deux colonnes différentes, la taille du couplage sera alors le nombre de ses zéros.

Ainsi, sur la matrice 3, les zéros $(A, e), (B, d), (C, c)$ forment un couplage de taille 3; par contre les zéros $(A, d), (B, d), (C, a)$, ne forment pas de couplage (colonne $d$ commune aux deux premiers).

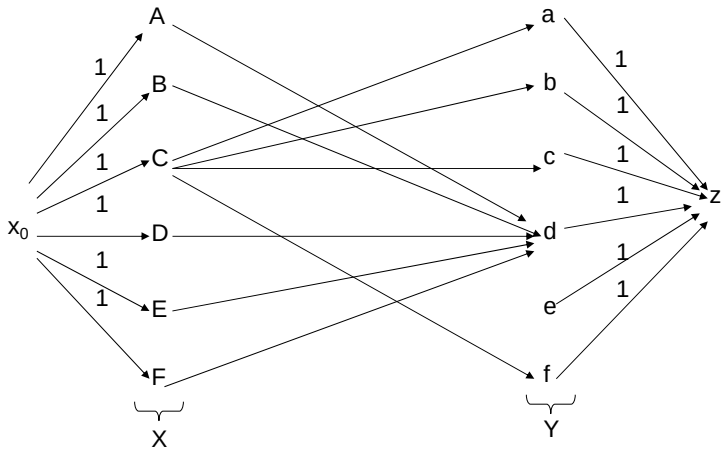
La taille d'un couplage est évidemment limitée par $n$ dans le cas général et sur l'exemple présent par 6.

Dans ces conditions, si nous recherchons le couplage de taille maximale sur la matrice 3, et si nous trouvons que ce couplage a pour taille 6, le problème est résolu : on a trouvé l'optimum; si la taille du couplage obtenu est inférieure à 6, il faudra transformer de nouveau la matrice, de façon à faire apparaître un couplage de taille 6.

*Phase 2 - Recherche du couplage maximal*

Pour chercher ce couplage maximal (*i.e* de taille maximale), nous allons nous aider de l'algorithme de Ford-Fulkerson.

Pour cela, nous allons faire correspondre à la matrice 3 le graphe $G$ ci-dessous, qui est un réseau de transport :




Ce graphe a :

- une entrée $x_0$ reliée par un arc de capacité 1 à chacun des sommets $ABCDEF$ (les moyens, qui constituent l'ensemble $X$).

- une sortie $z$, reliée par un arc de capacité 1 à chacun des sommets $abcdef$ (les tâches, qui constituent l'ensemble $Y$).
- des arcs reliant un sommet de $X$ à un sommet de $Y$ si et seulement si le coût de la matrice 3 est nul pour la ligne et la colonne correspondantes. Ces arcs ont une capacité infinie.

Nous allons poser l'équivalence suivante : sélectionner un zéro dans la matrice 3 pour construire un couplage consistera à porter un flot d'une unité sur l'arc correspondant allant de $X$ à $Y$.

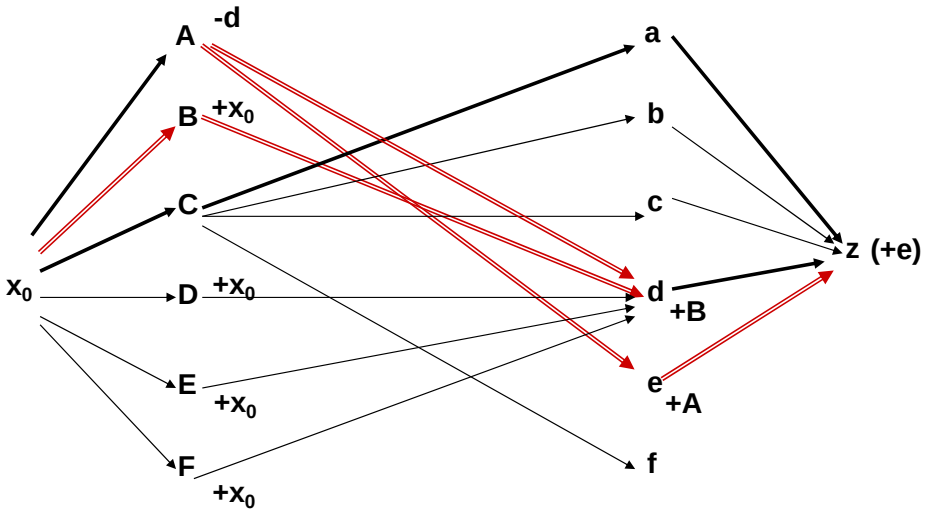
On voit alors que chercher le couplage optimal revient à chercher le flot maximal sur le réseau de transport $G$.

En effet, le fait que les arcs joignant $x_0$ à $X$ et $Y$ à $z$ soient de capacité 1 nous assurent que deux arcs adjacents parmi ceux joignant $X$ à $Y$ ne peuvent avoir tous les deux un flot d'une unité : si l'on sélectionne les zéros correspondant aux arcs de flot non nul, il s'agira donc bien d'un couplage; par ailleurs le flot total entrant en $x_0$ ou sortant en $z$ représentera bien le nombre total de zéros sélectionnés : maximiser ce flot revient bien à chercher le couplage de taille maximale.

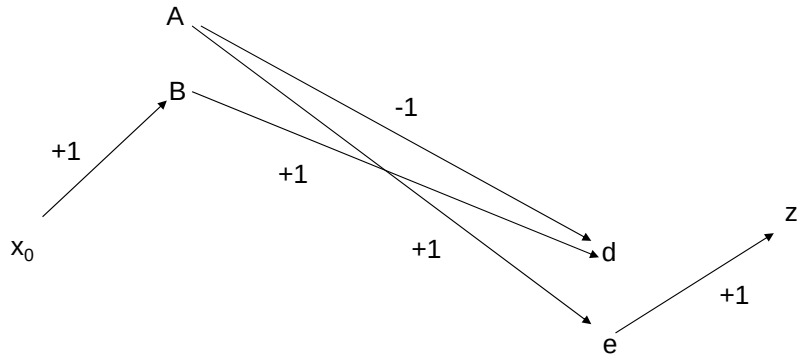
Pour maximiser le flot, utilisons l'algorithme de Ford-Fulkerson.

Tout d'abord, pour faire passer un flot au jugé convenable dans le graphe $G$, il suffit d'ordonner les deux ensembles de sommets et d'essayer de saturer dans l'ordre les sommets de $X$ (c'est-à-dire faire passer un flot d'une unité entre 0 et le sommet considéré) ; à chaque fois, on relie un sommet de $X$ au premier sommet possible non saturé de $Y$ (flot nul entre ce sommet et $z$).

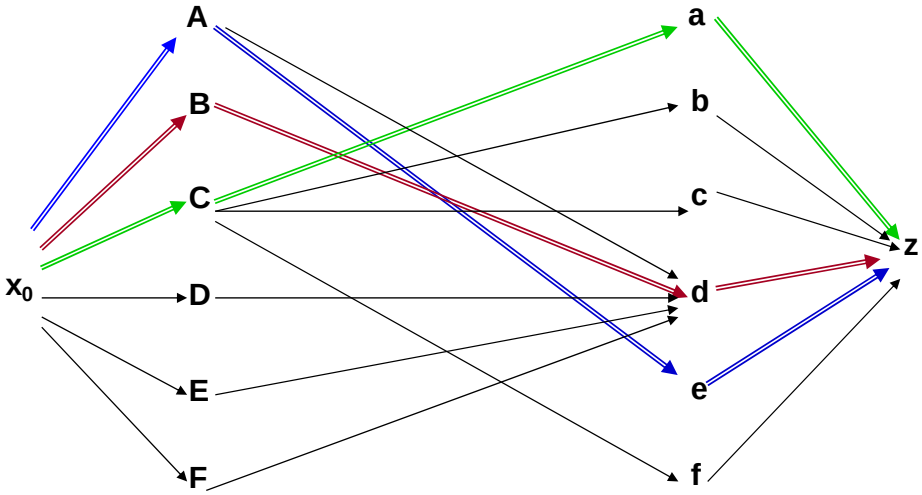
On obtient pour l'exemple pris : (arcs de flot égal à 1 en traits épais)



Ce flot est-il maximal ? Pour répondre, effectuons le marquage de Ford-Fulkerson. Ce marquage aboutit à $z$ par la chaîne $x_0BdAez$. On peut donc améliorer le flot, en modifiant les flots sur cette chaîne.




Le flot obtenu peut alors se représenter par la figure suivante :




Un nouveau marquage ne permet pas de marquer $z$. On a donc obtenu le flot maximal et le couplage optimal (ou plutôt *un* couplage optimal). Ce couplage, constitué de $(A, e), (B, d), (C, a)$ est de taille égale à 3 : nous n'avons donc pas affecté tous les moyens aux tâches et il sera nécessaire d'améliorer le couplage.

Auparavant, convenons de repérer le couplage obtenu dans la matrice des coûts (la matrice 3) de la manière suivante :

- les zéros appartenant au couplage (correspondant aux arcs $X \to Y$ saturés) sont encadrés,

- les zéros n'appartenant pas au couplage sont barrés.

Pour la matrice 3, on obtient :

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 4 | 1 | 9 | ~~0~~ | 0 | 10 |
| B | 14 | 4 | 8 | 0 | 7 | 4 |
| C | 0 | ~~0~~ | ~~0~~ | 1 | 3 | ~~0~~ |
| D | 6 | 5 | 7 | ~~0~~ | 7 | 5 |
| E | 5 | 4 | 10 | ~~0~~ | 2 | 4 |
| F | 2 | 7 | 13 | ~~0~~ | 4 | 1 |

*Phase 3 - Amélioration du couplage*

Analysons le graphe ci-dessus. Nous pouvons, comme nous l'avons fait pour le programme de transport, décomposer $X$, ensemble des moyens, et $Y$ ensemble des tâches en deux sous-ensembles.

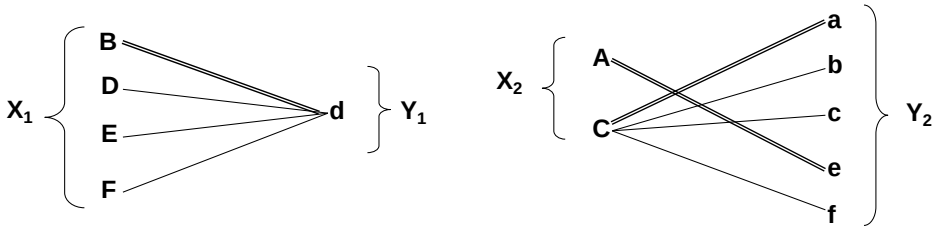
$X_1$ : ensemble des sommets marqués de $X$

$X_2$: ensemble des sommets non marqués de $X$

$Y_1$: ensemble des sommets marqués de $Y$

$Y_2$: ensemble des sommets non marqués de $Y$

Cette décomposition donne ici :




À propos de cette décomposition, deux résultats très simples vont nous servir pour la suite du raisonnement :

1) il n'existe pas d'arc de flux = 1 entre $X_2$ *et* $Y_1$.
2) il n'existe pas d'arc entre $X_1$ *et* $Y_2$.

En effet, pour 1), s'il en était ainsi, le sommet de $X_2$, extrémité initiale de l'arc saturé entre $X_2$ et $Y_1$ serait marqué. Tous les arcs $X_2Y_1$ sont donc de flot nul.

Pour 2), c'est également évident, car les arcs $XY$ étant de capacité infinie, un arc entre $X_1$ et $Y_2$, le sommet initial dans $X_1$ étant marqué, entraînerait le marquage du sommet de $Y_2$.

On peut alors réécrire la matrice des coûts (ici la matrice 3) en réordonnant les lignes et les colonnes de la façon suivante :

| | Y1 | Y2 |
|---|---|---|
| X1 | | Pas de 0 ni de ∅ |
| X2 | pas de 0 | |

D'après ce qu'on vient de voir, les éléments $[X_1, Y_2]$ sont tous *positifs* (pas d'arcs $X_1 \rightarrow Y_2$ dans le graphe équivalent). Aucun des éléments $[X_2, Y_1]$ n'est un zéro encadré.

Prenons alors le plus petit élément du tableau $X_1Y_2$ ; soit $\alpha$ cet élément. On a $\alpha > 0$. Retranchons $\alpha$ des lignes $X_1$; on fait apparaître au moins un zéro dans le tableau $[X_1, Y_2]$. Comme il peut y avoir des zéros dans le tableau $[X_1, Y_1]$, nous allons ajouter $\alpha$ aux colonnes $Y_1$. Cette double opération se traduit par le bilan final :

| | Y1 | Y2 |
|---|---|---|
| X1 | -α + α = 0 | - α |
| X2 | + α | 0 |

Comme les zéros encadrés du couplage maximal se trouvent dans les tableaux $[X_1, Y_1]$ et $[X_2, Y_2]$ on voit que l'*on n'a transformé aucun de ces zéros*. Par contre, on a fait apparaître un zéro dans le tableau $[X_1, Y_2]$. Ce faisant on est sûr d'obtenir un nouveau couplage de taille au moins égale à celle du couplage précédent.

Plus précisément, une fois la matrice transformée suivant les opérations décrites ci-dessous, on considère le nouveau graphe associé pour y chercher le flot maximal, donc le couplage maximal.

Par rapport au graphe précédent, l'addition d'un arc $X_1 \rightarrow Y_2$ aura pour effet :

- soit d'augmenter le couplage si le sommet $\in Y_2$ n'était pas saturé dans le couplage précédent.
- soit d'augmenter le nombre de sommets de $Y_1$, dans le cas contraire : en effet, le sommet de l'arc ajouté appartenant à $Y_2$ devient alors marqué, et doit être ajouté à $Y_1$; par ailleurs, certains arcs $X_2 \rightarrow Y_1$ peuvent disparaître, mais comme ils n'étaient pas saturés dans le couplage précédent, ce ne sont pas eux qui avaient entraîné le marquage des sommets de $Y_1$, qui restent marqués.

En conséquence, soit le couplage augmente, soit l'ensemble $Y_1$ possède plus de sommets qu'auparavant.

Mais alors, dans le second cas, comme la taille du couplage n'est pas encore celle que l'on recherche, c'est-à-dire $n$, il existe des arcs $Y \rightarrow Z$ non saturés et au bout d'un certain temps d'opérations analogues (addition d'arc $X_1 \rightarrow Y_2$), comme $Y_1$ augmente, on réussira à marquer $z$ et donc à augmenter le couplage.

On voit ainsi (sans que ce qui précède constitue une démonstration) que le processus est convergent et qu'on aboutira au bout d'un nombre fini d'étapes à un couplage de zéros de taille égale à $n$ (ici 6).

Pratiquement, il n'est pas commode de changer l'ordre des rangées de la matrice des coûts. Pour constituer les différents tableaux $[X, Y]$ considérés ci-dessus, on procède de la façon suivante:

Sur la matrice 3, où les zéros du couplage maximal sont encadrés, les autres barrés,

a) on marque toute ligne n'ayant pas de zéro encadré (ces lignes correspondent à des sommets insaturés de $X$, donc à des sommets de $X_1$).

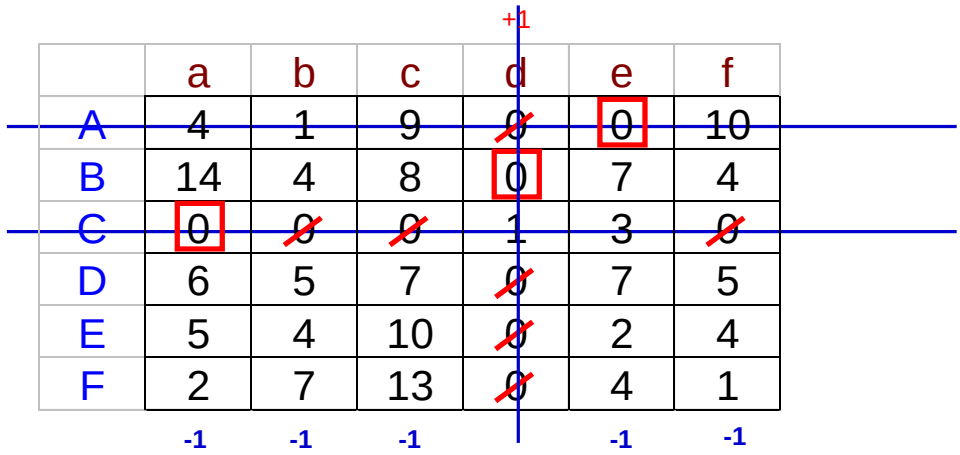
b) on marque toute colonne ayant un zéro barré dans une ligne marquée (la ligne étant marquée correspond à un sommet marqué; l'existence d'un zéro barré dans la colonne envisagée signifie l'existence d'un arc de flot nul entre le sommet marqué associé à la ligne et le sommet associé à la colonne; on peut donc marquer ce dernier).

c) on marque toute ligne ayant un zéro encadré sur une colonne marquée (existence d'un arc de flot non nul entre un sommet de $X$ et un sommet de $Y_1$, donc marquage du sommet de $X$).

d) on revient en b).

Après épuisement de cette procédure, les lignes marquées correspondent à $X_1$ et les colonnes marquées à $Y_1$.

Sur la matrice 3, par exemple, cette procédure donne :




On retrouve bien $X_1 = \{B, D, E, F\}, X_2 = \{A, C\}, Y_1 = \{d\}$ *et* $Y_2 = \{a, b, c, d, e, f\}$

Remarquons que l'on pourrait se passer de cette procédure puisque le traitement du graphe associé $G$ nous a déjà fourni les ensembles $X_1, Y_1, X_2, Y_2$ ; mais parfois le couplage maximal est tellement évident qu'il ne nécessite pas la prise en considération de ce graphe et que l'on peut alors travailler directement sur la matrice des coûts.

À présent, nous savons qu'il faut isoler le tableau $[X_1, Y_2]$ pour y repérer l'élément de plus faible valeur.

On voit alors qu'il suffit sur la matrice précédente de rayer les lignes non marquées et les colonnes marquées pour obtenir le tableau restant $[X_1, Y_2]$. On vérifie que tous les éléments non rayés sont bien strictement positifs. On cherche alors le plus petit de ces éléments positifs; c'est ici l'élément $(F, f)$ égal à 1. On sait dans ces conditions :

a) qu'on le retranche des éléments non rayés $[X_1, Y_2]$.

b) qu'on l'ajoute aux éléments $[X_2, Y_1]$ c'est-à-dire aux éléments rayés deux fois.

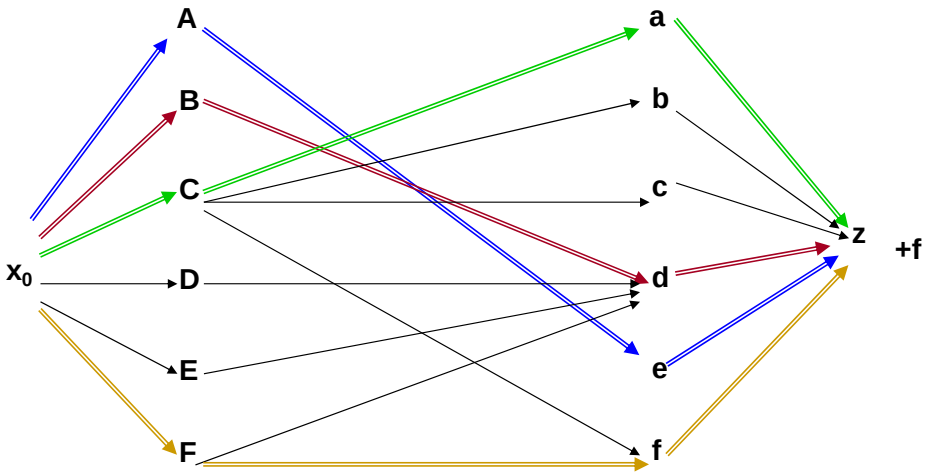
c) qu'on ne touche pas aux éléments $[X_1, Y_1]$ et $[X_2, Y_2]$ c'est-à-dire aux éléments rayés une seule fois.

Ces opérations donnent alors la nouvelle matrice des coûts, la matrice 4 :

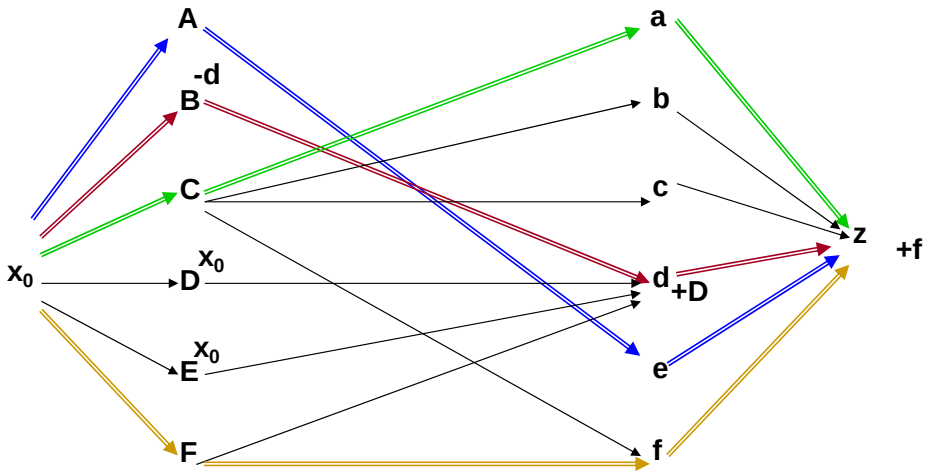
| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 4 | 1 | 9 | 1 | 0 | 10 |
| B | 13 | 3 | 7 | 0 | 6 | 3 |
| C | 0 | 0 | 0 | 2 | 3 | 0 |
| D | 5 | 4 | 6 | 0 | 6 | 4 |
| E | 4 | 3 | 9 | 0 | 1 | 3 |
| F | 1 | 6 | 12 | 0 | 3 | 0 |

On retourne avec cette nouvelle matrice de coûts, équivalente aux premières, à la phase 2, pour trouver sur le nouveau graphe associé le couplage maximal.

Pour cela, on peut partir du couplage obtenu précédemment, et examiner si on peut l'améliorer.



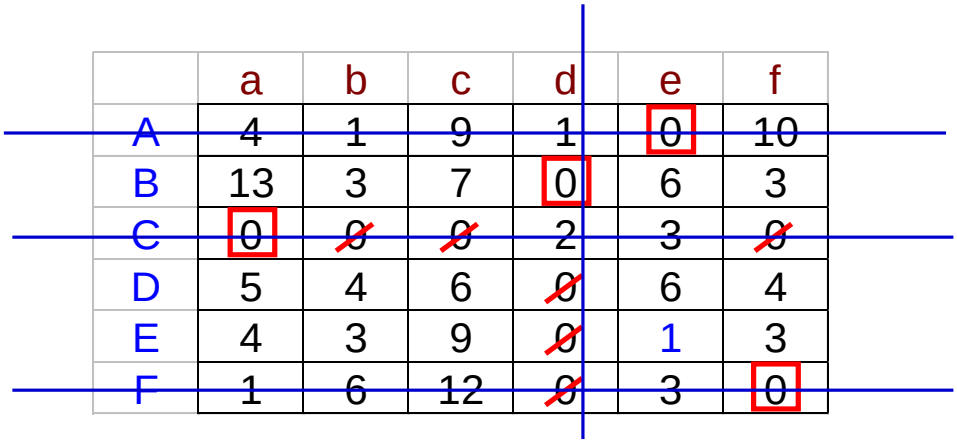

Il est évident que l'on peut remarquer $Z$; après amélioration, on obtient :




On ne peut plus marquer $Z$ avec ce flot et le couplage est donc maximal : c'est le couplage $(A,e),(B,d),(C,a),(F,f)$. Il est de taille 4 et il conviendra d'essayer de l'améliorer encore.

Pour cela, on transforme la matrice 4 suivant les opérations exposées ci-dessus, après avoir encadré les zéros du couplage et barré les autres zéros.

Sur la matrice 4, l'isolement du tableau $[X_1, Y_2]$ donne :

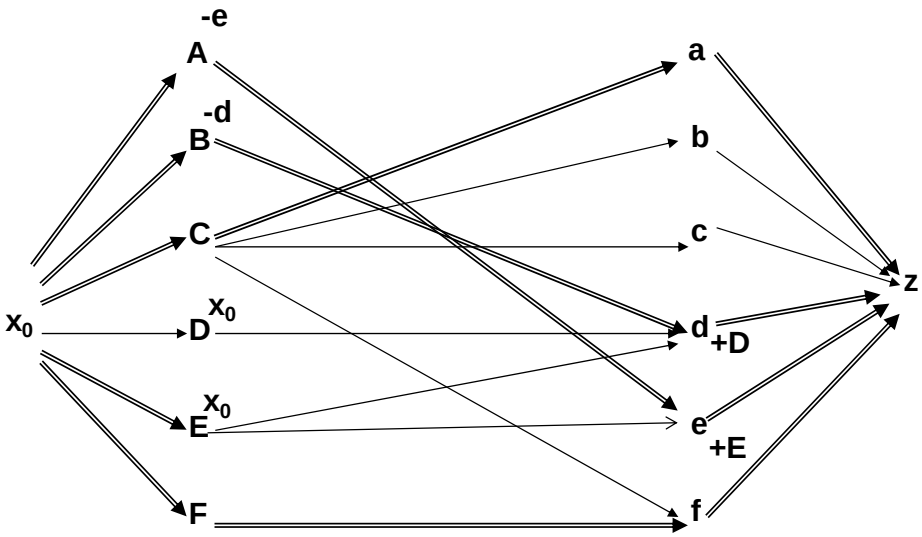


| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 4 | 1 | 9 | 1 | 0 | 10 |
| B | 13 | 3 | 7 | 0 | 6 | 3 |
| C | 0 | *0* | *0* | 2 | 3 | *0* |
| D | 5 | 4 | 6 | *0* | 6 | 4 |
| E | 4 | 3 | 9 | *0* | 1 | 3 |
| F | 1 | 6 | 12 | *0* | 3 | 0 |

Le plus petit élément de $[X_1, Y_2]$ est l'élément $(E, e)$ égal à 1. On l'enlève aux éléments non rayés et on l'ajoute aux éléments rayés deux fois, pour obtenir la matrice 5.

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 4 | 1 | 9 | 2 | 0 | 10 |
| B | 12 | 2 | 6 | 0 | 5 | 2 |
| C | 0 | 0 | 0 | 3 | 3 | 0 |
| D | 4 | 3 | 5 | 0 | 5 | 3 |
| E | 3 | 2 | 8 | 0 | 0 | 2 |
| F | 1 | 6 | 12 | 1 | 3 | 0 |

On continue alors en revenant à la phase 2; graphe associé à la matrice 5, et muni du couplage précédent :



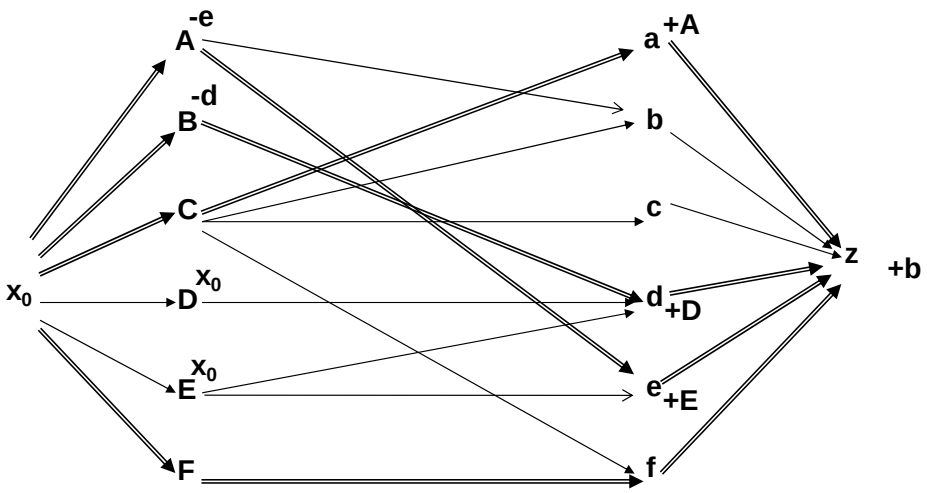

On n'arrive pas à marquer $Z$. Le couplage précédent reste optimal. On a ici l'exemple d'une itération sans progrès apparent, puisque la taille du couplage n'augmente pas. Cependant, comme on l'avait annoncé, c'est la taille de l'ensemble $Y_1$ qui augmente, passant de 1 à 2. La matrice 5 décomposée, donne :

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 4 | 1 | 9 | 2 | 0 | 10 |
| B | 12 | 2 | 6 | 0 | 5 | 2 |
| C | 0 | 0 | 0 | 3 | 3 | 0 |
| D | 4 | 3 | 5 | 0 | 5 | 3 |
| E | 3 | 2 | 8 | 0 | 0 | 2 |
| F | 1 | 6 | 12 | 1 | 3 | 0 |

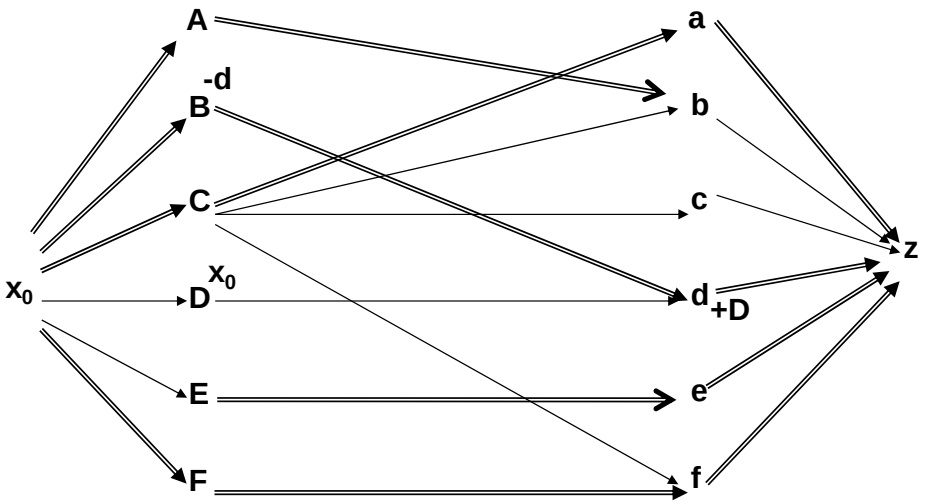
Plus petit élément : $(A, b) = 1$ ; nouvelle matrice 6

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 3 | 0 | 8 | 2 | 0 | 9 |
| B | 11 | 1 | 5 | 0 | 5 | 1 |
| C | 0 | 0 | 0 | 4 | 4 | 0 |
| D | 3 | 2 | 4 | 0 | 5 | 2 |
| E | 2 | 1 | 7 | 0 | 0 | 1 |
| F | 1 | 6 | 12 | 2 | 4 | 0 |

Nouveau graphe associé :



On réussit à marquer $Z$ par la chaîne $x_0$EeAbZ. L'amélioration du flot donne le graphe :




Ce flot est optimal, puisqu'on ne marque pas $Z$. La décomposition de la matrice 6 donne :

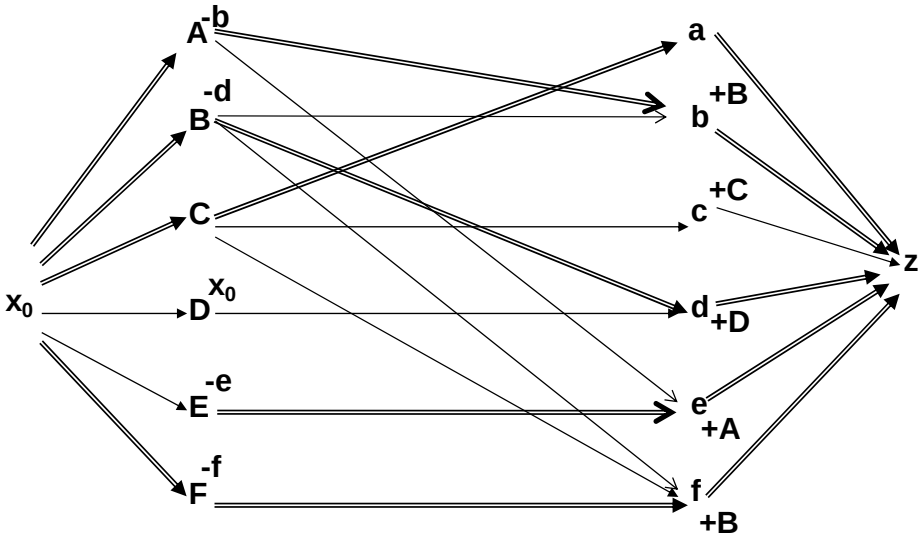
| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 3 | 0 | 8 | 2 | 0 | 9 |
| B | 11 | 1 | 5 | 0 | 5 | 1 |
| C | 0 | 0 | 0 | 4 | 4 | 0 |
| D | 3 | 2 | 4 | 0 | 5 | 2 |
| E | 2 | 1 | 7 | 0 | 0 | 1 |
| F | 1 | 6 | 12 | 2 | 4 | 0 |

Plus petits éléments : $(B,b) et\ (B,f) = 1$.

Nouvelle matrice : matrice 7.

| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 3 | 0 | 8 | 3 | 0 | 9 |
| B | 10 | 0 | 4 | 0 | 4 | 0 |
| C | 0 | 0 | 0 | 5 | 4 | 0 |
| D | 2 | 1 | 3 | 0 | 4 | 1 |
| E | 2 | 1 | 7 | 1 | 0 | 1 |
| F | 1 | 6 | 12 | 3 | 4 | 0 |

Nouveau graphe associé:




On n'arrive pas à marquer $Z$, le couplage précédent reste optimal.

Décomposition de la matrice 7 :

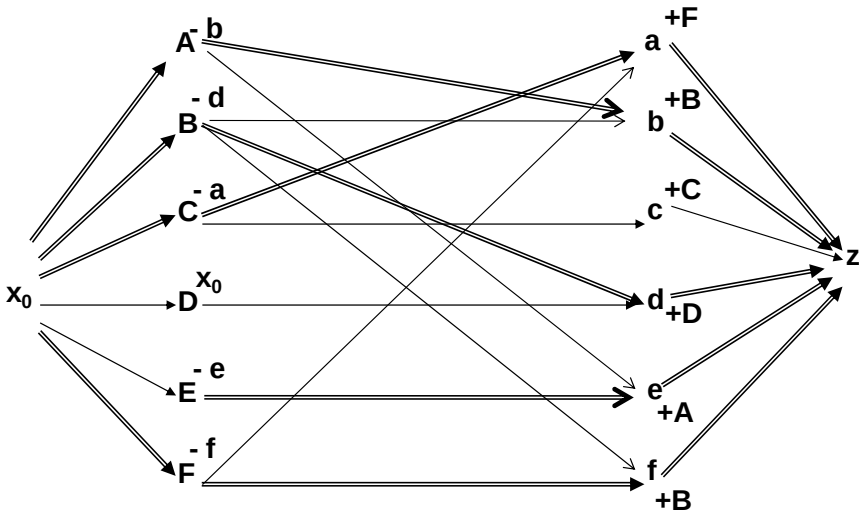
| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 3 | 0 | 8 | 3 | ~~0~~ | 9 |
| B | 10 | ~~0~~ | 4 | 0 | 4 | ~~0~~ |
| C | 0 | ~~0~~ | ~~0~~ | 5 | 4 | 1 |
| D | 2 | 1 | 3 | ~~0~~ | 4 | 1 |
| E | 2 | 1 | 7 | 1 | 0 | 1 |
| F | 1 | 6 | 12 | 3 | 4 | 0 |

Plus petit élément : $(F, a) = 1$

Nouvelle matrice des coûts : Matrice 7

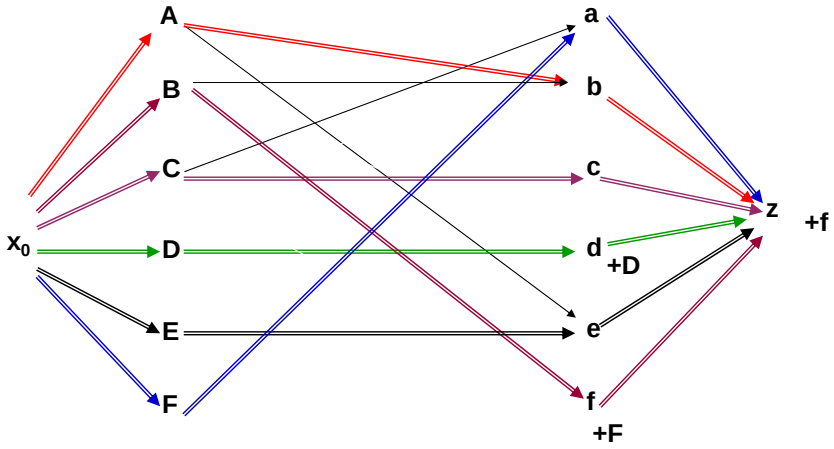
| | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| A | 2 | 0 | 7 | 3 | 0 | 9 |
| B | 9 | 0 | 3 | 0 | 4 | 0 |
| C | 0 | 1 | 0 | 6 | 5 | 1 |
| D | 1 | 1 | 2 | 0 | 4 | 1 |
| E | 1 | 1 | 6 | 1 | 0 | 1 |
| F | 0 | 6 | 11 | 3 | 4 | 0 |

Graphe associé:




On peut sur ce graphe marquer le sommet $Z$ par la chaîne $x_0DdBfFaCcZ$.

On améliore donc le flot qui devient:




On obtient en conséquence un couplage maximal de taille 6; cela veut dire que nous avons obtenu l'optimum, la matrice des coûts 8 étant équivalente à la matrice1, et la fonction économique calculée sur la matrice 8 à partir de l'affectation donnée par le couplage obtenu étant la valeur nulle.

Cette affectation est :

| |
|---|
| (A, b) |
| (B,f) |
| (C,c) |
| (D,d) |
| (E,e) |
| (F,a) |

Le coût de cette affectation, calculé sur la matrice initiale 1 est égal à 133.

Cet exemple a été choisi parce que l'application de l'algorithme hongrois permettait d'arriver à la solution en un nombre d'itérations important pour la dimension du problème (on a modifié 7 fois la matrice initiale).

Il faut cependant se rendre compte que traiter ce petit exemple par la programmation linéaire aurait conduit à des calculs beaucoup plus pénibles !

D'une façon générale, dans beaucoup de problèmes de recherche opérationnelle, on aboutit à une formalisation du type de programmation linéaire. Ce phénomène peut-être expliqué par une utilisation (souvent abusive) des concepts de la comptabilité dans les calculs de coûts. Quoiqu'il en soit, il convient toujours, avant de se lancer dans les applications souvent lourdes des algorithmes de la programmation linéaire, d'examiner si la nature du problème ne pourrait pas conduire à des résolutions plus aisées, à l'aide de la théorie des graphes, par exemple.

# Chapitre 10 • Les chaînes de Markov

Dans cette partie du cours, nous allons aborder le problème du traitement des phénomènes aléatoires dans les processus de gestion.

Par phénomène aléatoire, on entendra tout phénomène auquel on peut associer, pour le formaliser, une ou plusieurs variables aléatoires. On distinguera par conséquent l'aléatoire de l'incertain, pour lequel il n'est pas possible de faire intervenir des distributions de probabilités. Les problèmes de choix en avenir incertain ne seront d'ailleurs pas du tout abordés ici, ce qui constituera une restriction importante, dans la mesure où, pour bon nombre de processus réels, l'hypothèse de l'existence de distributions de probabilité est parfois hardie. De même, on ne traitera pas ici de problèmes de choix en situation concurrentielle, où le décideur ne sait pas non plus affecter des probabilités aux évènements dans la mesure où ces évènements dépendent des autres acteurs. Il s'agit là du champ de la Théorie des Jeux.

L'étude des phénomènes aléatoires, sous l'angle de la Recherche Opérationnelle, se rattache en principe à la discipline mathématique dite « Processus stochastique », où l'on étudie des variables aléatoires dépendant du temps. Il s'agit là d'une discipline assez difficile, donnant souvent lieu à des développements mathématiques raffinés. Comme pour les autres chapitres (programmation linéaire et graphes), nous éviterons ici de nous attacher aux difficultés théoriques, qui sont nombreuses, pour envisager les phénomènes étudiés sous un angle beaucoup plus pratique, quitte à perdre sur le plan de la rigueur. Encore une fois, le but essentiel ici est de montrer comment certains problèmes se réclamant du terme « gestion » peuvent se formaliser, et donc se lire d'une certaine façon.

Les chapitres suivants seront structurés non pas suivant un cadre théorique préétabli, mais suivant une série de questions pouvant relever du calcul des probabilités (stocks, fiabilité, files d'attente ...).

Toutefois, le chapitre 10 sera consacré à une classe particulière de processus stochastiques ne se rattachant pas à une question particulière, mais intervenant dans bon nombre de problèmes : les chaînes de Markov.

Rappelons qu'en langage probabiliste, un processus stochastique se définit de la façon suivante:

Soit un ensemble $T$, engendré par l'indice $t$ et une variable aléatoire $X_t$, indicée par $t$. On appelle processus stochastique l'ensemble : $\{X_t, t \in T\}$.

$t$ aura très fréquemment la signification du temps. $T$ sera dit alors l'ensemble - temps. Si $X_t$ est pris dans un ensemble $X$ ($R$ ou $R^n$ par exemple), $X$ sera l'ensemble - état.

$T$ et $X$ peuvent être finis ou infinis, discrets ou continus. Les combinaisons de ces différentes qualités fournissent une classification possible des processus stochastiques.

Nous étudierons dans ce premier chapitre les processus stochastiques où $T$ et $X$ sont discrets et dénombrables : ces processus stochastiques seront appelés suites stochastiques à ensemble-état discret, le terme suite signifiant que l'ensemble temps est discret, et un cas particulier important sera constitué par **Les chaînes de Markov.**

## 10.1. DEFINITION D'UNE CHAINE DE MARKOV

### 10.1.1. Suite stochastique

Soit un système pouvant se trouver dans un certain nombre d'états dénombrables, $E_1, E_2, \ldots E_i \ldots$ et cela à certaines périodes ou instants, également dénombrables, numérotés $0, 1, 2, \ldots t \ldots \ (t \in N)$.

Entre l'instant $t \; et \; t+1$, le système passe aléatoirement d'un état $E_i$ à un état $E_j$. Définissons la variable aléatoire $X_t$ de la façon suivante:

$X_t = i$ signifie qu'à la date $t$, le système se trouve dans l'état $E_i$. L'ensemble :

$\{X_t \quad t = 0, 1, 2, \ldots n, \ldots\}$ est une suite stochastique à ensemble-état discret.

**Exemples :**

1. Sur un chantier fonctionne un certain nombre de machines. Pendant un jour donné, certaines peuvent tomber en panne. Elles sont alors envoyées le soir du jour considéré à l'atelier de réparation. La durée de réparation est aléatoire, mais les machines, une fois réparées, repartent au chantier pour y arriver le matin qui suit la fin de leur réparation.

   On peut alors s'intéresser au nombre de machines en fonctionnement sur le chantier pour une matinée quelconque : l'état $E_k$, signifie qu'il y a $k$ machines en fonctionnement, et l'instant $t$ est la $t^{ième}$ matinée considérée. $X_t$, est alors le nombre (aléatoire, évidemment) de machines en fonctionnement à la matinée $t$.

2. Un médecin a un planning de rendez-vous, pour ses consultations, où les patients sont inscrits dans l'ordre où ils se présentent.

   S'il n'accepte d'examiner par jour qu'un contingent limité de malades, il existe en général des délais de rendez-vous. Le nombre de malades prenant rendez-vous pendant une journée considérée étant aléatoire, on peut considérer par exemple la variable aléatoire $X_t$ suivante :

   $X_t = k$ signifie qu'au début de journée $t$, le délai de rendez-vous est $k$, c'est-à-dire que le premier malade téléphonant pour prendre rendez-vous ne pourra être reçu que dans $k$ jours.

   On pourrait aussi prendre pour variable $X_t$ le nombre total d'inscrits sur le carnet de rendez-vous au début du $t^{ième}$ jour.

### 10.1.2. Chaîne de Markov

Les chaînes de Markov sont des suites stochastiques particulières : on appellera chaîne de Markov une suite stochastique à ensemble-état discret telle que l'évolution du système entre l'instant $t-1$ ne dépend pas des états du système avant l'état $t-1$ et cela pour tout instant $t$. Autement dit, avec les notations classiques des probabilités, on a :

$$P[X_t = k/\, X_0 = i_0,\, X_1 = i_1 \ldots X_{t-1} = i_{t-1}]$$
$$= P\ [X_t = k\ \ | X_{t-1} = i_{t-1}]$$

$P[A/B]$ étant la probabilité de l'évènement $A$ conditionné par l'évènement $B$.

**Remarque** : plus généralement, on dit qu'on a affaire à une chaîne de Markov d'ordre $m$ si l'on a :

$$P\ [X_t = k \mid X_0 = i_0,\ X_1 = i_1, \ldots . X_{t-1} = i_{t-1}]$$

$$= P\ [X_t = k\ \mid X_{t-m} = i_{t-m}, X_{t+1-m} = i_{t+1-m} \ldots . X_{t-1} = i_{t-1}]$$

En d'autres termes : l'état à une période quelconque dépend des états rencontrés pendant les $m$ périodes précédentes. Nous nous contentons ici d'étudier les chaînes de Markov d'ordre 1.

Pour les exemples considérés ci-dessus, on voit que le premier ne correspond pas en général à un processus markovien : la durée de réparation étant aléatoire et pouvant dépasser plusieurs jours, le nombre de machines en fonctionnement, pour une matinée donnée, dépend non seulement de ce qui s'est passé le jour précédent, mais également des autres jours.

Pour le second, si les nombres de patients prenant rendez-vous chaque jour sont des variables indépendantes, on voit que, au début d'une journée quelconque, la loi de probabilité de la variable $X_t$ (nombre de malades inscrits par exemple), ne dépend que de l'état du carnet de rendez-vous au début de la journée précédente. On a bien ici affaire à une chaîne de Markov.

Le problème que l'on se pose dans une chaîne de Markov est de connaître la distribution de probabilité de $X_t$ pour tout $t$, c'est-à-dire les quantités :

$P[X_t = i] = P_i(t)$

On voit que si l'on connaît la distribution de probabilités initiale, c'est-à-dire :

$P[X_0 = i] = P_i(0)$

$t = 0$ étant l'instant de démarrage du processus, et si l'on connaît toutes les probabilités

$P[X_t = j/X_{t-1} = i] = P_{ij}(t)$

appelées *probabilités de transition* à la date $t$, on peut résoudre complètement ce problème. On a en effet pour tout $t$

$$P_j\ (t) = \sum_I P_i\ (t-1)\ P_{ij}\ (t)$$

On peut calculer ainsi de proche en proche $P_i(1), P_i(2), etc$.

Dans la suite de l'exposé, nous nous limiterons à la classe des chaînes de Markov dites *homogènes*, c'est-à-dire telles que les probabilités de transition $P_{ij}(t)$ ne dépendent pas du temps. On posera alors : $P_{ij}(t) = P_{ij}$, et on aura :

$$P_j(t) = \sum_i P_i(t-1)P_{ij}$$

Nous nous limiterons encore aux chaînes de Markov finies, c'est-à-dire telles que le nombre d'états possibles est fini et en nombre $m$. Dans ces conditions,

$$P_j(t) = \sum_{i=1}^{m} P_i(t-1)\ P_{ij} \qquad (1)$$

On peut donner une expression plus synthétique de cette formule de récurrence. Si on appelle $P(t)$ le vecteur ligne constitué par les $P_i(t)$.

$$P(t) = [P_1(t),\ P_2(t) \ldots . P_i(t) \ldots . P_m(t)]$$

et $M$ la matrice $m \times m$ des $P_{ij}$ : $M = (P_{ij})$

($P_{ij}$ est la probabilité de passer de l'état $E_i$ à l'état $E_j$ dans une phase quelconque du processus). On peut écrire, d'après (1)

$$P(t) = P(t-1)M \qquad (2)$$

ce qui donne immédiatement :

$$P(t) = P(0)M^t \qquad (3)$$

$P(t)$ *est le vecteur d'état à l'instant* $t$.

$M$ est appelée *matrice de transition de la chaîne de Markov*.

D'une façon générale, une telle matrice, dont les éléments sont tous positifs ou nuls, et dont la somme des éléments d'une ligne quelconque est égale à 1[8] est appelée *matrice stochastique.* $P(t)$ est, de la même façon appelé *vecteur stochastique*. Chacun des éléments de ce vecteur est compris entre 0 et 1, et leur somme est égale à 1.

La formule (3), à partir du moment où l'on connaît $(0)$ et $M$, permet théoriquement de calculer $P(t)$ pour tout $t$.

---

[8]Attention : il n'y a aucune raison que la somme des éléments d'une colonne soit égale à 1. Pour un $j$ donné, on peut avoir par simple $P_{ij} = 0\ V_i$, ce qui signifie que l'état $j$ ne sera jamais atteint.

**Remarques importantes :**

1) Appelons $P_{ij}^{(t)}$ l'élément $(i,j)$ de la matrice $M^t$. Il est évident que $P_{ij}^{(t)}$ est égal à la probabilité de passer de l'état $E_i$. à l'état $E_j$ en exactement $t$ épreuves. On a

   $P_{ij}^{(1)} = P_{ij}$. $P_{ij}^{(t)}$seront appelées probabilités de transition d'ordre $t$.

2) On a les équations suivantes:

$$P_{ij}^{(t)} = \sum_{k=1}^{m} P_{ik}^{(t_1)} \; P_{kj}^{(t_2)} \qquad (4)$$

   pour tous $t_1$ $et$ $t_2$ entiers tels que $t_1 + t_2 = t$. Il s'agit des équations de Chapmann-Kolmogorov.

## 10.2. PROBLEME DU VECTEUR STOCHASTIQUE LIMITE

Comme nous l'avons déjà souligné, la formule (3) permet d'obtenir le vecteur $P(t)$ pour tout $t$. En fait, dans la plupart des cas, on observe des phénomènes qui se déroulent sur un grand nombre de périodes. Il suffit alors d'examiner ce que devient $P(t)$ lorsque $t$ devient grand, ou encore de calculer la limite de $P(t)$, si elle existe, lorsque $t$ tend vers l'infini.

Dans ces conditions, la limite de $P(t)$ lorsque $t$ tend vers l'infini, fournit les probabilités qu'a le système de se trouver dans les divers états $E_1, E_2 \ldots E_m$, et cela à n'importe quelle date suffisamment éloignée de l'instant initial du processus.

Par ailleurs on exige en général du vecteur limite, lorsqu'il existe, une autre qualité : si l'on appelle $P^*$ cette limite,

$$P^* = \lim_{t \to \infty} P(t)$$

on s'intéresse plus particulièrement aux chaînes de Markov pour lesquelles $P^*$ est indépendant de $P(0)$ c'est-à-dire de l'état du système à l'instant initial. De telles chaînes de Markov sont dites *ergodiques.*

L'intérêt de telles chaînes vient du fait que, d'une part le processus est *stationnaire,* au bout d'un temps suffisamment long, c'est-à-dire que la distribution des états du système reste constante au cours du temps, et que d'autre part, cette distribution ne dépend pas de l'instant initial qui est en général indéterminé. Cela dit, dans quelles conditions une chaîne de Markov est-elle ergodique ?

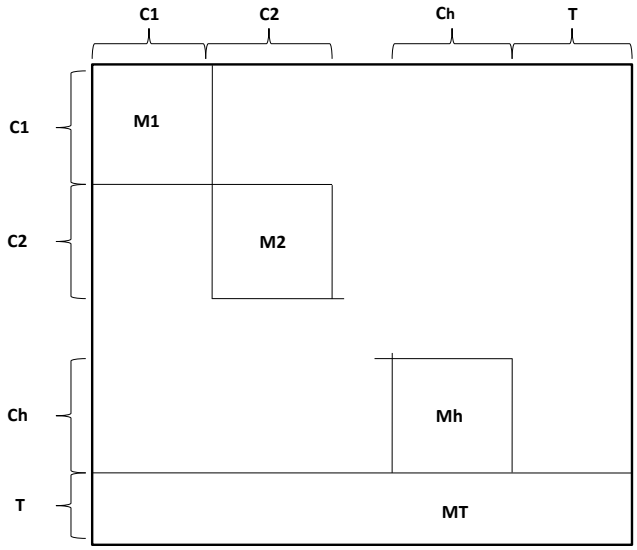
Nous allons donner ces conditions sans entrer dans les démonstrations correspondantes, mais en essayant de montrer à quels phénomènes simples renvoient les résultats annoncés.

## 10. 3. CONDITIONS D'ERGODICITE

Nous donnerons d'abord deux définitions que nous expliquerons par la suite.

### 10.3.1. Matrice stochastique réductible

Une matrice stochastique $M(m \times m)$ est dite réductible si l'on peut classer les différents états possibles $E_1 \dots E_m$ en $h+1$ classes $C_1, C_2 - C_h, T$ de telle façon que la matrice $M$ puisse s'écrire de la façon suivante (en réordonnant les états).




avec : $M_1, M_2, \dots M_h$ des matrices carrées non nulles, et $M_T$ une matrice rectangulaire.

Tous les éléments de $M$ n'appartenant à aucune des matrices $M_1, M_2, \dots M_h, T$ sont nuls. C'est-à-dire que pour un état $E_i$ appartenant à une classe $C_l$ on a:

$$P_{ij} = 0 \;\; si \;\; E_j \notin C_\ell \tag{5}$$

Une matrice qu'on ne peut décomposer de cette façon est dite *irréductible.*

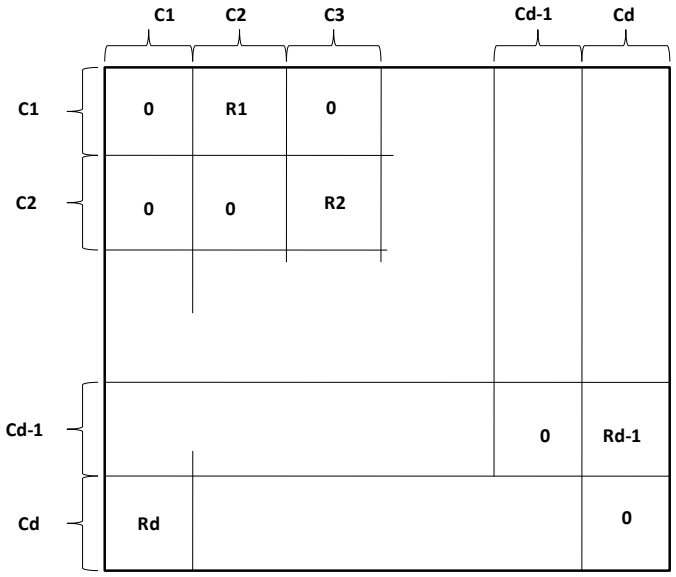
### 10.3.2. Matrices stochastiques périodiques

Plaçons-nous dans le cas où la matrice stochastique considérée est *irréductible.*

Une matrice stochastique $M$ irréductible est dite *périodique* si l'on peut trouver $d$ sous-ensembles d'états $C_1, C_2 \dots C_d$, chacun étant de cardinal

$$r_1, r_2, \dots r_d (r_1 + r_2 + \cdots r_d = m),$$

de telle façon que $M$ puisse s'écrire de la façon suivante (en réordonnant les éléments) :




où les 0 de la diagonale principale représentent des matrices carrées nulles,

$r_1 \times r_1, r_2 \times r_2, \ldots r_j \times r_j \ldots r_d \times r_d$, et où les matrices $R_j$ sont des matrices non nulles, rectangulaires de dimension $r_j \times r_{j+1}$, tous les autres éléments de $M$ sont nuls.

Une matrice $M$ qu'on ne peut mettre sous cette forme est dite *apériodique.*

### 10.3.3. Théorème

**Pour qu'une chaîne de Markov soit ergodique, il faut et il suffit que la matrice stochastique qui lui correspond soit irréductible et apériodique.**

Donnons quelques indications permettant de comprendre pourquoi des matrices stochastiques réductibles ou périodiques ne correspondent pas à des chaînes de Markov ergodiques.

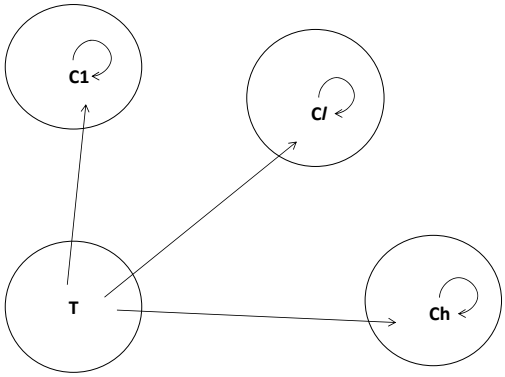
*a) Matrices réductibles*

Reprenons l'écriture d'une matrice réductible, telle que nous l'avons proposée ci-dessus; nous avons pu classer les différents états du système en $h+1$ sous-ensembles $C_1, C_2 \ldots C_h, T$ tels que:

Si $E_i \in C_l \; P_{ij} = 0 \; si \; E_j \notin C_l$

Par contre si $E_i \in T$ il existe au moins un $E_j$ n'appartenant pas à $T$ tel que $P_{ij} \neq 0$.

On peut symboliser cette propriété de la façon suivante:




Les flèches à l'intérieur des classes $C_l$ signifient que lorsque l'on se trouve en un état d'une de ces classes, on ne peut sortir de la classe. Par contre, il existe des possibilités de passages de $T$ dans certaines classes $C_l$ (pas forcément toutes). $C_1, C_l, \dots C_h$ seront appelées *ensembles d'états fermés.*

$T$ sera l'ensemble des *états transitoires.*

Dans ces conditions, si l'on peut effectivement décomposer les états d'une chaîne de Markov de cette façon, il est bien évident qu'on ne peut avoir d'ergodicité. En effet, il est clair qu'à chaque instant $t$, donc également lorsque $t$ devient infini, le vecteur d'état dépend de l'état initial. En particulier, si l'état initial est dans la classe d'état fermée $C_l$, on a $P_i(t) = 0 \; \forall t \; et \; i \;\; tel \; que \; i \notin C_l$.

Ce qui n'est plus vrai si l'on part d'autres classes $C_i$ ou encore de la classe $T$ des états transitoires.

Si aucune des matrices stochastiques $M_1, M_2, \dots M_h$ correspondant aux classes $C_1, C_l, \dots C_h$ n'est périodique, le vecteur $P(t)$ a bien une limite, mais qui dépend de $P(0)$ Cette limite, par ailleurs, n'est pas facile à calculer[9]. Nous laisserons donc de côté cet aspect du problème.

**Exemple de matrice réductible :** Soit la matrice *M* stochastique suivante:

| | E1 | E2 | E3 | E4 | E5 | E6 |
|---|---|---|---|---|---|---|
| E1 | 0 | 0,3 | 0,4 | 0,3 | 0 | 0 |
| E2 | 0 | 1 | 0 | 0 | 0 | 0 |
| E3 | 0 | 0 | 0 | 0 | 0,5 | 0,5 |
| E4 | 0,6 | 0 | 0,2 | 0 | 0,2 | 0 |
| E5 | 0 | 0 | 0 | 0 | 0 | 1 |
| E6 | 0 | 0 | 1 | 0 | 0 | 0 |

[9] Ce qui est intuitif (et vrai) est que lorsque $t$ tend vers l'infini $P_i(t) \to 0$ pour tout $i$ tel que $E_i \in T$ (d'où l'appellation « états transitoires »).

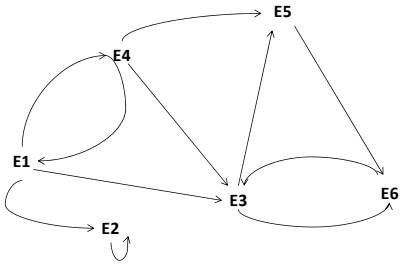
On peut réordonner les états de façon à suivre $M$ sous la forme suivante:

| | E2 | E3 | E5 | E6 | E1 | E4 |
|---|---|---|---|---|---|---|
| E2 | 1 | 0 | 0 | 0 | 0 | 0 |
| E3 | 0 | 0 | 0,5 | 0,5 | 0 | 0 |
| E5 | 0 | 0 | 0 | 1 | 0 | 0 |
| E6 | 0 | 1 | 0 | 0 | 0 | 0 |
| E1 | 0,3 | 0,4 | 0 | 0 | 0 | 0,3 |
| E4 | 0 | 0,2 | 0,2 | 0 | 0,6 | 0 |

On a fait apparaître deux classes d'états fermées:

$\{E_2\}$ $et$ $\{E_3, E_5, E_6\}$ et la classe d'états transitoires: $\{E_1, E_4\}$.

Sur de grosses matrices, une telle décomposition n'est pas toujours évidente. On utilise alors le graphe de transition associé à la chaîne de Markov: ce graphe est construit en associant un sommet $E_i$ à chaque état $E_i$ et en reliant $E_i$ à $E_j$ par un arc si $P_{ij} > 0$. Ainsi pour la chaîne ci-dessus, le graphe associé est :

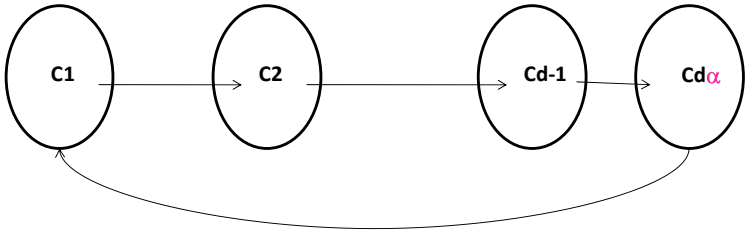



Les classes $C_1, C_2, \dots C_h$ d'états fermés correspondent alors aux sous-graphes fortement connexes de ce graphe. Elles peuvent être trouvées en utilisant la matrice booléienne associée au graphe (cf. partie II). Si $B$ est cette matrice, on l'élève à une puissance (booléienne) convenable (c'est-à-dire à une puissance $P$ telle que $P \geq m - 1$, si $m$ est le nombre d'états, donc la taille de la matrice). Il suffit alors d'isoler dans la matrice obtenue des sous-matrices carrées remplies de 1 pour obtenir une classe d'états fermés. Une matrice irréductible correspond à un graphe associé fortement connexe.

*b) Matrices périodiques*

Là encore, reprenons l'écriture générale d'une matrice périodique (cf. ci-dessus) et examinons ce qu'elle signifie pour les possibilités de passage d'un ensemble d'état à un autre.

Si l'on symbolise les $d$ sous-ensembles qui partitionnent l'ensemble d'états du système, on peut les schématiser de la façon suivante :




L'absence de flèches à l'intérieur des classes $C_i$ signifie que lorsque l'on se trouve en un des états d'une classe $C_i$, à l'étape suivante, on sort obligatoirement de la classe en question. Plus précisément, on passe en un des états de la classe $C_{i+1}$. On voit alors que si l'on part de la classe $C_1$, on y revient à coup sûr au bout de $d$ étapes, cela étant vrai pour n'importe quelle classe $C_i$. En conséquence, l'on peut dire que la périodicité des passages en une classe quelconque $C_i$ est égale à $d$ (d'où le nom donné à ce type de processus). Dans ce cas encore, la raison pour laquelle il ne peut y avoir d'ergodicité est évidente:

Si l'état initial est $E_i$ on est sûr que :$P_i^{(t)} = 0\ si\ t$ n'est pas un multiple de $d$

En revanche si $t = kd$ avec $k$ entier, alors $P_i^{(t)} \neq 0$.

En d'autres termes, le vecteur $P(t)$ n'est pas stable : suivant le reste de la division de $t$ par $d$, certains $P_i^{(t)}$ ne sont pas nuls (plus précisément, ceux relatifs aux $E_i$ appartenant à $C_{r+1}, si\ t = kd + r$ et tous les autres sont nuls). Il ne peut y avoir de limite. On démontre en fait qu'il peut y avoir une limite dans un sens plus général, c'est-à-dire au sens de Cesàro [10].

[10] En analyse réelle ou complexe, la moyenne de Cesàro d'une suite ($a_n$) est la suite obtenue en effectuant la moyenne arithmétique des *n* premiers termes de la suite. Le nom de Cesàro provient du mathématicien italien Ernesto Cesàro. Le théorème de Cesàro ou lemme de Cesàro précise que, lorsque la suite ($a_n$) a une limite, la moyenne de Cesàro possède la même limite.

**Exemple de chaînes périodiques** :

Soit la matrice :

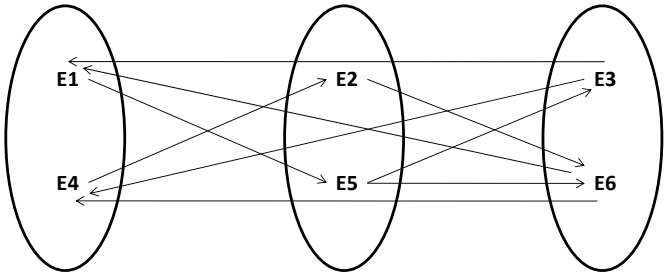
| | E1 | E2 | E3 | E4 | E5 | E6 |
|---|---|---|---|---|---|---|
| E1 | 0 | 0 | 0 | 0 | 1 | 0 |
| E2 | 0 | 0 | 0,5 | 0 | 0 | 0,5 |
| E3 | 0,4 | 0 | 0 | 0,6 | 0 | 0 |
| E4 | 0 | 1 | 0 | 0 | 0 | 0 |
| E5 | 0 | 0 | 0,2 | 0 | 0 | 0,8 |
| E6 | 0,3 | 0 | 0 | 0,7 | 0 | 0 |

On peut réordonner cette matrice de la façon suivante :

| | E1 | E4 | E2 | E5 | E3 | E6 |
|---|---|---|---|---|---|---|
| E1 | 0 | 0 | 0 | 1 | 0 | 0 |
| E4 | 0 | 0 | 1 | 0 | 0 | 0 |
| E2 | 0 | 0 | 0 | 0 | 0,5 | 0,5 |
| E5 | 0 | 0 | 0 | 0 | 0,2 | 0,8 |
| E3 | 0,4 | 0,6 | 0 | 0 | 0 | 0 |
| E6 | 0,3 | 0,7 | 0 | 0 | 0 | 0 |

On voit qu'on a pu mettre la matrice $M$ sous la forme générale d'une matrice périodique, dégageant ainsi une périodicité égale à 3.

On peut aussi tracer le graphe associé à $M$ :



La mise en évidence des classes d'états $C_i$ d'une matrice périodique n'est pas toujours évidente : elle se fait par la détermination des circuits passant par un sommet quelconque du graphe (voir partie II).

## 10.4. RECHERCHE DU VECTEUR D'ETAT LIMITE

Nous nous placerons dans le cas où le vecteur d'état $P(t)$ a une limite $P^*$ indépendante de $P(0)$, c'est-à-dire dans le cas de l'ergodicité. La matrice $M$ correspondante est irréductible et apériodique. Dans ces conditions, le calcul de $P^*$ peut s'effectuer de façon intuitive comme si l'on procédait sur une suite du type :

$\cup_n + 1 = f(\cup_n)$

On sait que si $\cup_n$ a une limite, elle est trouvée par

$$\ell = f\ (\ell)$$

Ici, étant donné que l'on a :

$$P(t+1) = P(t) \times M$$

On trouvera $P^*$, limite de $P(t)$ lorsque $t$ tend vers l'infini par :

$$P^* = P^* \times M \qquad (6)$$

Posons alors $P^* = (\omega_1, \omega_2, \ldots\ \omega_l \ldots \omega_m)$

L'équation (6) fournit un système d'équations homogènes avec pour inconnues les $\omega_j$ qui peut s'écrire :

$$P^*(I - M) = 0 \qquad (7)$$

La solution triviale $P^* = 0$ n'est pas unique, car le déterminant de la matrice $I - M$ n'est pas nul. En effet, il est évident qu'une des valeurs propres de la matrice $M$ est 1 : si $V$ est le vecteur colonne :

$V = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ constitué uniquement de 1 on a $MV = V$ du fait de :

$$\sum_{j=1}^{m} P_{ij} = 1$$

ce qui montre que $V$ est un vecteur propre associé à la valeur propre 1[11].

Donc : $DET\ (I - M) = 0$

Le système (7) est dégénéré. Pour avoir les $\omega_j$ il faut ajouter à (7) l'équation triviale :

$$\sum_{j=1}^{m} \omega_j = 1$$

On démontre que le système a une solution et une seule. Il s'écrit :

$$\sum_{i=1}^{m} P_{ij}\ \ \omega_i = \omega_j \qquad\qquad j = 1 \ldots m$$

$$\sum_{j=1}^{m} \omega_i \qquad = 1 \qquad\qquad (8)$$

**Remarque :** Une autre façon de raisonner aurait été de chercher la limite de la matrice $M^{(t)}$ lorsque $t$ tend vers l'infini, puisque

$$P(t) = P(0) \times M^{(t)}$$

Dans ce cas si
$\Pi = \lim_{t \to \infty} M^t$
On peut écrire, en utilisant le même raisonnement qu'auparavant :
$\Pi \times M = \Pi$
si l'on prend une ligne particulière $\pi_i de\ \pi$ on a :

$\Pi_i \times M = \Pi_i$

Par ailleurs

$$\sum_{j=1}^{m} \Pi_{ij} = 1$$

ce qui signifie que $\pi$ est également solution du système (8).

En conséquence, toutes les lignes de la matrice $\pi$ sont identiques et égales au vecteur d'état limite $P^*$. Si l'on cherche alors $P^*$ par

$$P^* = \lim_{t \to \infty} P(0) \times M^t = P(0) \times \Pi$$

[11]Signalons ici que de nombreux résultats portent sur les valeurs propres des matrices stochastiques, et que ces résultats pourraient servir de base à une démonstration rigoureuse des développements que nous fournissons sans démonstration.

on a bien $P(0) \times \Pi = P^*$ puisque la somme des éléments de $P(0)$ est égale à 1, et que toutes les lignes de $\pi$ sont identiques à $P^*$.

**Exemple :** Nous allons calculer le vecteur d'état limite $P^*$ sur un petit exemple particulier.

Soit une machine qui a la probabilité $\lambda$ de tomber en panne pendant un jour donné. Si elle tombe en panne, elle est envoyée le soir à un atelier de réparation où elle est réparée en une journée avec la probabilité $\mu$ et, en deux journées avec la probabilité $1-\mu$. Une fois réparée, elle est remise en marche le lendemain matin du jour de la réparation.

Plaçons-nous au début d'une journée quelconque, et posons-nous la question de savoir quelle est la probabilité que la machine soit en fonctionnement ce jour donné. Nous avons trois états :

$F$: la machine est en fonctionnement au début du jour considéré.

$R_1$: la machine subit son premier jour de réparation.

$R_2$ : la machine subit son second jour de réparation.

On a alors la matrice de probabilités de transition suivante, entre la date $t$ (début du jour $t$) et la date $t+1$ ( début du jour $t+1$)

| | F | R1 | R2 |
|---|---|---|---|
| F | 1-$\lambda$ | $\lambda$ | 0 |
| R1 | $\mu$ | 0 | 1-$\mu$ |
| R2 | 1 | 0 | 0 |

Si nous nous plaçons en régime permanent, c'est-à-dire lorsque $t$ est suffisamment grand, cherchons le vecteur limite $P^* = (\omega_1, \omega_2, \omega_3)$ et la probabilité cherchée sera $\omega_1$. D'abord, il est tout à fait évident qu'ici la matrice $M$ n'est pas réductible ni périodique. En conséquence, le processus est ergodique, $P^*$existe bien, et est indépendant de $P(0)$.

On a :

$P^*M = P^*$

Soit

$\omega_1(1-\lambda) + \omega_2\mu + \omega_3 = \omega_1 \qquad (a)$

$\omega_1\lambda = \omega_2 \qquad (b)$

$\omega_2(1-\mu) = \omega_3 \qquad (c)$

On vérifie que ce système homogène est dégénéré puisque $(b)+(c)$ donne $(a)$.

A $(b)$ et $(c)$ adjoignons alors

$$\omega_1 + \omega_2 + \omega_3 = 1$$

On obtient par substitutions :

$$\omega_1 + \omega_1 \lambda + \omega_1 \lambda (1 - \mu) = 1$$

Soit la solution :

$$\omega_1 = \frac{1}{1 + \lambda + \lambda(1 - \mu)}$$

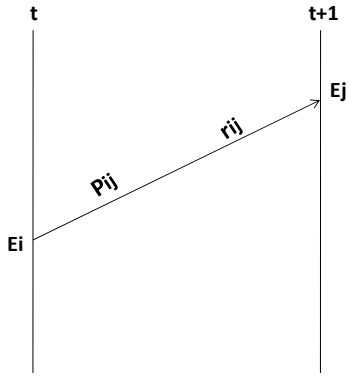
$$\omega_2 = \frac{\lambda}{1 + \lambda + \lambda(1 - \mu)}$$

$$\omega_2 = \frac{\lambda(1 - \mu)}{1 + \lambda + \lambda(1 - \mu)}$$

## 10.5. CHAINE DE MARKOV AVEC VALEURS DE TRANSITION

Dans les phénomènes économiques, il est en général peu intéressant de connaître la probabilité de tel ou tel état si l'on ne prend pas en compte les conséquences qui comptent au niveau des comparaisons entre différentes politiques possibles. C'est pourquoi, en général, on associe à une chaîne de Markov des *valeurs de transition* de la façon suivante :

Entre la date $t$ et $t + 1$, le passage de l'état $E_i$ à l'état $E_j$, qui se fait avec la probabilité $P_{ij}$ apporte un revenu (ou un gain) $r_{ij}$.




On voit que l'on fait la même hypothèse d'homogénéité sur les $r_{ij}$ que sur les $P_{ij}$, c'est-à-dire que l'on suppose que les gains $r_{ij}$ sont constants dans le temps. On pourrait bien sûr s'en passer mais les résultats obtenus seraient loin d'être aussi simples et frappants que dans le cadre de l'homogénéité :

Posons :

$$q_i = \sum_{j=1}^{m} P_{ij} \; r_{ij} \qquad (9)$$

$q_i$ est l'espérance de gain sur la période $(t\,;\, t+1)$ si l'on se trouve en $E_i$ à la date $t$. Appelons $q$ le vecteur colonne $q = \{q_i\}$.

Définissons maintenant une autre notion. Supposons que $t$ varie de 0 à $N$, c'est-à-dire que le nombre de phrases du processus est $N$.

Appelons $V_i(t)$ l'espérance de gain entre la date $t$ et la date finale $N$ si l'on se trouve en l'état $E_i$ à la date $t$.

$V_i(N)$ représente le gain associé au fait d'être en l'état $E_i$ à la date finale $N$. ($V_i(N)$ est souvent nul si l'on se contente des gains de transition $r_{ij}$).

Etablissons facilement une relation de récurrence. On a en effet :

$$V_i(t) = \sum_{j=1}^{m} P_{ij} \; [r_{ij} + V_j\,(t+1)]$$

$$V_i(t) = q_i + \sum_{j=1}^{m} P_{ij}\, V_j(t+1)$$

si l'on définit le vecteur colonne $V(t) = \{V_i(t)\}$, on a :

$V(t) = q + MV(t+1)$ (10)

si $M$ est la matrice stochastique des $P_{ij}$.

La formule (10) permet de calculer tous les $V_i(t)$ à partir du moment où l'on dispose des $P_{ij}, r_{ij}\ et\ V_j(N)$. On voit que pour cela, on décrit le temps à l'envers, en partant de l'état final. On a alors l'espérance de gain au début du processus donné par :

$$V(0) = [I + M + \ldots + M^{N-1}]\, q + M^N\, V(N) \qquad (11)$$

On voit qu'en général, si $N$ augmente indéfiniment, les composantes du vecteur $V(0)$, c'est-à-dire les espérances de gain total augmentent également indéfiniment. C'est pourquoi l'on préfère prendre en compte un autre critère, qui est l'espérance de gain par période ou phase, égale à :

$$g = \frac{V(0)}{N}$$

$$g = \frac{1}{N}[I + M + \ldots + M^{N-1}]\; q + \frac{M^N\, V(N)}{N} \qquad (12)$$

Il est facile de voir à partir de (12) que si la chaîne de Markov est irréductible et apériodique, et si $N$ tend vers l'infini, alors $g$ tend vers $\Pi_q$, $\Pi$ étant la limite de la matrice $M^N$. Les lignes de $\Pi$ étant toutes identiques et égales $P^* = (\omega_1, \omega_2 \ldots \omega_m)$ alors, tous les éléments de $g$ sont également identiques. Si $g_m$ désigne l'un de ces éléments, on a :

$$g_m = \sum_{j=1}^{m} \omega_j \, q_j \qquad (13)$$

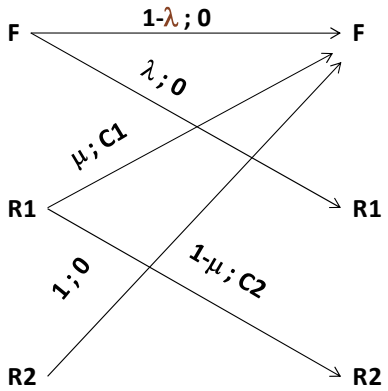
$g_m$ représente la moyenne du gain par phase si le processus se déroule sur un laps de temps important.

Si un agent économique retient ce critère (ce qui n'est pas évident), il peut choisir dans certains cas entre plusieurs solutions possibles. Revenons ainsi au petit exemple de la machine. Supposons que le coût de la panne soit :

$C_1$ si la réparation demande 1 jour

$C_2$ si la réparation demande 2 jours

avec $C_2 > C_1$




On a donc le graphe de transition suivant où les probabilités et les coûts sont indiqués :

On a donc dans ce cas :

$$q_1 = 0$$

$$q_2 = \mu C_1 + (1-\mu) C_2$$

$$q_3 = 0$$

par ailleurs on a vu que le vecteur d'état limite $P^*$ était :

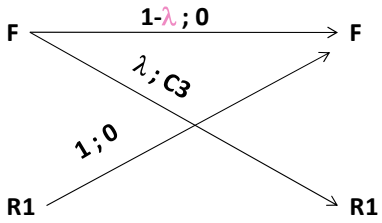
$$P^* = (\omega_1, \omega_2, \omega_3) = \frac{1}{1 + \lambda + \lambda(1-\mu)} \{1, \lambda + \lambda(1-\mu)\}$$

On a alors le coût moyen par période égal à :

$$g_m = \frac{1}{1+\lambda+\lambda(1-\mu)} \ (C_1\mu + C_2(1-\mu))$$

Supposons alors que l'atelier de réparation assure à l'entreprise qui gère la machine qu'elle réparera cette dernière en un jour exactement pour toutes les pannes, à condition que l'entreprise le rémunère pour cet engagement de telle façon que le coût de la panne devienne $C_3$ supérieur à $C_2$.

Si l'entreprise accepte cette solution, la chaîne de Markov devient :




On a $q_1 = \lambda C_3$

$q_2 = 0$

Par ailleurs :

$$P^* = \left( \frac{1}{1+\lambda} \ \frac{1}{1+\lambda} \right)$$

et le nouveau coût moyen par jour est $g'_m$.

$$g'_m = \frac{\lambda C_3}{1+\lambda}$$

Cette nouvelle politique est intéressante pour l'entreprise si

$$\frac{\lambda C_3}{1+\lambda} \leq \frac{\lambda}{1+\lambda+\lambda(1-\mu)} (C_1\mu + C_2(1-\mu))$$

c'est-à dire si

$$C_3 \leq \frac{1+\lambda}{1+\lambda+\lambda(1-\mu)} (C_1\mu + C_2(1-\mu))$$

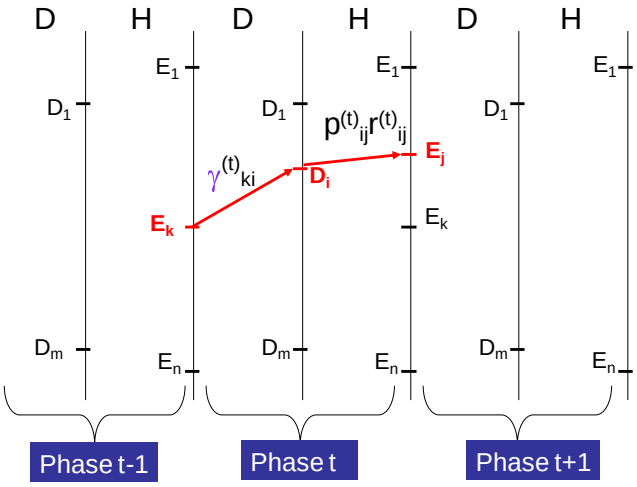
une condition nécessaire est en particulier

$$C_3 \leq C_1\mu + C_2\ (1-\mu)$$

Par la considération des coûts (ou gains) moyens par phase, on peut ainsi comparer entre elles plusieurs politiques, correspondant à plusieurs chaînes de Markov. Le problème se complique lorsque le hasard n'est pas seul à intervenir, c'est-à-dire lorsque, à chaque étape du processus, on peut essayer d'infléchir l'avenir par des décisions. On tombe alors dans le domaine de la programmation dynamique.

## 10.6. PROGRAMMATION DYNAMIQUE DISCRETE

Nous allons étudier le processus suivant, évoluant par phases, mais enrichi par rapport au précédent schéma (chaîne de Markov avec valeurs de transition) par le fait qu'une phase quelconque se décompose en deux étapes : une étape décision et une étape hasard. On a alors la configuration suivante :




En début de phase $t$, on peut se trouver en $n$ états $E_1 \ldots E_k \ldots E_n$. On prend une décision (étape décision) amenant en un état parmi $m$ possibles $D_1, \ldots D_i \ldots D_m$ ; le coût (ou le revenu) de cette décision est $\gamma_{ki}(t)$ si l'état de départ est $E_k$ et l'état « décidé »$D_i$.

À partir de cet état $D_i$ le système évolue de façon aléatoire (étape hasard) pour se trouver en fin de phase $t$ en un état $E_j$. On connaît $P_{ij}^{(t)}$, probabilité de passer de $D_i$ en $E_j$ et $r_{ij}^{(t)}$, coût où revenu correspondant.

Le processus continue alors de cette façon jusqu'à la phase terminale $T$.

**Remarques** *:*

a) On a supposé, provisoirement, pour simplifier les notations, que, à chaque phase $t$, on avait affaire aux mêmes états $D_1 \dots D_m$, $E_1 \dots E_n$. On aurait pu évidemment se passer de cette hypothèse en indiçant les états par $t$.

b) Le système peut se trouver en un état bien déterminé en phase 1 (début du processus) ou en phase $T$ (fin du processus). L'état de départ, comme l'état d'arrivée, peut au contraire ne pas être spécifié.

c) Dans la figure ci-dessus, l'étape décision précède l'étape hasard. Il s'agit d'un *processus D.H*. On aurait pu faire l'hypothèse inverse; il s'agirait alors d'un *processus H.D*. Dans le reste de l'exposé, nous n'envisagerons que des processus D.H, le passage aux H.D n'étant en rien difficile.

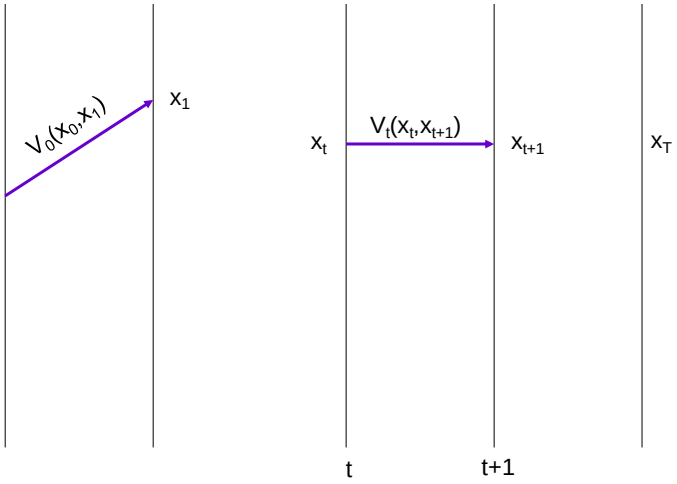
d) Un état $D_i$ peut ne pas être accessible à partir de n'importe quel état $E_k$. Lorsque l'on se trouve en un $E_k$, à la phase $t$, l'ensemble des états $D_i$ atteignables forment un sous-ensemble que l'on appelle $\Gamma^{(t)}(E_k)$. De même, en ce qui concerne l'étape hasard, il se peut qu'un état $E_j$ ne soit pas atteignable à partir d'un état $D_i$.

Cela dit, la question que l'on se pose pour ce type de processus est la suivante : supposons que, en début de phase $t$, l'on se trouve en l'état $E_k$. Quelle décision $D_i$ prendre alors? Plus précisément si, à chaque phase $t$, on associe à chaque état $E_k$ une décision $D_i$, de façon univoque, l'ensemble de ces correspondances s'appelle une *stratégie*. On recherche alors la stratégie optimale. Optimale par rapport à un critère que nous avons déjà utilisé dans le paragraphe précédent, et qui est celui de l'espérance mathématique.

Avant de passer à l'explicitation et la résolution de ce problème par le principe d'optimalité (1), il convient d'exposer brièvement, pour faciliter la compréhension de l'exposé, le cas où il n'y a pas d'étape hasard, c'est-à-dire le cas complètement déterministe (c'est le cas complémentaire de celui exposé en VI, qui était le cas du système complètement aléatoire).

### 10.6.1. Le cas déterministe

On a pour le cas déterministe la configuration suivante:




On a symbolisé les différents états possibles à la date $t$ par la seule variable $x_t$. Par exemple on a

$$x_t = \{E_0^{(t)}, E_1^{(t)} \dots E_{i(t)}^{(t)}\}$$

On remarquera que par ailleurs, cette écriture permet d'envisager la situation où la variable état est *continue.*

Lorsque l'on passe de $x_t$ à $x_{t+1}$ le coût de cette transition est $v_t(x_t, x_{t+1})$. Par ailleurs, les $x_{t+1}$ accessibles à partir de $x_t$ forment un sous-ensemble que l'on appellera $\Gamma(x_t)$.

Dans ces conditions, nous appellerons *politique* (et non plus stratégie, pour opposer le cas déterministe au cas aléatoire), un ensemble de $T-1$ choix possibles

$x_t \to x_{t+1}\ pour\ t = 0 \dots T-2$

$x_{t+1} \in \Gamma(x_t)$

**Remarque :**

On pourra toujours supposer $x_0$ et $x_T$ imposés. Si ce n'était pas le cas, il suffirait d'ajouter des phases supplémentaires « fictives » avant ou après le processus. Par exemple, si $x_T$ n'est pas fixé, il suffit d'ajouter une phase $T+1$ avec un seul état $x_{T+1}$ et $v_T(x_T, x_{T+1}) = 0\ \forall\ x_T$. On se ramène ainsi au cas où $x_T$ est fixé. Pour une politique déterminée, le coût total est :

$$V = \sum_{t=0}^{T-1} v_t(x_t, x_{t+1})$$

Il s'agit alors de trouver la politique qui minimise[12] $V$, soit trouver $x_1, x_2 \ldots x_{T-1}$ tels que

$$V = \sum_{t=0}^{T-1} v_t(x_t, x_{t+1})$$

soit minimal (ou maximal)

avec $\boldsymbol{x_{t+1}} \in \Gamma(\boldsymbol{x_t})$

Plaçons-nous maintenant à une date $i$ quelconque et en un état déterminé $x_i$. On appelle sous-politique de 0 à $i$ le choix de $x_1, x_2 \ldots x_{i-1}$, et on appelle sous-politique optimale de 0 à $i$ pour l'état $x_i$, le choix de $x_1, x_2 \ldots x_{i-1}$ qui minimise la fonction

$$V_i(x_i) = \sum_{k=0}^{i-1} v_k(x_k\ x_{k+1})$$

avec $x_{k+1} \in \Gamma(x_k)\ et\ x_i\ fixé$

$$V_i^*(x_i) = Min \sum_{k=0}^{i-1} v_k(x_k\ x_{k+1})$$

$x_1 \ldots x_{i-1}$

Nous allons alors appliquer le principe d'optimalité qui peut ici s'énoncer de la façon suivante :

**Une sous-politique optimale *de 0 à i* est formée d'une sous-politique optimale de 0 à $\boldsymbol{i-1}$.**

Ce principe est évident. Soit en effet $x_0, x_1^*, x_2^*, \ldots x_{i-1}^*, x_i$ ($x_0\ et\ x_i$ sont fixés) la sous politique optimale de 0 à $i$, pour $x_i$. La sous-politique de 0 à $i-1$, pour $x_{i-1}^*$ est optimale. En effet, s'il existait une sous_politique de 0 à $i-1$ meilleure, soit

$x_0, \overline{x}_1, \overline{x}_2 \ldots \overline{x}_{i-2}, x_{i-1}^*$ *l*a sous − politique de 0 à $i$

$x_0, \overline{x}_1, \overline{x}_2 \ldots \overline{x}_{i-2}, x_{i-1}^*$ $x_i$ serait la meilleure que la sous − politique.

$x_0, x_1^*, x_2^* \ldots x_{i-2}^*, x_{i-1}^*$ $x_i$ que l′on a supposé pourtant optimale.

---

[12] On pourrait traiter aussi des problèmes de maximisation : rien ne serait changé aux raisonnements; dans ce qui suit, il suffirait de changer l'opérateur *Min* par l'opérateur *Max*.

Ce principe va nous permettre de résoudre le problème posé. Supposons en effet que, à la date $i-1$, on connaisse les sous-politiques optimales de 0 à $i-1$, pour chaque état $x_{i-1}$. On a donc calculé :

$$V_{i-1}^*(x_{i-1}) = Min \sum_{k=0}^{i-1} v_k(x_k\ x_{k+1})$$

$$x_1, x_2 \ldots x_{i-2}$$

avec

$$x_{k+1} \in \Gamma(x_k)$$

Pour un $x_i$ donné, d'après le principe d'optimalité, on a alors, si $\Gamma^{-1}$ est l'application inverse de $\Gamma$ :

$$V_i^*(x_i) = Min\ [v_{i-1}(x_{i-1}\ x_i) + V_{i-1}^*(x_{i-1})]$$
$$x_{i-1} \in \Gamma^{-1}(x_i) \qquad (14)$$

Cette formule fondamentale permet de proposer l'algorithme suivant :

***Phase 1***

$$V_1^*(x_1) = v_0(x_0, x_1)$$

***Phase 2***

$$V_2^*(x_2) = Min\ \ [v_1(x_1\ x_2) + V_1^*(x_1)]$$
$$x_1 \in \Gamma^{-1}(x_2)$$

***Phase i***

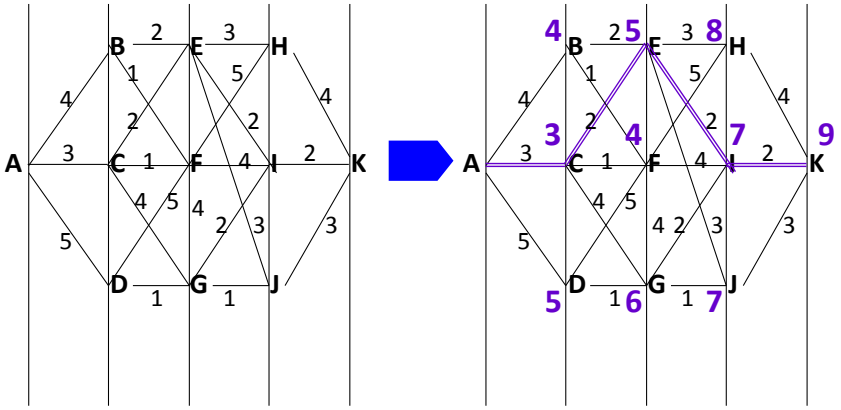
$$V_i^*(x_i) = Min\ \ [V_{i-1}(x_{i-1}\ x_i) + V_{i-1}^*(x_{i-1})]$$
$$x_{i-1} \in \Gamma^{-1}(x_i)$$

***Phase T***

$$V_T^*(x_T) = Min\ [V_{T-1}(x_{T-1}\ x_T) + V_{T-1}^*(x_{T-1})]$$
$$x_{T-1} \in \Gamma^{-1}(x_T)$$

$V_T^*$ est alors la valeur de l'optimum cherché. Pour trouver la politique optimale, il suffit de remonter le temps : la dernière équation, par laquelle on calcule $V_T^*(x_T)$, permet en même temps d'avoir $x_{T-1}^*$ qui assure le minimum de $V_T(x_T)$. L'avant dernière équation, par laquelle on calcule $V_{T-1}(x_{T-1}^*)$ permet de repérer $x_{T-2}^*$ etc.

Pour illustrer cet algorithme, nous allons prendre un exemple : soit à chercher le chemin de valeur minimale entre les sommets $A$ et $K$ du graphe non orienté suivant : (les valeurs des arcs sont indiquées)




On vérifie que les sommets sont répartis en cinq niveaux et que l'on se trouve bien devant un problème relevant de la programmation dynamique déterministe, où l'indice « temps » a simplement été remplacé par l'indice « niveaux ». On a par exemple si l'on utilise les notations précédentes :

$x_1 = \{B, C, D\}$

$x_2 = \{E, F, G\}$

etc.

$\Gamma(B) = \{E, F\}$

$\Gamma^{-1}(I)$={E, F, G}

etc.

$v_1(B, E) = 2$

$v_2(F, I) = 4$

etc.

Appliquons alors l'algorithme :

$V^*(A) = 0$

***Phase 1***

$V^*(B) = 4$ $V^*(C) = 3$ $V^*(D) = 5$

(Les valeurs des sous-politiques optimales $V^*$, pour chacun des sommets, sont entourées sur la figure).

***Phase 2***

$$V^*(E) = Min\left[(V^*(B) + v(B,E)\right), (V^*(C) + v(C,E))$$

$$= V^*(C) + v(C,E) = 5$$

$$V^*(\mathrm{F}) = \mathrm{Min}\left[(\mathrm{V}^*(\mathrm{B}) + \mathrm{v}(\mathrm{B},\mathrm{F})\right), (\mathrm{V}^*(\mathrm{C}) + \mathrm{v}(\mathrm{C},\mathrm{F})), (\mathrm{V}^*(\mathrm{D}) + \mathrm{v}(\mathrm{D},\mathrm{F})]$$

$$= V^*(C) + v(C,F) = 4$$

$$V^*(G) = Min\,[V^*(C) + v(C,G)), (V^*(D) + v(D,G))]$$

$$= V^*(D) + v\,(D,G) = 6$$

***Phase 3***

$$V^*(H) = Min\,[(V^*(E) + v(E,H)), (V^*(F) + v(F,H))$$

$$= V^*(E) + v(E,H) = 8$$

$$V^*(I) = Min\,[(V^*(E) + v(E,I)), (V^*(F) + v(F,I)), V^*(G) + v(G,I)]$$

$$= V^*(E) + v(E,I) = 7$$

$$V^*(J) = Min\,[(V^*(E) + v(E,J)), (V^*(G) + v(G,J)]$$

$$= V^*(G) + v(G,J) = 7$$

***Phase 4***

$$V^*(K) = Min\,[(V^*(H) + v(H,K)), (V^*(I) + v(I,K), V^*(J) + v(J,K))]$$

$$= V^*(I) + v(I,K) = 9$$

Le chemin optimal a donc pour valeur 9. Pour avoir les sommets de ce chemin, il suffit de reprendre le processus à l'envers : l'optimum 9 est obtenu par le sommet $I$. Le sous optimum en $I$ est obtenu par le sommet $E$ etc. On obtient le chemin $ACEIK$, en trait doublé sur la figure. Cela étant, quel est l'avantage du principe d'optimalité dans le cas déterministe? Le problème initial était :

$$\underset{x_1 \ldots x_{T-1}}{Min} \sum_{t=0}^{T-1} v_t(x_t, x_{t+1})$$

avec

$x_{t+1} \in \Gamma(x_t)$

Il s'agit donc de la minimalisation d'une fonction de $T-1$ variables.
Le principe d'optimalité a permis de remplacer ce problème par une série de problèmes de minimisation du type :

$$\underset{x_{i-1}}{\text{Min}}\, V_i(x_i) = v_{i-1}(x_{i-1}x_i) + V^*_{i-1}(x_{i-1})$$

$x_{i-1} \in \Gamma^{-1}(\mathrm{x_i})$

problème qui doit être résolu pour chaque $x_i$ de chaque période $i$, mais qui est un *problème de minimisation d'une fonction à une seule variable.*

Pour fixer un peu plus les idées, supposons que le nombre de valeurs de $x_i$ possibles (ou nombre d'états à la phase $i$), soit constant quel que soit $i$ et égal à $n$, et que, par ailleurs de la date $i-1$ à la date $i$, on puisse atteindre tous les états de la date $i$.

L'énumération de toutes les solutions possibles conduit à $n^{T-1}$ solutions, pour chacune desquelles il faut calculer une somme du type :

$$\sum_{t=0}^{T-1} V_t(x_t\ x_{t+1})$$

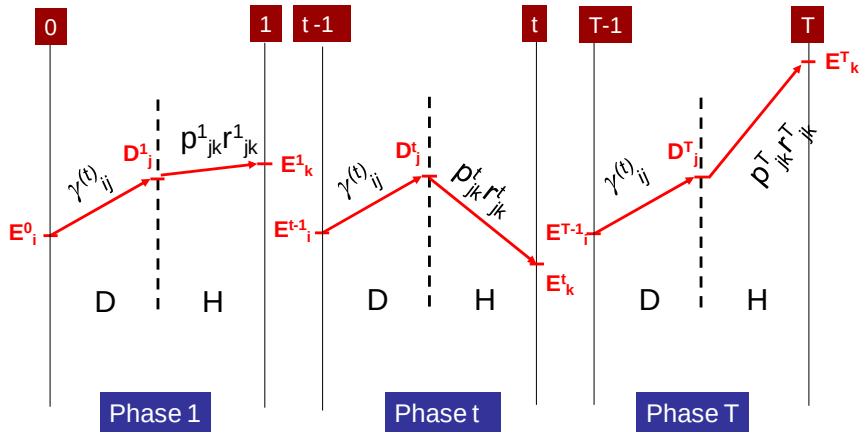
Le principe d'optimalité conduit, lui, à résoudre $(T-1)n$ problèmes de minimisation d'une fonction à 1 variable. Si l'on s'en tient là aussi à l'énumération, l'exploration de tous les $x_{i-1}$ pouvant mener à $x_i$ aboutit au calcul de $n$ quantités du type

$$v_{i-1}(x_{i-1}\ x_i) + V^*_{i-1}(x_{i-1})$$

Dans un cas, (principe d'optimalité) on a donc $n^2(T-1)$ sommes de $V_i$ à calculer. Dans l'autre, (énumération complète) on a $n^{T-1}$ sommes de $V_i$ à calculer. On voit en conséquence le gain que l'on peut obtenir en temps de calcul grâce au principe d'optimalité. On a notamment affaire à un algorithme polynomial.

### 10.6.2. Le cas aléatoire

Revenons au problème qui nous intéresse ici, c'est-à-dire à l'étude d'un phénomène se déroulant en plusieurs phases, elles-mêmes séparées en deux étapes : une étape décision et une étape hasard. Nous nous intéresserons ici au cas où l'étape décision précède l'étape hasard (DH). Encore une fois, l'étude des autres processus, c'est-à-dire les processus HD est tout à fait analogue. Dans ces conditions, on a le schéma suivant :




où rappelons-le :

- $E_1^{t-1} \dots E_i^{t-1} \dots E_{m(t)}^{t-1}$ sont des états possibles au début de la phase $t$.

- $D_1^t \dots D_j^t \dots D_{n(t)}^t$ sont des décisions possibles à la phase $t$.

- $\gamma_{ij}^t$ est le coût de la décision $D_j^t$ lorsque l'on est dans l'état $\mathrm{E_i^{t-1}}$.

- $P_{ik}^t$ est la probabilité de passer de l'état $D_j^t$ à l'état $\mathrm{E_k^{t+1}}$.

- $r_{jk}^t$ le coût de ce passage.

- $T$ le nombre total de phases.

Appelons par ailleurs $\Delta(E_i^t)$ l'ensemble des décisions que l'on peut prendre à partir de $E_i^t$. Rappelons qu'une stratégie est constituée de $T$ applications

$$E_i^{t-1} \rightarrow D_j^t \in \Delta(E_i^{t-1})$$

En des termes plus simples, une stratégie dit quelle décision il faut prendre lorsque l'on se trouve en un état donné d'une phase donnée.

Cela dit, si l'on veut choisir entre plusieurs stratégies, quel critère utiliser ? C'est habituellement, puisque nous sommes en univers aléatoire, l'espérance mathématique, ce qui suppose en général que le nombre de phases prises en compte est suffisamment grand (loi des grands nombres).

On appellera donc stratégie optimale, la stratégie qui minimisera l'espérance mathématique des coûts.

Comme dans le cas déterministe, où l'on parlait de sous-politique, définissons une sous-stratégie : une-sous stratégie de la phase $t$ à la phase $T$ est un ensemble de $T - t + 1$ applications.

$E_i^{\tau-1} \rightarrow D_j^\tau \; pour \; \tau = t, t+1 \dots T$

On peut également parler de sous-stratégie optimale : ce sont celles qui minimisent l'espérance mathématique de coût entre une date $t$ et la date finale $T$.

Le principe d'optimalité s'énonce alors de la façon suivante :

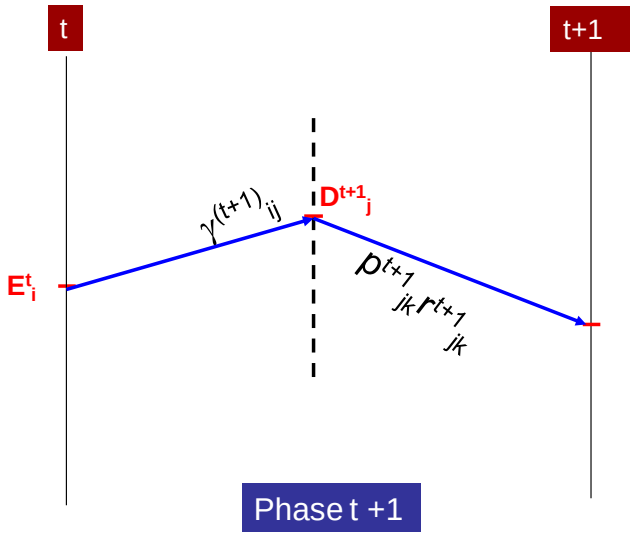
**Une sous-stratégie optimale de $t$ à $T$est formée d'une sous-stratégie optimale de $t+1$ à $T$.**

Ce principe, là encore va nous permettre d'écrire des équations importantes.

Supposons que nous connaissions la sous-stratégie optimale de la date $t+1$ à la date $T$. Cela veut dire que nous connaissons

$V_k^*(t+1) pour\ k = 1 \dots m(t+1)$

qui représente l'espérance mathématique la plus faible possible lorsque l'on se trouve en $E_k^{t+1}, k^{ième}$ état de la fin de la phase $t+1$, et $m(t)$ étant le nombre d'états atteignables en fin de période $t$.




Plaçons-nous maintenant en un état quelconque $E_i^t$, au début de la phase $t+1$ (ou fin de la phase $t$ ). Comment choisir $D_j^{t+1}$ à partir de $E_i^t$ ? En d'autres termes, puisque $i$ est quelconque, quelle est la sous-stratégie optimale de la date $t$ à la date finale, et quelles sont les espérances mathématiques $V_i^*(t)$ correspondantes ?

D'après la principe d'optimalité, il suffit de remarquer qu'en chaque état $E_k^{t+1}$est affectée d'une espérance de gain qu'il ne convient pas de remettre en cause,
c'est-à-dire $V_k^*(t+1)$.

Comme par ailleurs, les espérances mathématiques se composent, on a alors :

$$V_i^*(t) = Min\ [\gamma_{ij}^{t+1} + \sum_{k=1}^{m(t+1)} (r_{jk}^{t+1} + V_k^*(t+1)\ P_{jk}^{t+1})] \qquad (15)$$

$$j/D_j^{t+1} \in \Delta(E_i^t)$$

formule qui permet d'obtenir à la fois $D_j^t$ et l'espérance correspondante. L'algorithme de résolution est alors le suivant :

***Phase T***

$$V_i^*(t) = (T-1) = Min[\gamma_{ij}^T + \sum_{k=1}^{m(T)} P_{jk}^T r_{jk}^T]$$

$$j/D_j^T \in \Delta(E_i^{T-1})$$

***Phase T − 1***

$$V_i^*(T-2) = Min[\gamma_{ij}^{T-1} + \sum_{k=1}^{m(T-1)} P_{jk}^{T-1}(r_{jk}^{T-1} + V_k^*(T-1))]$$

$$j/D_j^{T-1} \in \Delta(E_i^{T-2})$$

***Phase t***

$$V_i^*(t-1) = Min\,[\gamma_{ij}^t + \sum_{k=1}^{m(t)} P_{jk}^t\ (r_{jk}^t + V_k^*(t))]$$

$$j/D_j^t \in \Delta(E_i^{t-1})$$

***Phase 1***

$$V_i^*(0) = Min\,[\gamma_{ij}^1 + \sum_{k=1}^{m(1)} P_{jk}^1\ (r_{jk}^1 + V_k^*(1))]$$
$$j/D_j^1 \in \Delta(E_i^0)$$

**Remarque très importante** : on est obligé, en programmation dynamique aléatoire, de *remonter le temps*. En effet, le terme d'espérance mathématique en un état donné, à un

instant donné, n'a de sens, que par rapport à ce qui va se passer après cet instant et à partir de l'état considéré. On ne peut calculer les espérances mathématiques à la date $t$ avant d'avoir celles de la date $t+1$.

Au contraire, en programmation dynamique déterministe, comme on pourra se l'assurer facilement, on peut décrire le temps dans les deux sens.

**Autre remarque** : nous avons ici une application très particulière du principe d'optimalité à la formalisation de problèmes aléatoires. Le champ d'utilisation de ce principe est beaucoup plus vaste. En particulier, nous traitons dans le cas présent des phénomènes qui sont caractérisés par des variables d'état discrètes $(E_i^t, D_j^t \dots)$ et une variable temps discrète (c'est pourquoi nous avons intitulé ce paragraphe « Programmation dynamique discrète »). Mais le principe d'optimalité peut s'appliquer également, et fructueusement, à des phénomènes à variables d'état et /ou variables temps continues. Dans ce dernier cas (variable temps continue), la programmation dynamique se trouve en voisinage étroit avec la théorie du contrôle optimal (voir en particulier tous les développements théoriques sur le principe de Pontryagin).
Comme les écritures qui viennent d'être proposées sont un tant soit peu rébarbatives, traitons un petit exemple.

### 10.6.3. Un exemple de programmation dynamique aléatoire

Considérons une unité de production produisant un bien rare, pour lequel la demande trimestrielle obéit à la loi de probabilité suivante :

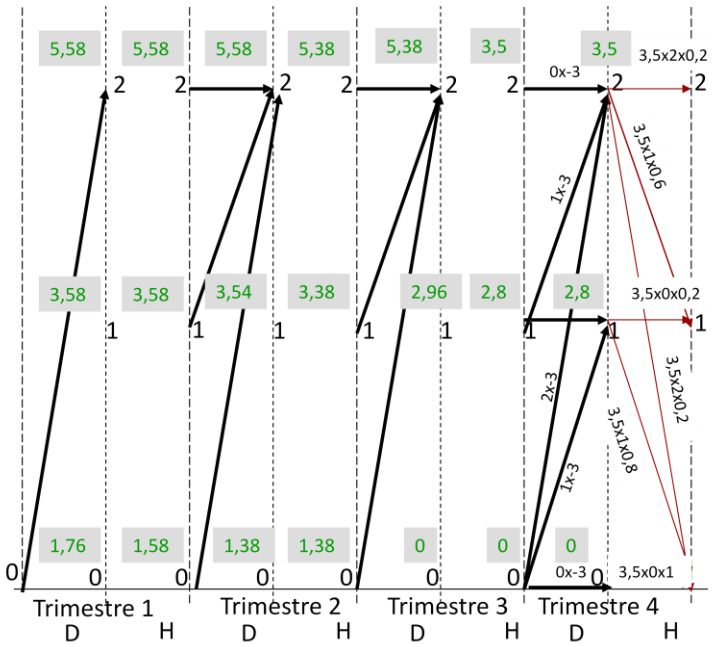
| d | 0 | 1 | 2 | > 2 |
|---|---|---|---|---|
| p(d) | 0,2 | 0,6 | 0,2 | 0 |

$d$ est la demande et $p(d)$ la probabilité correspondante.

Par ailleurs, sur une année considérée, les coûts de production unitaires et les prix de vente varient de la façon suivante, en fonction du trimestre :

| Trimestre | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Coût de production unitaire | 1,9 | 2 | 2 | 3 |
| Prix | 2 | 2,2 | 3 | 3,5 |

En début de chaque trimestre de l'année considérée, on se pose la question suivante: quelle quantité faut-il produire $0, 1, ou\ 2$ pour maximiser l'espérance de profit, sachant qu'en début du trimestre 1, le stock est nul ? On voit que ce problème peut être schématisé par le processus suivant, où l'ordonnée représente le nombre d'articles en stock.




Par souci de simplicité, on n'a indiqué les probabilités et valeurs de transition que pour le trimestre 4.

En reprenant les notations précédentes on a :

- Pour la phase hasard de la date 4, les gains et probabilités suivants :

$r_{22}^4 = 0$ $P_{22}^4 = 0{,}2$
(offre : 2 demande : 0)

$r_{21}^4 = 3{,}5$ $P_{21}^4 = 0{,}6$
(offre : 2 demande : 1)

$r_{20}^4 = 7$ $P_{20}^4 = 0{,}2$
(offre : 2 demande : 2)

$r_{11}^4 = 0$ $P_{11}^4 = 0{,}2$
(offre : 2 demande : 2)

$r_{10}^4 = 3{,}5$ $P_{10}^4 = 0{,}8$
(offre : 1 demande : 1 ou 2 vente :1)

$r_{00}^4 = 0 \qquad P_{00}^4 = 1$

(offre : 0 demande : 0, 1 ou 2 vente 0)

- Pour la phase décision, les gains suivants : $E_2^3 \rightarrow D_2^4 : 0$ (stock en début de période : 2, seule décision possible : ne rien produire).

$E_1^3 \rightarrow D_2^4 : -3$

(stock : 1 production : 1 coût : 3)

$E_1^3 \rightarrow D_1^4 : -0$

(stock :1 production : 0)

$E_0^3 \rightarrow D_2^4 : -6$

(stock :0 production : 2 coût : 6)

$E_0^3 \rightarrow D_1^4 : -3$

(stock :0 production : 1 coût : 3)

$E_0^3 \rightarrow D_0^4 : -0$

(stock : 0 production : 0)

Pour appliquer l'algorithme, issu du principe d'optimalité et exposé ci-dessus, on peut alors procéder de la façon suivante :

1) On calcule l'espérance mathématique de gain en chaque état

$$D_0^4, D_1^4, D_2^4 : (c'est la\ somme \sum_{k=1}^{m(T)} P_{ij}^T r_{jk}^T\ dans\ la\ théorie)$$

Soit :

en $D_0^4$ : espérance mathématique $= 0$

en $D_1^4$ : $0{,}2 \times 0 + 0{,}8 \times 3{,}5 = 2{,}8$

en $D_2^4$ : $0{,}2 \times 0 + 0{,}6 \times 3{,}5 + 0{,}2 \times 7 = 3{,}5$

On inscrit ces chiffres en les encadrant sur la figure.

2) On se place en chaque état $E_0^3, E_1^3, E_2^3$ et on calcule l'espérance mathématique de chaque transition possible $E_i^3 \rightarrow D_j^4$. Pour un $E_i^3$ donné, on choisit la transition qui maximise le gain; on retient la valeur de l'espérance mathématique en l'encadrant sur le dessin.

$E_0^3 \rightarrow D_0^4\ espérance : 0 + 0 = 0 \quad \leftarrow$

$E_0^3 \rightarrow D_1^4\ espérance : 2{,}8 - 3 = -0{,}2$

$E_0^3 \rightarrow D_2^4\ espérance : 3{,}5 - 6 = -2{,}5$

(meilleur choix : $E_0^3 \rightarrow D_0^4$ ; on signale ce choix en renforçant la flèche correspondante sur la figure)

$E_1^3 \rightarrow D_1^4$ $espérance : 2{,}8 - 0 = 2{,}8 \leftarrow$

$E_1^3 \rightarrow D_2^4$ $espérance : 3{,}5 - 3 = 0{,}5$

(meilleur choix $E_1^3 \rightarrow D_1^4$ : on le marque sur la figure)

$E_2^3 \rightarrow D_2^4$ $espérance : 3{,}5 - 0 = 3{,}5$ $(seul\ choix\ possible)$

Les calculs pour les autres phases reprennent alors ceux que l'on vient de faire pour la phase 4. C'est pourquoi nous les donnons ci-dessous sans commentaires.

***Phase 3 Hasard :***

$D_0^3 : espérance : 0$

$D_1^3 : espérance : 0{,}2 \times 2{,}8 + 0{,}8(0 + 3) = 2{,}96$

$D_2^3 : espérance : 0{,}2 \times 3{,}5 + 0{,}6(2{,}8 + 3) + 0{,}2(0 + 6) = 5{,}38$

***Décision :***

$E_0^2 \rightarrow D_0^3 : epérance : 0$

$E_0^2 \rightarrow D_1^3 : epérance : 2{,}96 - 2 = 0{,}96$

$E_0^2 \rightarrow D_2^3 : epérance : 5{,}38 - 4 = 1{,}38 \leftarrow$

$E_1^2 \rightarrow D_1^3 : epérance : 2{,}96 - 0 = 2{,}96$

$E_1^2 \rightarrow D_2^3 : epérance : 5{,}38 - 2 = 3{,}38 \leftarrow$

$E_2^2 \rightarrow D_2^3 : epérance : 5{,}38 - 0 = 5{,}38 \leftarrow$

***Phase 2 Hasard :***

$D_0^2$ $espérance : 1{,}38$

$D_1^2$ $espérance : 0{,}2 \times 3{,}38 + 0{,}8(1{,}38 + 2{,}2) = 3{,}54$

$D_2^2$ $espérance : 0{,}2 \times 5{,}38 + 0{,}6(3{,}38 + 2{,}2) + 0{,}2(1{,}38 + 4{,}4) = 5{,}51$

***Décision :***

$E_0^1 \rightarrow D_0^2 : epérance : 1{,}38 - 0 = 1{,}38$

$E_0^1 \rightarrow D_1^2 : epérance : 3{,}54 - 2 = 1{,}54$

$E_0^1 \rightarrow D_2^2 : epérance : 5{,}51 - 4 = 1{,}51 \leftarrow$

$E_1^1 \rightarrow D_1^2 : epérance : 3{,}54 - 0 = 3{,}54$

$E_1^1 \rightarrow D_2^2 : epérance : 5{,}51 - 2 = 3{,}51 \leftarrow$

$E_2^1 \rightarrow D_2^2 : epérance : 5{,}51 - 0 = 5{,}51 \leftarrow$

***Phase 1 Hasard***

$D_0^1\ espérance : 1{,}51$

$D_1^1\ espérance : 0{,}2 \times 3{,}51 + 0{,}8(1{,}51 + 2) = 3{,}51$

$D_2^1\ espérance : 0{,}2 \times 5{,}51 + 0{,}6(3{,}51 + 2) + 0{,}2(1{,}51 + 4) = 5{,}51$

***Décision***

(état initial ; $E_0^0$)

$E_0^0 \rightarrow D_0^1 : epérance : 1{,}51 - 0 = 1{,}51$

$E_0^0 \rightarrow D_1^1 : epérance : 3{,}51 - 1{,}9 = 1{,}61$

$E_0^0 \rightarrow D_2^1 : epérance : 5{,}51 - 3{,}8 = 1{,}71 \leftarrow$

On voit alors que la stratégie optimale pour cet exemple peut se résumer ainsi : produire au maximum (c'est-à-dire 2 unités) pour les trimestres 1, 2 et 3. Pour le trimestre 4, ne pas produire.

### 10.6.4. Programmation dynamique et chaîne de Markov

Supposons que l'on soit en présence d'un phénomène relevant des modèles exposés au paragraphe précédent, c'est-à-dire de la programmation dynamique discrète et aléatoire.

Choisissons une stratégie quelconque : cela signifie qu'à chaque $E_i^{t-1}$ on fait correspondre un $D_j^t$ et un seul. À partir de $D_j^t$ le système évolue aléatoirement vers les états $E_k^t$ avec les probabilités $P_{jk}^t$. Mais on peut dire aussi, comme on décide de passer obligatoirement de $E_i^{t-1}$ à $D_j^t$, qu'à partir de $E_i^{t-1}$ le système évolue vers les états $E_k^t$ avec les probabilités $P_{jk}^t$.

On voit alors que le choix d'une stratégie permet alors de ramener l'étude du système à un modèle du type de ceux étudiés dans le paragraphe 10.4, c'est-à-dire les chaînes de Markov avec valeurs de transition (sans que l'hypothèse d'homogénéité soit faite, dans la plupart des cas).

En particulier, si l'on se donne une stratégie, on peut calculer les espérances mathématiques en chacun des états $E_i^t$ grâce aux équations de récurrence (10). On peut ainsi comparer différentes stratégies.

Mais, là aussi, on voit de quelle aide peut être le principe d'optimalité : le nombre de stratégies possibles devient tout de suite très grand dès que le nombre de phases et d'états dépasse des valeurs très modestes.

Un cas particulièrement important, en théorie, de la programmation dynamique discrète est celui où il y a homogénéité, à la fois des états, des probabilités et des revenus :

$$E_i^t = E_i \qquad D_k^t = D_k \ \forall t$$

$$\gamma_{ij}^t = \gamma_{ij} \qquad \forall t$$

$$P_{jk}^t = P_{jk} \qquad \forall t$$

$$r_{jk}^t = r_{jk} \qquad \forall t$$

Comme aux paragraphes précédents, il est alors usuel de se placer en régime permanent c'est-à-dire de faire tendre $t$ vers l'infini.
On démontre alors qu'il y a une stratégie permanente, c'est-à-dire qu'à chaque phase $t$, il convient de prendre les mêmes décisions :

$$E_i \rightarrow D_j$$

Des méthodes permettent de trouver ces stratégies permanentes. Elles s'appuient sur les résultats du paragraphe 10.4. L'exposé de ces points dépasse le cadre limité de cet ouvrage. Nous nous contenterons ici de souligner la caractère extraordinairement limitatif des hypothèses nécessaires pour formaliser un problème de gestion par un processus DH ou HD homogène dans le temps.

# Chapitre 11 • Phénomènes d'attente

## INTRODUCTION

Une file d'attente est en général caractérisée par le fait que ni les arrivées des « clients » ni les temps de « service » ne sont des données connues, mais constituent au contraire des variables aléatoires.

On comprend alors que l'étude de ces phénomènes ait pu constituer un chapitre important des processus stochastiques appliqués à la Recherche Opérationnelle.

D'une façon générale, une file d'attente est caractérisée par plusieurs éléments qui constituent le système :

- les clients, qui arrivent de façon irrégulière d'une origine que nous appellerons « source ». Il peut s'agir de personnes (clients aux guichets de banque, par exemple) ou d'objets (machine en réparation dans un atelier, autre exemple).
- le centre d'attente, où les clients stationnent avant d'être servis.
- le centre de service, constitué par une ou plusieurs stations de services, où le temps de service est en général aléatoire.

Le système étudié peut être plus ou moins complexe : par exemple, la source peut être finie, dans la mesure où le client servi est replacé dans la source (exemple des machines tombant en panne) ou infinie (exemple : clients dans un magasin). Dans le premier cas, le système est dit *fermé*, dans le second cas, il sera *ouvert*. Par ailleurs, il peut y avoir plusieurs stations de service en parallèle ou en série, ou encore un mélange de ces deux configurations (réseaux d'attente). Bien sûr, les lois d'arrivée des clients et les lois des temps de service pourront être plus ou moins compliquées. Enfin, l'attitude des clients et des serveurs dans le système peut rendre difficile la formalisation du phénomène : il peut y avoir impatience des clients, qui quittent alors le système lorsque leur temps d'attente leur apparaît prohibitif, ou encore influence de l'engorgement du système sur les performances des serveurs.

On conçoit alors, devant toutes ces éventualités, la multiplicité des configurations possibles. Nous nous contenterons d'insister sur un cas très particulier et très simple : celui d'un système ouvert à une station, où les arrivées et les temps de service entrent dans le cadre d'un processus stochastique bien connu : le processus poissonnien. Nous ne ferons que donner des aperçus sur les autres systèmes.

## 11.1. SYSTEME OUVERT A UNE SEULE STATION. ARRIVEES POISSONNIENNES ET SERVICE EXPONENTIEL

Dans ce paragraphe, nous nous intéresserons au phénomène suivant : des clients arrivent à une station de service où ils sont servis dans l'ordre où ils arrivent. Ils attendent éventuellement, c'est-à-dire lorsque la station est déjà occupée quand ils arrivent. Lorsqu'ils sont servis, ils disparaissent du système (système ouvert : il n'y a pas recyclage des clients). Par ailleurs, nous supposerons que :

- les arrivées des clients sont poissonniennes,
- le service est exponentiel.

Cette hypothèse mérite un rappel rapide sur les processus poissonniens.

### 11.1.1. Rappels sur les processus poissonniens

Dire que les arrivées des clients se font de façon poissonnienne, c'est faire les hypothèses suivantes :

*H 1. Indépendance*

Si $N_1, N_{2,} \dots N_k$ sont les nombres (aléatoires) de clients arrivés respectivement sur les intervalles de temps disjoints $(t_1, t'_1)$, $(t_2, t'_2) \dots (t_k, t'_k)$, alors $N_1, N_{2,} \dots N_k$ sont indépendants en probabilité quel que soit $k$, et quels que soient les intervalles de temps choisis. Dans la mesure où $N_i$ ne dépend pas du nombre d'arrivées survenu pendant les intervalles de temps précédents, on dit aussi que le processus est sans mémoire.

*H. 2. Stationnarité.*

La loi de l'ensemble $N_1, N_{2,} \dots N_k$ reste inchangée si les $2k$ valeurs $t_1,\ t'_1, t_2,\ t'_2 \dots\ t_k,\ t'_k$ sont augmentés ou diminués d'une même quantité.

Ces deux hypothèses conduisent au fait suivant : si l'on prend un intervalle de temps quelconque $(t, t')$ alors la loi de probabilité de ce nombre ne dépend que de la quantité $t' - t$.

A ces deux hypothèses essentielles, il convient d'en rajouter deux autres, pour exclure des processus aberrants :

*H 3.*

Si $p_0(\Delta t)$ est la probabilité qu'il n'arrive aucun client sur l'intervalle de temps $\Delta t$, on a

$0 < p_0(\Delta t) < 1$ pour tout $\Delta t > 0$ ($si\ p_0(\Delta t) = 1$, il n'arrive jamais aucun client ; ($si\ p_0(\Delta t) = 0$, il arrive toujours un client; ce sont des cas inintéressants).

*H 4.*

La probabilité qu'il arrive deux clients au moins simultanément est nulle.
Sous ces hypothèses, on démontre (voir cours de probabilité) les résultats suivants :

### 11. 1. 2. Loi du nombre d'arrivées sur un intervalle de temps $\Delta t$

Si $N(\Delta t)$ est le nombre d'arrivée de clients sur l'intervalle de temps $\Delta t > 0$, on a :

$$P[N(\Delta t) = n] = e^{-\lambda \Delta t} \frac{(\lambda \Delta t)^n}{n!}$$

$\lambda$ *étant une constante* $> 0$

*11.1.2.1. Probabilité d'une arrivée sur un intervalle de temps* $\boldsymbol{dt}$ *infiniment petit*

La probabilité qu'il y ait une arrivée entre $t$ *et* $t = dt$ est égale à $\lambda dt$. Cette propriété met en valeur la signification du paramètre $\lambda$ qui sera appelé taux d'arrivée.

*11.1.2.2. Espérance mathématique et variance du nombre d'arrivées*

On a : $E[N(\Delta t)] = \lambda \Delta T$
et $V[N(\Delta t)] = \lambda \Delta T$

$E$ *et* $V$ étant respectivement les opérateurs espérance mathématique et variance.

*11.1.2.3. Loi de l'intervalle de temps séparant deux arrivées successives.*

Soit $U$ la variable aléatoire constituée par l'intervalle de temps séparant deux arrivées successives. Il s'agit d'une variable aléatoire continue de fonction de densité $f(u)$.

$$f(u) = \lambda e^{-\lambda u}$$

On trouve, la loi exponentielle, qui est donc liée étroitement au processus poissonnien.

Si nous prenons pour l'étude de notre file d'attente un service exponentiel, cela signifie que le temps de service $S$ d'un client quelconque a pour loi de densité une exponentielle $\mu e^{-\mu s}$, $\mu$ étant appelé le taux de service. Dans ces conditions la probabilité pour que, à un instant $t$ quelconque, un client en service à $t$ soit effectivement servi entre $t$ et $t + dt$ (et donc sorte du système) est égale à $\mu dt$. Comme le résultat 11.1.1.2, cette propriété est fondamentale et simplifie de beaucoup l'étude du phénomène en cause.

*11.1.2.4. Espérance mathématique et variance du temps de service.*

Si le temps de service $S$ d'un client quelconque est exponentiel, avec un taux de service $\mu$, on a :

$$E(S) = \frac{1}{\mu} \qquad V(S) = \frac{1}{\mu^2}$$

Ces quelques rappels étant effectués, passons à l'étude de la file d'attente considérée, c'est-à-dire à une station en système ouvert.

### 11.1.3. Etude de la file d'attente. Equation d'état

Supposons donc que les arrivées soient poissonniennes, de taux $\lambda$ et les services exponentiels, de taux $\mu$, identiques pour chaque client.


Taux d'arrivée moyen $\lambda$ clients/unité de temps

Taux de service moyen $\mu$ clients/unité de temps

À partir de maintenant, nous désignerons par $N(t)$ (variable aléatoire) le nombre total de clients dans le système à l'instant $t$, c'est-à-dire la somme du nombre de clients en attente d'être servis et du client qu'on est en train de servir à l'instant $t$ (s'il existe, car la station peut fort bien être inoccupée). Nous appellerons $p_n(t)$ la probabilité que $N(t) = n$.

$$P_n(t) = P[N(t) = n]$$

Par ailleurs, nous dirons que, si à l'instant $t$, il y a $n$ clients en tout dans le système, ce dernier est dans l'état $E_n$.

Dans ces conditions, considérons deux instants $t \; et \; t + dt$, $dt$ étant l'intervalle de temps infiniment petit. D'après les lois choisies pour les arrivées et les services, la probabilité pour qu'il arrive plus d'un client entre $t \; et \; t + dt$, est négligeable, ainsi que la probabilité pour qu'il y ait plus d'un client servi sur le même intervalle de temps. Alors, entre $t \; et \; t + dt$ les seules transitions possibles sont les suivantes, si $E_n$ est l'état du système à l'instant $t + dt$, et si $n$ est positif :

$E_n \rightarrow E_n$ (1 client arrivé et 1 client sorti ou 0 client arrivé et 0 client sorti)

$E_{n+1} \rightarrow E_n$ (0 client arrivé et 1 client sorti)

$E_{n-1} \rightarrow E_n$ (1 client arrivé et 0 client sorti)

Si $n$ est nul, les seules transitions possibles sont :

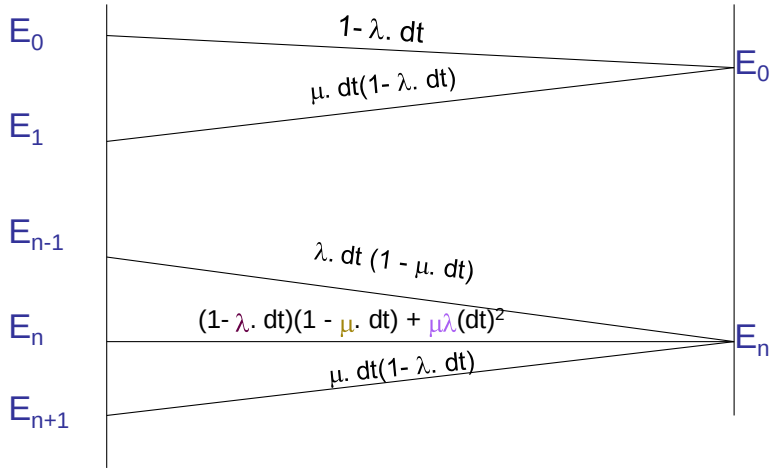
$E_0 \rightarrow E_0$ (0 client arrivé)

$E_1 \rightarrow E_0$ (0 client arrivé et 1 client sorti)

Or, dans le cas qui nous intéresse, c'est-à-dire les arrivées poissonniennes et le service exponentiel, les probabilités de transition entre les états $E_i$ de l'instant $t$ à l'instant $t + dt$ sont particulièrement faciles à déterminer.

En effet, le probabilité qu'il arrive un client entre $t\ et\ t + dt$ est égale à $\lambda dt$, donc la probabilité qu'il n'y ait pas d'arrivée pendant le même laps de temps est $1 - \lambda dt$. De la même façon, si à $t$ un client est en train d'être servi, la probabilité que ce service soit terminé entre $t\ et\ t + dt$ est égale à $\mu dt$ et la probabilité qu'il n'en soit pas ainsi est $1 - \mu dt$.

En conséquence, on peut écrire le graphe de transition suivant :




Si l'on néglige les termes du second ordre, c'est-à-dire les termes en $(dt)^2$ on obtient les équations suivantes :

Pour
$n > 0$ $$p_n(t+dt) = \lambda\ p_{n-1}(t)\ dt + (1-\lambda\ dt - \mu\ dt)\ p_n(t) + \mu\ p_{n+1}(t)\ dt$$

Pour
$n = 0$ $$p_0(t+dt) = (1-\lambda\ dt)\ p_0(t) + \mu\ dt\ p_1(t)$$

Ce qui peut encore s'écrire :

Pour

$n > 0$ $$\frac{p_n(t+dt) - p_n(t)}{dt} = \lambda\ p_{n-1}(t) - (\lambda + \mu)\ p_n(t) + \mu\ p_{n+1}(t)$$

Pour

$$n = 0 \qquad \frac{p_0(t+dt) - p_0(t)}{dt} = -\lambda\ p_0(t) + \mu\ p_1(t)$$

Si l'on fait tendre $dt$ vers 0, on obtient de nouvelles équations (à condition, bien sûr, que certaines conditions de continuité et de dérivabilité soient vérifiées, ce que nous supposerons : encore une fois, il ne s'agit pas de faire des mathématiques). Ces équations, dites équations d'état du système, sont :

$$\begin{aligned} n > 0 \qquad & p_n(t) = \lambda\ p_{n-1}(t) - (\lambda + \mu)\ p_n(t) + \mu\ p_{n+1}(t) \\ n = 0 \qquad & p_0'(t) = -\lambda\ p_0(t) + \mu\ p_1(t) \end{aligned} \qquad (1)$$

Ces équations permettent théoriquement de calculer les fonctions $p_n(t)$, si l'on adjoint les conditions :

$$p_n(0) = 0 \quad \forall n > 0^{13} \qquad p_0(0) = 1$$

Mais ce calcul, faisant intervenir les transformées de Laplace, est assez complexe. Par ailleurs, comme dans le cas des chaînes de Markov, il est rarement intéressant de connaître les probabilités d'état pour chaque instant $t$. Il importe d'avantage de connaître le *régime permanent* s'il existe, c'est-à-dire la limite des probabilités $p_n(t)$ lorsque $t$ tend vers l'infini. Si ces probabilités tendent effectivement vers des limites $p_n^*$ et si par ailleurs ces limites sont indépendantes de l'état initial, on dira là encore, comme dans le cas des chaînes de Markov que le phénomène est *ergodique.*

À l'aide des équations (1), on démontre bien que $p_n(t)$ tend vers une limite lorsque $t$ tend vers l'infini. Comment calculer cette limite?

Il suffit pour ce faire de remarquer que si $P_n(t) \to p_n^*$ lorsque $t \to \infty$, $p_n(t) \to 0$ dans les mêmes conditions.

Les équations (1) donnent alors immédiatement les équations (2), équations à la limite :

$$\begin{aligned} & -\lambda\ p_0^* + \mu\ p_1^* = 0 \\ & \lambda\ p_{n-1}^* - (\lambda + \mu)\ p_n^* + \mu\ p_{n+1}^* = 0 \end{aligned}$$

On en tire :

$$p_1^* = \frac{\lambda}{\mu} p_0^*$$

---

13 On pourrait ainsi imposer la condition :

$$p_{n_0}(0) = 1 \text{ et } p_n(0) = 0 \text{ pour } n \neq n_o.$$

$$p_2^* = \left(\frac{\lambda}{\mu}\right)^2 p_0^*$$

$$p_3^* = \left(\frac{\lambda}{\mu}\right)^3 p_0^*$$

et par récurrence :

$$p_n^* = \left(\frac{\lambda}{\mu}\right)^n p_0^*$$

Pour avoir complètement $p_n^*$, il suffit de remarquer que :

$\sum_{i=0}^{\infty} p_i^* = 1$, ce qui donne :

$$p_0 \sum_{i=0}^{\infty} \left(\frac{\lambda^i}{\mu}\right) = 1$$

Si $\frac{\lambda}{\mu} < 1$, la série qui figure au premier membre de cette équation converge vers

$$\frac{1}{1-\frac{\lambda}{\mu}} \; soit \; vers \; \frac{1}{1-\tau} \; si \; l'on \; pose \; \tau = \frac{\lambda}{\mu}$$

Dans ces conditions, on a :

$$p_0^* = 1 - \tau$$

et

$$p_n^* = \tau^n (1-\tau) \qquad (3)$$

On voit donc que grâce à l'élaboration des équations d'états (1), on connaît complètement le *régime permanent* (ou l'état limite) du système, puisque l'on connaît les probabilités des états $E_i$. Ces probabilités, par ailleurs, ne dépendent pas de l'état initial. Il faut noter que ce résultat n'est valable que si $\tau < 1$.

Que se passe-t-il lorsque $\tau \geq 1$ ? On démontre que dans ce cas, la file d'attente s'allonge indéfiniment presque sûrement. Ce résultat est bien normal puisque λ mesure le nombre moyen d'arrivées par unité de temps et μ le nombre moyen de services par unités de temps. Si μ est supérieur à λ, il est logique que le système s'engorge et ne puisse résorber la file d'attente.

$$\tau = \frac{\lambda}{\mu} \; sera \; appelé \; intensité \; de \; trafic$$

### 11.1.4. Calcul de différentes grandeurs

Les résultats précédents vont nous permettre de calculer quelques grandeurs caractéristiques du régime permanent du système étudié.

#### *11.1.4.1. Nombre moyen d'unités dans le système*

Soit $n$ le nombre moyen d'unités dans le système. C'est l'espérance mathématique de la variable aléatoire $N$, dont on connaît la distribution résumée par les probabilités $p_n^*$. On a donc :

$$\bar{n} = \sum_{n=0}^{\infty} n\, p_n^*$$

$$\bar{n} = \sum_{n=0}^{\infty} (1-\tau)\, \tau^n\, n$$

$$= (1-\tau) \sum_{n=0}^{\infty} n\, \tau^n = (1-\tau)\, \tau \sum_{n=0}^{\infty} n\, \tau^{n-1}$$

D'où

$$\overline{n} = (1-\tau)\, \tau\, d\left[\frac{1}{1-\tau}\right]$$

$$\overline{n} = \frac{\tau}{1-\tau} \qquad (4)$$

#### *11.1.4.2. Nombre moyen d'unités en attente*

Soit $\overline{v}$ le nombre moyen de clients en attente.

$\overline{v}$ n'est pas identique à $\overline{n}$. En effet si $v$ est le nombre de clients en attente à un instant donné, on a :

$v = 0 \quad si\ n = 0 \quad et\ v = n - 1 \quad si\ n > 0$

Donc :

$$\overline{v} = \sum_{n=1}^{\infty} (n-1) p_n^*$$

Soit $v = n - (1 - p_0)$

$$= \frac{\tau}{1-\tau} - 1 + 1 - \tau$$

Soit

$$\overline{v} = \frac{\tau^2}{1-\tau} \qquad (5)$$

### *11.1.4.3. Temps d'attente total moyen*

Le temps d'attente total d'un client est le temps qu'il passe dans le système (temps de service compris).

La méthode que nous avons suivie jusqu'ici ne permet pas de trouver facilement ce temps moyen. Cependant, on peut ici effectuer un raisonnement en « moyenne ».

En effet, le nombre moyen de sorties du système pendant une unité de temps est égal à $\mu(1-p_0^*)$ ou encore à $\lambda$, puisqu'en régime permanent, il est indispensable que « nombre moyen d'entrée = nombre moyen de sorties ».

En conséquence, le temps d'attente moyen $t_a$ dans le système est :

$$\overline{t_a} = \frac{\overline{n}}{\mu(1-p_0)} = \frac{\overline{n}}{\lambda}$$

$$\overline{t_a} = \frac{1}{\lambda}\ \frac{\tau}{1-\tau} \qquad (6)$$

### *11.1.4.4. Temps d'attente moyen dans la file*

Le temps d'attente moyen dans la file $t_f$ est de la même façon :

$$\overline{t_f} = \frac{\overline{v}}{\lambda} = \frac{\overline{v}}{\mu(1-p_0)}$$

$$\overline{t_f} = \frac{1}{\lambda}\ \frac{\tau^2}{1-\tau} \qquad (7)$$

La méthode que nous venons d'utiliser est appelée *« méthode différentielle »* par opposition à la méthode intégrale, que nous verrons ci-dessous.

Pour le moment, la méthode différentielle nous a permis de décrire convenablement un système extrêmement simple, constitué d'une seule station, avec arrivées poissonniennes et service exponentiel. Mais ce service ne permet pas de choix économique (sauf peut-être ceux qui portent sur le taux de service). C'est pourquoi, nous allons examiner maintenant l'extension de la méthode à un système de plusieurs stations, en supposant toujours les arrivées poissonniennes et le service exponentiel.

## 11.2. SYSTEME OUVERT A PLUSIEURS STATIONS. ARRIVEES POISSONNIENNES ET SERVICE EXPONENTIEL

Dans le cas de plusieurs stations, un problème théorique se pose dans la mesure où l'évolution du système dépend *a priori* des choix effectués par les clients et leur file d'attente.

Nous supposerons ici que chacune des stations est caractérisée par un service exponentiel de taux identique.

Dans ce cas, c'est-à-dire arrivées poissonniennes et service exponentiel de taux égal pour toutes les stations, on verra que ce problème ne se pose pas si l'on utilise la méthode différentielle.

### 11.2.1. Equation d'état du système

De la même façon que précédemment, considérons la variable aléatoire $N(t)$, représentant le nombre de clients dans le système à l'instant $t$.

$p_n(t) = p[N(t) = n]$

Si $N(t) = n$ à l'instant $t$, nous dirons que le système est dans l'état $E_n$.

Nous désignerons par $s$ le nombre de stations. A un instant $t$ quelconque, nous pouvons avoir les deux situations suivantes :

- $n < s$ Les $n$ clients dans le système sont en train de se faire servir ; il y a $(s - n)\ stations$ inoccupées. On peut alors considérer que, chaque client étant servi suivant un service exponentiel de taux $\mu$, la superposition de ces $n$ services conduit à un service global exponentiel de taux $n\mu$ c'est-à-dire que la probabilité qu'un client sorte du système entre $t\ et\ t + dt$ est égale à $n\mu dt$. Ce résultat provient du fait que la somme de variables poissonniennes indépendantes est une variable poissonnienne, dont le taux est égal à la somme des taux des variables initiales.
- $n \geq s$ . Il y a $s$ clients en instance de sortie du système. Le résultat est analogue à celui dégagé pour le cas précédent. Mais ici, le taux global de service est $s\mu$, c'est-à-dire que la probabilité pour qu'un client sorte entre $t\ et\ t + dt$ est égale à $s\mu dt$.

Dans ces conditions, comme pour le cas à une station, les seules transitions envisageables entre $t\ et\ t + dt,\ si\ dt$ est infiniment petit sont :

$E_n \rightarrow E_n$

$E_n \rightarrow E_{n+1}$

$E_n \rightarrow E_{n-1}$

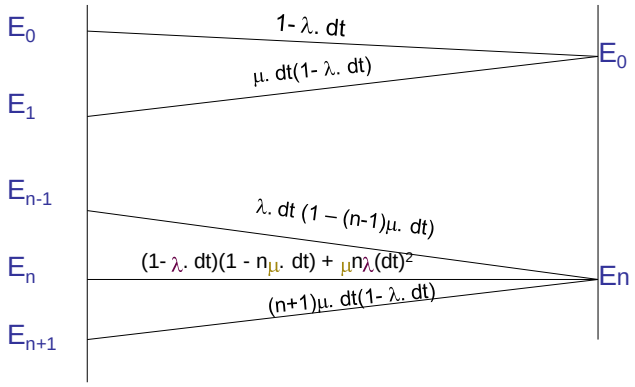
$pour\ n > 0, et$

$E_0 \rightarrow E_0$

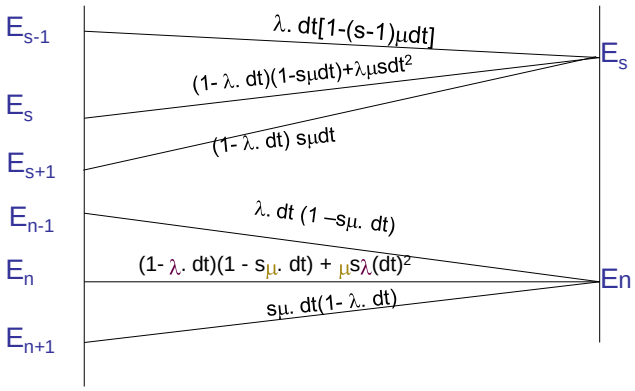
$E_0 \rightarrow E_1$

$pour\ n = 0$

Plus précisément, on a le graphe de transition suivant, plus compliqué que dans le cas d'une station, pour la raison que l'on est obligé de distinguer le cas $n < s$ et le cas $n \geq s$.




**n < s**




**n > s**

On a donc les équations suivantes, si l'on néglige les termes en $d(t)^2$:

$p_0(t + dt) = (1 - \lambda dt)p_0(t) + \mu dt p_1(t)$

$0 < n < s \; p_n(t+dt) = \lambda dt p_{n-1}(t) + (1 - \lambda dt - n\mu dt))p_n(t) + (n+1)\mu dt p_{n+1}(t)$

$n \geq s \quad p_n(t+dt) = \lambda dt p_{n-1}(t) + (1 - \lambda dt - s\mu dt))p_n(t) + s\mu dt p_{n+1}(t)$

D'où les équations d'états, obtenues en faisant tendre $dt$ vers $0$ :

$p_0(t) = -\lambda p_0(t) + \mu p_1(t)$

$1 \leq n < s \qquad p_n(t) = \lambda p_{n-1}(t) - (\lambda + n\mu)p_n(t) + (n+1)\mu p_{n+1}(t$ (8)

$n \geq s \qquad p_n(t) = \lambda p_{n-1}(t) - (\lambda + s\mu)p_n(t) + s\mu p_{n+1}(t$

Si l'on se situe en régime permanent, dont on démontre qu'il existe à condition que la quantité $\tau' = \frac{\lambda}{s\mu}$ soit inférieure à 1, on a les équations suivantes,

si $p_n(t) \to p_n^*$ lorsque $t \to +\infty$

$-\lambda p_0^* + \mu p_1^* = 0$

$1 < n < s \quad \lambda p_{n-1}^* - (\lambda + n\mu)p_n^* + (n+1)\mu p_{n+1}^* = 0$

$n \geq s \quad \lambda p_{n-1}^* - (\lambda + s\mu)p_n^* + s\mu p_{n+1}^* = 0$

On en tire les résultats suivants :

$$p_1^* = \frac{\lambda}{\mu} p_0^* = \tau \; p_0^*$$

$$p_2^* = p_0^* \left(\frac{\lambda}{\mu}\right)^2 \times \frac{1}{2!} = p_0^* \frac{\tau^2}{2}$$

Par récurrence :

$Pour \;\; 1 < n \leq s \qquad p_n^* = p_0^* \frac{\tau^n}{n!}$

$Pour \; n > s \qquad p_n^* = p_0^* \frac{\tau^n}{s^{n-s} s!}$ (10)

avec $\tau$ (taux de service) $= \frac{\lambda}{\mu}$

Pour calculer $p_0^*$, il suffit de remarquer que :

$$\sum_{n=0}^{\infty} p_n^* = 1$$

on obtient alors l'expression suivante :

$$p_0^* = \frac{1}{\frac{\tau^s}{s!\left(1-\frac{\tau}{s}\right)} + \sum_{n=0}^{s-1} \frac{\tau^n}{n!}} \qquad (11)$$

Cette formule permet d'obtenir les $p_0^*$. On peut alors calculer, avec des procédés analogues à ceux que nous avons utilisés pour le cas d'une station, un certain nombre de grandeurs caractéristiques telles que :

$\overline{n} = nombre\ moyen\ de\ clients\ dans\ le$ système = $\sum_{n=0}^{\infty} np_n$

$\overline{v}$ = nombre moyen de clients dans la file d'attente $= \sum_{n=s+1}^{\infty}(n-s)p_n$

$\overline{t_a}$ = temps d'attente moyen total dans le système

$\overline{t_f}$ = temps d'attente moyen dans la file

Le calcul de ces grandeurs ne présente pas de difficultés particulières. On aboutit à des formules plus ou moins compliquées qui n'ont pas tellement d'intérêt en elles-mêmes. Nous renvoyons aux ouvrages spécialisés le lecteur soucieux de les connaître (par exemple Kaufmann-Cruon).

Il est utile de souligner que ces calculs n'ont de sens que si $\frac{\lambda}{\mu s} < 1$.
(intensité de trafic <1).

### 11.2.2. Choix économique du nombre de stations

Si $\lambda$ et $\mu$ sont connus, une variable peut faire l'objet d'un choix : le nombre de stations $s$ ; ce nombre doit de toute façon obéir à l'inégalité :

$$\frac{\lambda}{\mu s} < 1\ soit\ s > \frac{\lambda}{\mu}$$

Sinon, la file d'attente devient infinie. Cela dit, comment choisir $s > \frac{\lambda}{\mu}$ ?

Deux facteurs contradictoires interviennent en général :

- d'une part, le coût d'exploitation du système qui croît avec le nombre de stations
- d'autre part, les temps d'attente des unités dans le système, qui décroissent avec le nombre de stations.

On utilise très souvent les hypothèses suivantes :

- le coût d'exploitation par unité de temps en système est une fonction linéaire du nombre de stations, égale à $C_1 s$.

- on prend en charge les temps d'attente des clients par l'intermédiaire d'une « valeur du temps d'attente » : une unité de temps perdue par un client quelconque coûte $C_2$.

Dans ces conditions, sur un intervalle de temps $T$ moyen, la fonction à minimiser est :

$$r(s) = c_1\, sT + c_2\, \lambda\, \overline{t_a}\; T \qquad (12)$$

Il est équivalent de minimiser le coût par unité de temps :
$c_1 s + c_2\, \lambda\, t_a$

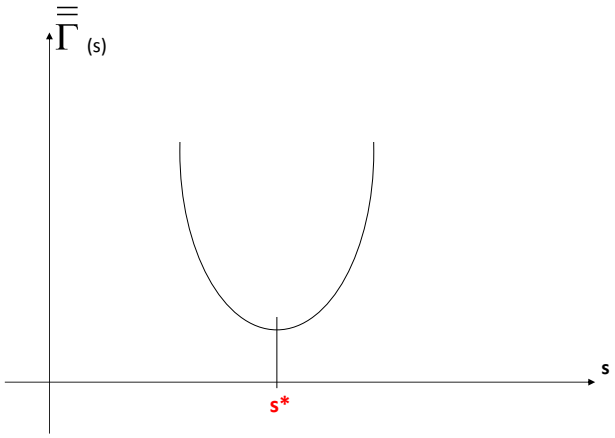
Si $\overline{n}$ est le nombre moyen d'unités dans le système, cette quantité est aussi égale à :

$\overline{\overline{\Gamma}}$ (s)=$c_{1s} + c_2\overline{n}$ (13)

Dans ce cas précis (arrivées poissonniennes et service exponentiel), $\overline{\overline{\Gamma}}$ est une expression assez complexe de $s$, qui est par ailleurs une variable entière. Une méthode pour résoudre ce problème consiste alors à calculer les différentes valeurs de $\overline{\overline{\Gamma}}$ pour $s$ croissant et de s'arrêter à $s^*$tel que :

$\overline{\overline{\Gamma}}(s^*) < \overline{\overline{\Gamma}}(s^* - 1)$ et $\overline{\overline{\Gamma}}(s^*) < \overline{\overline{\Gamma}}(s^* + 1)$

En fonction de $s$, $\overline{\overline{\Gamma}}(s)$ varie en effet de la façon suivante :




**Remarque importante :** Les hypothèses choisies pour l'expression du coût total (linéarité du coût d'exploitation et introduction d'une valeur du temps d'attente) sont des hypothèses fortes.

Nous constatons donc que lorsque les deux hypothèses : service exponentiel et arrivées poissonniennes sont respectées, il est assez aisé de résoudre le système à une ou plusieurs stations. La méthode différentielle qui consiste à passer par les probabilités de transition entre deux instants infiniment rapprochés $t$ et $t + dt$ est particulièrement

adaptée à ce cas, grâce aux propriétés de la loi de Poisson. Cela dit, lorsque les arrivées ne sont pas poissonniennes ou que le service n'est pas exponentiel, comment traiter ce genre de problème ?

Dans ce cas, la difficulté vient du fait que les probabilités de transition peuvent avoir des expressions complexes, si par exemple, il n'y a pas stationnarité pour la loi des arrivées (les probabilités de transition dépendraient alors du temps) ou s'il n'y a pas absence de mémoire (les probabilités de transition dépendraient alors des évènements antérieurs). La méthode différentielle paraît alors inapte à résoudre ce type de problèmes. Elle peut être cependant adaptée, pour certaines lois et au prix de complications souvent touffues.

Par ailleurs, une autre méthode que la méthode différentielle, appelée méthode intégrale, permet dans certains cas d'étudier des systèmes à arrivées non poissonniennes et services non exponentiels. Nous allons exposer rapidement les principes de cette méthode sur le cas simple d'une seule station.

## 11.3. RETOUR AU CAS D'UNE SEULE STATION. METHODE INTEGRALE

Dans la méthode intégrale, on n'examine pas les probabilités d'état à chaque instant $t$, mais on s'intéresse aux clients individualisés.

Dans cet esprit, on numérote ces clients, dans leur ordre d'arrivée par les numéros $1,2, \ldots n \ldots$ etc , et l'on s'intéresse aux grandeurs suivantes :

$\theta_n$ : instant d'arrivée du client *n*

$\tau_n$ : temps d'attente dans le système du client *n*

$s_n$ : temps de service du client n

$a_n = \theta_n - \theta_{n-1} =$ intervalle entre les instants d'arrivée des clients $n \; et \; n-1$

$v_n =$ temps d'attente dans la file de l'unité $n$ ...

On a alors les égalités suivantes :
$\tau_n = s_n + v_n$
$\tau_n = s_n + v_n$ et par ailleurs :
Si
$\theta_{n-1} + \tau_{n-1} \leq \theta_n$
soit si
$a_n \geq \tau_{n-1}$
alors
$v_n = 0$

Par contre si
$a_n < \tau_{n-1}$,
alors

$v_n = \tau_{n-1} - a_n$

Posons :

$F_n(x) = p\,[v_n \le x]$

et

$G_n(x) = p\,[\tau_n \le x]$

Si l'on fait alors les deux hypothèses suivantes :

- les variables $a_n$ sont indépendantes mutuellement et ont pour loi de densité $a(x)$,
- les variables $s_n$ sont mutuellement indépendantes et ont pour loi de densité $s(x)$,

On peut établir des lois de récurrence pour les fonctions $F_n(x) et\ G_n(x)$. En effet, on a :

$G_n(x) = P[s_n + v_n \le x]$, ce qui peut s'écrire encore sous la forme du produit de convolution :

$$G_n(x) = \int_0^x s(y)\ F_n(x-y)\ dy$$

Par ailleurs

$$F_n(x) = P[v_n \le x] = P[\tau_{n-1} - a_n \le x]$$

$$= P[\tau_{n-1} \le x + a_n]$$

ce qui peut encore s'écrire :

$$F_n(x) = \int_0^\infty a(y)\ G_{n-1}(x+y)\ dy$$

Par ailleurs, les conditions initiales sont :

$F_1(x) = 1 \qquad \forall x$

et

$$G_1(x) = \int_0^x s(y)\ dy$$

Si l'on suppose qu'à l'instant initial, la file d'attente est nulle.

De proche en proche, on peut alors calculer les fonctions $F_n(x)\ et\ G_n(x)$ qui donnent les distributions des temps d'attente des clients. Ce calcul est évidemment laborieux d'une façon générale, et l'on préfère se placer, là aussi, dans le cas du régime permanent.

Si ce dernier existe alors :

$F_n(x) \to F(x)$

Lorsque $n \to \infty$
et $G_n(x) \to G(x)$
et $F(x)\ et\ G(x)$ répondent aux équations :

$$F(x) = \int_0^{\infty} a(y)\, G(y+x)\, d\,y \qquad (14)$$

$$G(x) = \int_0^{x} s(y)\, F(x-y)\, d\,y$$

Dans certains cas, par exemple pour des lois d'arrivée classiques (Erlang-k (1), ou hyper exponentiel), ces équations peuvent se résoudre relativement facilement, grâce au calcul opérationnel.

**Exemple** : Arrivées poissonniennes et service quelconque (à titre d'exercice).
On a alors :

$a(y) = \lambda e^{-\lambda y}\ si\ \lambda$ est le taux d'arrivée.
Donc

$$F(x) = \lambda \int_0^{\infty} e^{-\lambda y} \qquad G(y+x)\, d\,y$$

Soit, en faisant le changement de variable $y + x = u$

$$F(x) = \lambda \int_x^{\infty} e^{-\lambda(u-x)}\ (u)\, du \qquad (15)$$

$$= \lambda[\int_0^{\infty} e^{\lambda(x-u)}\ (u)\, du - \int_0^{x} e^{\lambda(x-u)}\ G(u)\ du]$$

Appelons $F^*(p)$ la transformée de Laplace de la fonction $F(x)$.

Rappel : $F^*(p) = \int_0^{\infty} e^{-pt} F(t) dt$
Pour obtenir $F^*(p)$, il faut d'abord calculer la transformée de

$$\lambda\, e^{\lambda x} \int_0^{\infty} e^{-\lambda u}\ G(u)\, du$$

Cette transformée est

$$\frac{\lambda}{p-\lambda} \int_0^{\infty} e^{-\lambda u}\ G(u)\, du \qquad (16)$$

D'après la formule (15) de $F(x)$, on a :

$$F(0) = \lambda \int_0^{\infty} e^{-\lambda u}\ G(u)\, du$$

Donc l'expression (16) est égale à

$$\frac{F(0)}{p-\lambda} \tag{17}$$

Par ailleurs

$$\int_0^x e^{\lambda(x-u)}\ G(u)\,du$$

est le produit de convolution des fonctions $e^{\lambda x}$ et $G(x)$. Or la transformée de Laplace d'un produit de convolution, on le sait, est égale au produit des transformées de Laplace. Donc, la transformée de

$$\lambda \int_0^x e^{\lambda(x-u)}\ G(u)\,du$$

est égale à

$$\frac{\lambda}{p-\lambda}\ G^*(p) \tag{18}$$

si $G^*(p)$ est la transformée de $G(x)$. Au total, la transformée de $F(x), F^*(p)$ est égale, d'après (15), (17) et (18) à :

$$F^*(p) = \frac{F(0)}{p-\lambda} - \frac{\lambda}{p-\lambda}\ G^*(p)$$

$$F^*(p) = \frac{F(0)-\lambda\, G^*(p)}{p-\lambda} \tag{19}$$

Par ailleurs, d'après (14), on a

$$G^*(p) = s^*(p)\ F^*(p) \tag{20}$$

Si $s^*(p)$ est la transformée de la fonction $s(t)$.
(19) et (20) donnent alors :

$$F^*(p) = \frac{F(0)}{p-\lambda+\lambda\, s^*(p)} \tag{21}$$

Cette expression permet de calculer $F^*(p)\ donc\ F(x)$, si l'on se donne $s(x)$, donc $S^*(p)$, à condition que l'on puisse déterminer $F(0)$.
Pour déterminer $F(0)$, utilisons cette formule (21). On voit facilement que si l'on fait tendre $p$ vers 0, $F(0)$ est indéterminé.
En effet, $F^*(p) \to \infty$ lorsque $p \to 0$ et $s^*(p) \to 1$ lorsque $p \to 0$ d'après la signification de $F(x)$ et de $s(x)$.
Donc $p - \lambda + \lambda s^*(p) \to 0$ lorsque $p \to 0$.

Faisons alors intervenir la fonction :

$$S(x) = \int_x^{\infty} s(u)\,du = 1 - \int_0^x s(u)\,du$$

(probabilité que la durée de service soit supérieure à *x)*. Si l'on appelle $S^*(p)$ la transformée de $S(x)$, on a (propriété classique de la transformée de Laplace) :

$$S^*(p) = \frac{1 - s^*(p)}{p}$$

ou encore $s^*(p) = 1 - pS^*(p)$ (21) s'écrit alors :

$$F^*(p) = \frac{F(0)}{p[1 - \lambda\, S^*(p)]}$$

Soit $F(0) = pF^*(p)[1 - \lambda S^*(p)]$

Faisons tendre alors $p$ vers 0. Toujours d'après les propriétés de la transformée de Laplace, on a :

$$\lim_{x\to\infty} F(x) = \lim_{|P|\to 0} p\ F^*(p) = 1$$

Par ailleurs

$$S^*(p) = \int_0^{\infty} e^{-pt}\ S(t)\ dt$$

et

$$\lim_{p\to 0} S^*(p) = \int_0^{\infty} S(t)dt$$

D'où

$$F(0) = 1 - \int_0^{\infty} S(t)\,dt$$

Or $\int_0^{\infty} S(t)dt$ a une signification bien particulière. C'est en effet le temps moyen de service $\bar{s}$, car

$$\bar{s} = \int_0^{\infty} t\, s(t)\,dt = [-t\, S(t)]_0^{\infty} + \int_0^{\infty} S(t)\,dt$$

et $\lim_{t\to+\infty}[-ts(t)] = 0, si\ \bar{s}\ existe.$

Donc

$$F(0) = 1 - \lambda\, \bar{s}$$

On retrouve là le résultat obtenu pour le service exponentiel, où $\bar{s} = \frac{1}{\mu} F(0)$ est en effet la probabilité qu'un client n'attende pas, donc que la station soit inoccupée.

Finalement

$$F^*(p) = \frac{1}{p} \frac{1-\lambda \bar{s}}{1-\lambda S^*(p)} \qquad (22)$$

formule qui permet d'obtenir $F^*(p)$ si l'on connaît la loi des temps de service, donc $F(x) et\ G(x)$ par (14).

Par exemple, si le service est exponentiel, de taux $\mu$ ; on trouve facilement,

$(S(x) = e^{-\mu x})$ si l'on suppose $\tau = \frac{\lambda}{\mu}$

$$F(x) = 1 - \tau e^{(\lambda-\mu)x} \qquad (23)$$

formule qui n'est évidemment valable que si $\lambda < \mu$.

Nous voyons donc que la méthode intégrale permet de résoudre le système ouvert à une station, pour des lois d'arrivée et de service divers. Il faut cependant remarquer que, malheureusement, cette méthode est fort mal adaptée au traitement d'un système à plusieurs stations. En effet l'égalité

$$v_n = \max [0, \tau_{n-1} - a_n]$$

est toujours valable, mais pour une station particulière : $a_n$ représente l'intervalle de temps séparant l'arrivée de deux clients successifs dans une des files d'attente. Mais alors, la loi de $a_n$ dépend du système de choix des files d'attente par les clients, ce qui complique en général notablement le problème.

## 11.4. SYSTEMES PLUS COMPLEXES

Nous avons constaté qu'un phénomène d'attente pouvait très vite se compliquer : un système à plusieurs stations et arrivées non poissonniennes ou service non exponentiel peut en général difficilement se formaliser. D'autres circonstances peuvent encore rendre la modélisation pénible. Nous allons les passer rapidement en revue :

### 11.4.1. Stations de service non identiques

L'évolution du système dépend du choix des clients. En particulier lorsque le nombre de stations occupées est $n < s$. Il importe de déterminer ce choix pour calculer les probabilités de transition. On résout le problème en introduisant une distribution sur les choix des clients, au prix de complications laborieuses.

### 11.4.2. Systèmes fermés

Jusqu'ici, les clients étaient supposés en nombre illimité. Dans beaucoup d'exemples (atelier de réparation recevant des machines provenant d'une usine), le nombre de clients est limité et est par ailleurs recyclé dans le système.

Lorsqu'il est possible de conserver les hypothèses d'arrivées poissonniennes et de services exponentiel, il convient de remarquer que le taux d'arrivée $\lambda_n$ et le taux de service $\mu_n$ dépend de $n$, nombre de clients dans le système d'attente.

Dans ces conditions, les probabilités de transition entre $t \; et \; t + dt$ seront les suivantes:

$n > 0$

$E_n \to E_n \quad 1 - (\lambda_n + \mu_n)dt$

$E_n \to E_{n+1} \quad \lambda_n dt$

$E_n \to E_{n-1} \quad \mu_n dt$

$n = 0$

$E_0 \to E_0 \quad 1 - \lambda_0 dt$

$E_0 \to E_1 \quad \mu_1 dt$

Un phénomène qui est caractérisé par de telles probabilités de transition est appelé « processus de naissance et de mort ». Les équations d'état correspondantes sont :

$$p_n'(t) = \lambda_{n-1} \; p_{n-1}(t) - (\lambda_n + \mu_n) \; p_n(t) + \mu_{n+1} \; p_{n+1} \; (t) \qquad (24)$$

$$p_0'(t) = -\lambda_0 \; p_0(t) + \mu_1 \; p_1(t)$$

Ces équations peuvent décrire l'évolution de nombreux systèmes, caractérisés par une population d'éléments ou d'individus variant par l'effet de disparitions et de naissances.

En ce qui concerne les phénomènes d'attente, remarquons que les deux systèmes que nous avons étudiés constituent un tel processus.

Pour le premier (une station, arrivées poissonniennes, service exponentiel), on a

$\lambda_n = \lambda \qquad \forall_n \geq 0$

$\mu_n = \mu \qquad \forall_n \geq 0$

Pour le second ($s$ stations, arrivées poissonniennes, services exponentiels de taux identique $\mu$).

$\lambda_n = \lambda \qquad \forall_n \geq 0$

$\mu_n = n\mu \qquad n \leq s$

$\mu_n = s\mu \qquad n > s$

En ce qui concerne les systèmes fermés, si l'on prend l'exemple de l'atelier de réparation, on peut avoir la configuration suivante :

Si $m$ est le nombre de machines du parc,

Si $n$ est le nombre de machines en réparation,

Si $s < m$ est le nombre de stations de réparation,

Si la loi des pannes des machines est poissonienne, de taux $\lambda$ et

Si le temps de réparation par station est exponentiel, de taux $\mu\lambda_n = (m-n)\lambda$

$\mu_n = n\mu \qquad n \leq s$

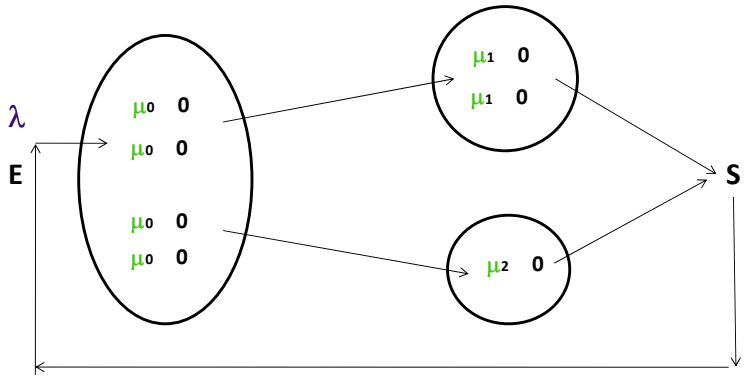
$\mu_n = s_\mu \qquad n > s$

ce qui permet d'écrire les équations (24)

avec des valeurs de $\mu_n$ et $\lambda_n$ pour $n = 0{,}1 \ldots m.$ Le régime permanent est obtenu de la même façon que pour les cas étudiés auparavent.

### 11.4.3. Réseaux d'attente

Très souvent, les clients n'ont pas à subir une seule file d'attente, mais plusieurs successivement. (exemple simple du paiement d'un chèque dans une banque où le client subit d'abord des formalités administratives et attend ensuite à la caisse).
On peut avoir alors des systèmes plus ou moins complexes, appelés « réseaux d'attente ».
Par exemple, on peut avoir le réseau suivant :




Soit :

- Une source de clients arrivant suivant le modèle poissonnien, avec un taux $\lambda$.
- Un premier centre d'attente avec quatre stations « exponentielles » le taux $\mu_0$.
- Deux autres centres d'attente en parallèle : les clients se rendent au premier avec la probabilité $p_1$ au second avec la probabilité $p_2 = 1 - p_1$.
  (la flèche reliant la sortie $S$ à l'entrée $E$, indique qu'il s'agit d'un système fermé).

Sous certaines hypothèses, ces réseaux peuvent se prêter assez bien à la formalisation (c'est le cas de l'exemple choisi).

### 11.4.4. Autres facteurs de complexité

D'autres circonstances peuvent compliquer encore la modélisation. Citons :

- les priorités : certains clients sont servis en priorité.
- l'impatience : si la file d'attente devient trop longue, certains clients peuvent s'impatienter et quitter le système.
- la limitation de la file d'attente, par exemple lorsque le centre d'attente a une capacité d'accueil limitée.

Encore une fois, sous certaines hypothèses, ces facteurs peuvent être intégrés dans la modélisation, et ils l'ont été parfois avec succès : de nombreuses publications existent à cet égard.

Cela dit, lorsque le système que l'on veut étudier se révèle rebelle à toute modélisation (lois d'arrivées ou de service complexes, systèmes sous forme de réseaux, etc...) que peut-on faire ? Il existe encore une solution qui consiste à effectuer une *simulation* sur ordinateur. Cette opération, dont nous ne pouvons parler dans le cadre de cet ouvrage, consiste à décrire sur ordinateur une « tranche de vie du système », cela par l'intermédiaire de deux catégories d'éléments :

- des échantillons des différentes variables aléatoires intervenant dans le système, obtenus soit par des tables, soit par une génération automatique (nombres pseudo-aléatoires).
- des relations logiques du système (succession des stations, ordre d'arrivée dans les files, choix des stations par les clients etc.)

À partir de ces éléments, on décrit tous les évènements qui interviennent, et l'on obtient ainsi une évolution possible du système (possible puisque l'on part de réalisations d'échantillons des variables aléatoires). Mais si l'intervalle de temps traité est suffisamment important, ainsi que le nombre d'évènements correspondants, on peut extraire de cette expérience (la loi des grands nombres le permet) des résultats intéressant le régime permanent, tels que les temps d'attente moyens, les taux d'occupation des différentes stations etc.

Cette méthode est très utilisée et son importance déborde largement le cadre des phénomènes d'attente. Signalons seulement qu'en général, elle est coûteuse en temps informatique, et qu'il convient de n'y avoir recours que lorsque toute étude analytique s'avère être impossible.

# Chapitre 12 • Problèmes de défaillances d'équipements

## INTRODUCTION

La Recherche Opérationnelle appliquée à l'aléatoire s'est développée considérablement dans le domaine de l'usure de l'équipement. Le calcul économique, on le sait, fournit des règles de choix lorsque la durée de vie des équipements peut être considérée comme fixe. Dans le cas où cette même durée de vie est aléatoire, le problème devient très compliqué : une panne pouvant entraîner des dépenses prohibitives (car en plus des dépenses même de réparation, il faut souvent ajouter d'autres « coûts », tels que les perte dues à une mise hors service d'un système), il convient *a priori* de s'en prémunir le plus possible; mais une politique intensive de réapprovisionnement et de remplacement peut également être très coûteuse. Il convient donc de trouver le juste équilibre, ce qui ne peut se faire que par l'étude précise des phénomènes en cause.

La durée de vie des équipements concernés étant aléatoire, il est tout d'abord nécessaire de cerner les lois de probabilité qui la régissent.

## 12.1. DEFINITIONS ET GENERALITES

Soit un équipement donné (machine, appareil, composant, système, etc.), destiné à assurer une certaine fonction. Nous appellerons $T$ sa durée de vie, c'est-à-dire l'intervalle de temps pendant lequel l'équipement assure cette fonction sans défaillance. $T$ est ici une variable aléatoire continue. Nous appellerons *fiabilité* la fonction $v(t)$ suivante :

$$v(t) = p[T \geq t] \qquad (1)$$

c'est-à-dire plus précisément :

On appelle fiabilité d'un équipement, la probabilité pour que cet équipement assure une fonction donnée sans défaillance pendant une durée donnée, sous les contraintes correspondant aux conditions d'emploi pour lesquelles il a été conçu.

On voit que si $F(t)$ est la fonction de répartition de la variable aléatoire $T$ :

$$v(t) = 1 - F(t) \qquad (2)$$

Classiquement, on peut également considérer la fonction de densité $i(t)$ de la variable $T$, que nous appellerons *mortalité relative*

$$i(t)\,dt = P[t \leq T < t + dt]$$

$$i(t) = F'(t) = -v'(t) \qquad (3)$$

Une autre notion très importante est le *taux de défaillance* λ(t). C'est la probabilité que l'équipement, ayant vécu jusqu'à l'instant $t$, ait une défaillance entre $t \; et \; t + dt$.

On a donc:

$$\lambda(t) = \frac{P[t \leq T < t + dt]}{P[T \geq t]}$$

Soit

$$\lambda(t) = -\frac{v'(t)}{v(t)} \qquad (4)$$

On doit avoir $v(0) = 1$.

L'équation (4) peut alors s'écrire, en résolvant par rapport à $v(t)$.

$$v(t) = e^{-\int_0^t \lambda(u)du} \qquad (5)$$

On voit que, pour spécifier complètement la variable aléatoire $T$ , il est indifférent de se donner $F(t), v(t), i(t) ou \; \lambda(t)$.

Si l'on continue à dresser la liste des concepts importants, il convient de signaler le temps moyen de fonctionnement $E$ qui est évidemment l'espérance mathématique de $T$.

$$\bar{t} = E(T) = \int_0^\infty t \; i(t) \; dt \qquad (6)$$

$\bar{t}$ est souvent appelé $MTBF$, suivant une terminologie anglo-saxonne (Mean Time Between Failures).

On peut donner une autre expression de $\bar{t}$. Si l'on intègre (6) par parties, on trouve en effet :

$$\bar{t} = -[t \; v(t)]_0^\infty + \int_0^\infty v(t) \; dt$$

Dans les cas les plus usuels, $\lim_{t \to \infty} tv(t) = 0$

donc

$$\bar{t} = \int_0^\infty v(t) dt \qquad (7)$$

De même, la variance (ou l'écart type) de $T$ est une grandeur qu'il est souvent utile de connaître.

$\sigma_T^2 = E[(T - \bar{t})^2] = E(T^2) - \bar{t}^2$

$\sigma_T^2 = \int_0^\infty t^2\, i(t)dt - (\bar{t})^2$

ou encore

$\sigma_T^2 = 2\int_0^\infty tv(t) - \bar{t}^2$

## 12.2. CLASSEMENT DES APPAREILS SUIVANT LES LOIS DE PROBABILITES DE LEUR DUREE DE VIE

On peut classer les appareils selon leur taux de défaillance $\lambda(t)$.

a) $\lambda(t)$ constant $= \lambda$.

Nous nous trouvons ici dans le cas de ce que l'on pourrait appeler les défaillances purement accidentelles. Les causes de ces défaillances surviennent suivant un processus de Poisson. En effet, la formule (5) donne

$v(t) = e^{-\lambda t}$

et l'on reconnaît le modèle exponentiel pour la durée de vie. Si l'intervalle de temps qui sépare deux défaillances successives obéit à la loi exponentielle, c'est que les évènements « défaillance » obéissent à la loi de Poisson, d'après ce qu'on a vu pour les phénomènes d'attente.

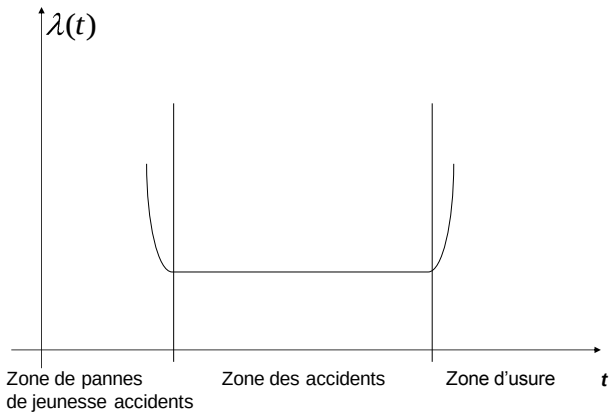
b) $\lambda(t)$ croissant avec $t$

C'est *a priori* le cas le plus courant : il s'agit d'équipements soumis à un phénomène d'usure, de dégradation (vibrations, chocs, érosion etc.) si bien que leur taux de défaillance croît constamment avec $t$.

c) $\lambda(t)$ décroissant

Ce cas peut paraître curieux, mais il existe souvent, au moins sur certaines plages de $t$ : il s'agit alors de ce que l'on peut appeler les maladies de jeunesse (malfaçons à la fabrication, imperfection du contrôle, etc.).

On peut avoir bien sûr, pour un équipement donné, les cas a, b et c à la fois; $\lambda(t)$ a alors la forme suivante :




## 12.3. QUELQUES LOIS DE PROBABILITES USUELLES POUR LES DUREES DE VIE

***a) Le modèle poissonnien***

Le plus connu et certainement le plus utilisé (au moins dans les calculs). On a :

$$\begin{aligned} v(t) &= e^{-\lambda t} \\ F(t) &= 1 - e^{-\lambda t} \\ i(t) &= \lambda \quad e^{-\lambda t} \\ \lambda(t) &= \lambda \end{aligned} \qquad (8)$$

(on trouve ici une propriété fondamentale de la loi de Poisson : la fonction de densité conditionnelle, ou taux de défaillance, est constante; on dit également que le processus est sans mémoire).

Enfin

$$\bar{t} = \frac{1}{\lambda} \qquad \sigma_T^2 = \frac{1}{\lambda^2}$$

***b) Le modèle Erlang-k***

Ce modèle dû au danois Erlang, qui étudiait les phénomènes d'attente dans un standard téléphonique, est dérivé du modèle poissonnien : supposons que les causes de défaillance surviennent suivant une loi de Poisson de taux $\lambda$, mais que la défaillance soit consécutive à la succession de $k$ de ces évènements (et non d'un seul, comme dans le cas de Poisson).

D'après cette définition, la durée de vie $T$ peut être écrite sous la forme :

$$T = \sum_{i=1}^{k} T_i$$

les $T_i$ étant des variables exponentielles indépendantes de taux $\lambda$.
A titre d'exercice, calculons la fonction de densité de la variable $T$. Si l'on appelle $N(t)$ le nombre d'évènements étant arrivés à l'instant $t$ on a :

$$F(t) = P[T < t] = P[N(t) \geq k]$$

Or l'on sait que

$$P[N(t) = n] = e^{-\lambda t} \frac{(\lambda t)^n}{n!}$$

(expression classique de la loi de Poisson).

Donc

$$P[N(t) \geq k] = 1 - p_0(t) - p_1(t) \ldots p_{k-1}(t)$$

$$F(t) = 1 - \left[ e^{-\lambda t} - e^{-\lambda t} \frac{\lambda t}{1!} \ldots - e^{-\lambda t} \frac{(\lambda t)^{k-1}}{(k-1)!} \right]$$

et, en dérivant :

$$i(t) = \lambda e^{-\lambda t} \left[ 1 + \frac{\lambda t}{1!} + \frac{\lambda^2 t^2}{2!} + \ldots \frac{(\lambda t)^{k-1}}{(k-1)!} \right]$$

$$- e^{-\lambda t} \left[ \frac{\lambda t}{1!} \frac{2\lambda^2 t}{2!} + \ldots \frac{(k-1)\lambda^{k-1} t^{k-2}}{(k-1)!} \right]$$

les termes de degré inférieur à $k - 1$ (pour $t$) s'annulant deux à deux, il reste :

$$i(t) = \lambda e^{-\lambda t} \frac{(\lambda t)^{k-1}}{(k-1)!} \qquad (9)$$

qui est la fonction de densité de la loi Erlang-k. Quant aux fonctions $F(t), v(t) et\ \lambda(t)$, elles n'ont pas d'écriture simple. Par contre, on a, bien évidemment :

$$\bar{t} = \frac{k}{\lambda} \qquad \sigma_T^2 = \frac{k}{\lambda^2}$$

**Remarque 1** : le taux d'avarie $\lambda(t) = -\frac{v'(t)}{v(t)}$ dépend ici de $t$ ; c'est donc un processus avec mémoire. Il est d'ailleurs facile de voir que $\lambda(t)$ est croissant avec $t$.

**Remarque 2** : Si l'on prend la variable $\cup = \lambda t \cup$ on trouve, pour loi de densité de $\cup$, d'après (9)

$$g(u) = e^{-u} \frac{u^{k-1}}{(k-1)!}$$

on retrouve ici une loi très classique, qui est la loi $\gamma$. On peut poser :

$$g(u) = \frac{1}{\Gamma(k)} e^{-u} u^{k-1} \qquad (10)$$

avec $\Gamma$ fonction eulérienne :

$$\Gamma(x) = \int_0^\infty e^{-u} u^{x-1} du$$

(on sait que $\Gamma(n) = (n-1)!$, avec $n$ entiers)

Cette remarque conduit à généraliser le processus Erlang-k, donné par (9), à des processus donnés par (10), avec $k$ non entier, mais réels : on a alors des durées de vie régies par des lois $\Gamma$.

***c) Le modèle hyperexponentiel***

Ce modèle correspond à un schéma plus complexe d'intervention de causes de défaillance.

Supposons qu'un système puisse être dans $k$ états incompatibles, avec les probabilités $w_1, w_2 \ldots w_k$ (chacun de ces états correspond à un ensemble de causes de défaillance).

Lorsque l'équipement est dans l'état $E_i$, il peut avoir des défaillances qui surviennent suivant le modèle poissonnien de taux $\lambda_i$. Dans ces conditions, calculons $v(t)$.

$v(t)$ est la probabilité qu'entre $0$ $et$ $t$, il n'y ait aucune défaillance = probabilité que l'équipement soit dans l'état $E_1$ et qu'il n'y ait aucune défaillance dans cet état, etc.
Donc :

$$v(t) = \sum_{i=1}^{k} w_i e^{-\lambda_i t} \qquad (11)$$

avec

$$\sum_{i=1}^{k} w_i = 1$$

on a alors :

$$F(t) = 1 - \sum_{i=1}^{k} w_i \, e^{-\lambda_i t}$$

$$i(t) = \sum_{i=1}^{k} w_i \;\; \lambda_i \, e^{-\lambda_i t}$$

$$\lambda(t) = \frac{\sum_{i=1}^{k} w_i \;\; \lambda_i \, e^{-\lambda_i t}}{\sum_{i=1}^{k} w_i \;\; e^{-\lambda_i t}} \qquad (12)$$

$$\bar{t} = \sum_{i=1}^{k} \frac{w_i}{\lambda_i}$$

$$\sigma_T^2 = 2 \sum_{i=1}^{k} \frac{w_i}{\lambda_i^2} - \overline{t^2}$$

Le modèle hyper exponentiel se prête bien à la description de temps de vie très dispersés.

***d) La loi de Weibull***

La loi de Weibull correspond à la forme la plus simple possible d'un taux de défaillance croissant. Si l'on prend

$$v(t) = \; e^{-(\alpha t)^{\beta}} \qquad (13)$$

avec $\alpha \geq 0$ et $\beta > 1$ on trouve

$$\lambda(t) = \beta \, \alpha^{\beta} \;\; t^{\beta - 1}$$

$t$ et $\alpha_T^2$ ont des expressions relativement compliquées.

***e) Autres lois***

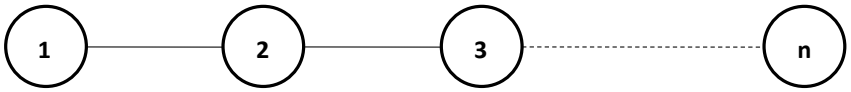
On peut évidemment, en dehors de ces quatre modèles (1) très connus, prendre d'autres lois de probabilités, comme la loi normale, la loi lognormale, etc.

***f) Système comprenant plusieurs composants***

Supposons que l'on ait un équipement complexe constitué de $n$ composants dont on connaît les lois de probabilité des durées de vie. La durée de vie de l'ensemble est une variable aléatoire qui est en général une fonction complexe des durées de vie de chacun des composants. Toutefois, deux cas simples donnent lieu à une formalisation aisée :

**1 - Composants en série**

On a la structure suivante




et l'ensemble ne survit à l'instant $t$ que si chacun des composants a vécu jusqu'à cet instant. Si $v(t)$ est la fiabilité de l'ensemble et $v_1(t) \dots v_i(t) \dots v_n(t)$ les fiabilités de chacun des composants, on a alors :

$$v(t) = \prod_{i=1}^{n} v_i(t) \qquad (14)$$

ce qui permet de calculer $v(t)$, et donc l'ensemble des caractéristiques de l'équipement, lorsque l'on a $v(t)$. Ainsi, si

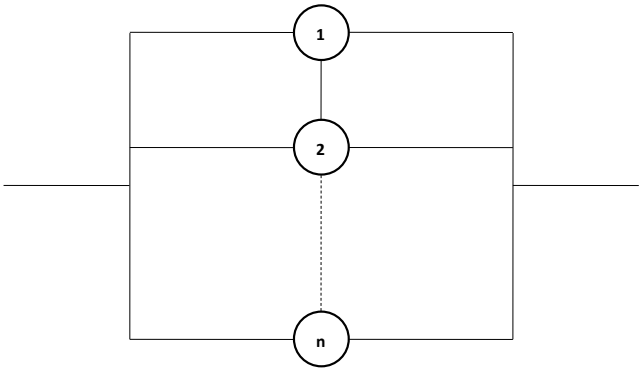
$$v_i(t) = e^{-\lambda_i t}$$

$$v(t) = e^{-t \sum_{i=1}^{n} \lambda_i}$$

On retrouve ce modèle exponentiel.

**2 - Composants en parallèle**

La configuration de l'équipement est alors la suivante :




et, à l'instant $t$, l'équipement est en panne si aucun composant n'a survécu au même instant. on a alors :

$$F(t) = \prod_{i=1}^{n} F_i(t)$$

si $F_1 \ldots F_i \ldots F_n$ sont les fonctions de répartition des durées de vie des composants $1 \ldots i \ldots n$. soit

$$v(t) = 1 - \prod_{i=1}^{n} [1 - v_i(t)] \qquad (15)$$

## 12.4. RACCORDEMENT D'UNE CHRONIQUE DE DEFAILLANCE A UNE LOI CONNUE

Bien entendu, dans la pratique, l'on ne dispose pas immédiatement des lois de probabilités des durées de vie. Tout ce qu'il est possible de faire, c'est d'observer, pour un équipement donné, une séquence assez importante de défaillances, qui fournit alors la réalisation d'un *échantillon* de la variable aléatoire : durée de vie. Un test statistique[14] permet ensuite, en théorie, de proposer une loi de probabilité décrivant « au mieux » le phénomène.

Cela dit, une telle opération pose problème. Supposons en effet que nous ayons l'expérience suivante : on a un parc de 1000 exemplaires d'un même équipement, que l'on met tous en marche à l'instant $t = 0$. On découpe le temps en phases $t$=0,1, 2, etc. et l'on observe à chaque phase le nombre d'équipements encore en vie $n(t)$. On a par exemple le tableau suivant :

| t | (t) |
|---|---|
| 0 | 1000 |
| 1 | 990 |
| 2 | 960 |
| 3 | 900 |
| 4 | 820 |
| 5 | 680 |
| 6 | 450 |
| 7 | 250 |
| 8 | 130 |
| 9 | 60 |
| 10 | 20 |
| 11 | 0 |

[14] par exemple, le test de chi-deux, le plus connu, mais il y en a d'autres.

On peut, sur la base de ces chiffres, proposer un modèle *discontinu* des défaillances, en définissant des concepts équivalents à ceux que nous avons définis pour le cas continu. Ainsi :

$\frac{n(t)}{n(0)}$ proportion d'équipements encore en vie à l'instant $t$, équivalent à $v(t)$

$1 - \frac{n(t)}{n(0)} = \frac{n(0)-(t)}{n(0)}$ équivalent à $F(t)$

$\frac{n(t-1)-n(t)}{n(0)}$ =proportion d'équipements ayant subi une défaillance entre $t-1$ *et* $t$ : équivalent à $i(t)$

$\frac{n(t-1)-n(t)}{n(t-1)}$ = proportion des équipements ayant une défaillance entre $t-1$ et $t$, prise par rapport aux équipements ayant vécu jusqu'à $t-1$ : équivalent à $\lambda$(t).

Ainsi, pour le tableau précédent, on peut, en utilisant ces concepts, fournir les résultats suivants :

| t | v(t) | F(t) | i(t) | λ(t) |
|---|---|---|---|---|
| 0 | 1 | 0 | | |
| 1 | 0,99 | 0,01 | 0,01 | 0,01 |
| 2 | 0,96 | 0,04 | 0,03 | 0,031 |
| 3 | 0,90 | 0,10 | 0,06 | 0,063 |
| 4 | 0,82 | 0,18 | 0,08 | 0,089 |
| 5 | 0,68 | 0,32 | 0,14 | 0,17 |
| 6 | 0,45 | 0,55 | 0,23 | 0,338 |
| 7 | 0,25 | 0,75 | 0,20 | 0,445 |
| 8 | 0,13 | 0,87 | 0,12 | 0,48 |
| 9 | 0,06 | 0,94 | 0,07 | 0,538 |
| 10 | 0,02 | 0,98 | 0,04 | 0,618 |
| 11 | 0 | 0 | 0,02 | 0 |

Dans ces conditions, deux attitudes sont possibles, lorsque, devant un problème particulier concernant ce type d'équipement, il s'avère nécessaire de connaître les lois de probabilités qui régissent les durées de vie :

- soit on se place dans l'optique « continue » et on cherche alors, par des tests statistiques à approcher au mieux les courbes expérimentales $v(t), i(t), ou\ \lambda(t)$

par des courbes théoriques continues, correspondant à un modèle classique (Weibull ou Erlang-k etc.).

- soit on se contente de la chronique que l'on a obtenue, et l'on se situe alors dans l'optique « discontinue » : on traite le problème posé en découpant le temps de la même façon que pour l'expérience effectuée, et on prend le parti de confondre fréquences et probabilités.

Cette solution, qui est souvent choisie en dernier ressort, correspond à l'idée suivante : l'ajustement d'une courbe théorique sur une série statistique n'est par définition jamais parfait; les tests statistiques utilisés permettent seulement d'affirmer qu'il n'est pas absurde de supposer que le processus obéit effectivement à la loi ajustée. Dans ces conditions, pourquoi se priver, plutôt que d'utiliser des fonctions théoriques dont on n'est pas sûr qu'elles décrivent convenablement les phénomènes, de tirer parti de toutes les informations fournies par l'expérience, c'est-à-dire finalement la chronique observée, telle qu'elle se présente ?

Dans ces conditions, on peut alors calculer $\bar{t}$ de la façon suivante :

$$\bar{t} = \sum_{t=1}^{\infty} t \;\; i(t)$$

ou encore :

$$\bar{t} = \sum_{t=1}^{\infty} \frac{t(n(t-1) - n(t))}{n(0)}$$

en développant :

$$t = \sum_{t=0}^{\infty} \frac{n(t)}{n(0)} = \sum_{t=0}^{\infty} v(t)$$

On retrouve une formule parfaitement analogue à celle obtenue pour le cas continu. Pour l'exemple considéré, on trouve $\bar{t} = 6{,}26$.

**Remarque :** En l'absence d'indications, il est plus cohérent en général de supposer les défaillances des équipements localisés en milieu de période. Dans ces conditions :

$$\bar{t} = \sum_{t=1}^{\infty} \left( t - \frac{1}{2} \right) \; i(t) \;\; dt = \sum_{t=0}^{\infty} v(t) - \frac{1}{2}$$

Ici, $\bar{t} = 5{,}76$

## 12.5. PROCESSUS DE RENOUVELLEMENT

Dans certains cas pratiques, on s'intéresse au processus suivant : à l'instant $0$, on met en marche un équipement de durée de vie aléatoire $T$ ; s'il tombe en panne, on le remplace

immédiatement par un équipement identique (même loi de probabilité de $T$). On obtient ainsi la suite $\theta_1, \theta_2, \ldots \theta_2, \ldots \theta_n \ldots$ qui sont les instants de défaillances successives. On donne à ce processus l'appellation de processus de renouvellement[15].

L'analyse de ce type de processus se fait en général, dans le cas continu, grâce à l'utilisation du calcul opérationnel. Donnons quelques exemples de grandeur que l'on peut ainsi évaluer.

### 12.5.1. Instant de la $n^{ième}$ défaillance

Appelons $S_n$ l'instant de la $n^{ième}$ défaillance. Si $T_1, T_2 \ldots T_n \ldots$ sont les durées de vie des équipements successifs mis en fonctionnement, on a : $S_n = T_1, T_2 \ldots T_n \ldots$

Donc $S_n$ est la somme de $n$ variables aléatoires indépendantes, et de même loi de probabilité; la loi de densité de $S_n$ peut être alors obtenue par le produit de convolution de la loi de densité de $T$ convoluée $n$ fois.

En fait, on passe en général par la transformée de Laplace, en profitant de ses propriétés (en particulier, transformée d'un produit de convolution = produit des transformées).

Si l'on appelle $\emptyset_n$ la loi de densité de $S_n$ et $i(t)$ la loi de densité de $T_i (i = 1 \ldots n)$ et $\emptyset_n^* (p) et\ i^*(p)$ leurs transformées de Laplace respectives, on a :

$$\phi_n^*(p) = [i^*(p)]^n \qquad (16)$$

*Exemple* : $T_i$ variable exponentielle de taux

$$i^*(p) = \frac{\lambda}{\lambda + p}$$

$$\phi_n^*(p) = \frac{\lambda^n}{(\lambda + p)^n}$$

La fonction objet de cette transformée est (d'après les tables)

$$\phi_n^*(t) = e^{-\lambda t} \ \frac{\lambda^n \ t^{n-1}}{(n-1)!}$$

On retrouve bien une loi Erlang-n.

---

15 Il s'agit ici, plus précisément, de renouvellement simple, un cas plus compliqué étant constitué par le fait qu'à $t = 0$ (début de l'observation), un équipement est déjà en marche (renouvellement modifié).

### 12.5.2. Nombre de renouvellements survenus à l'instant $t$

Appelons $R(t)$ le nombre de renouvellements survenus à l'instant $t$ on a :

$$P[R(t) < n] = P[S_n > t] \qquad n = 1, \ldots$$

$$= 1 - P[S_n < t]$$

$$= 1 - \int_0^\infty \phi_n(u)\, du$$

Si l'on a $\emptyset_n(t)$, on a donc la fonction de répartition de $R(t)$. Comme $R(t)$ est une variable aléatoire entière, il est plus pratique de connaître :

$$P_n(t) = P[R(t) = n] = P[R(t) < n+1] - P[R(t) < n]$$

Soit :

$$P_n(t) = \int_0^\infty [\phi_n(u) - \phi_{n+1}(u)]\, du \qquad (17)$$

Là aussi en général, on calcule $P_n(t)$ en passant par la transformée de Laplace. D'après une priorité caractéristique de cette dernière, si on l'appelle $P_n^*(p)$

$$P_n^*(p) = \frac{1}{p}[\phi_n^*(p) - \phi_{n+1}^*(p)] \qquad (18)$$

Soit

$$P_n^*(p) = \frac{[i^*(p)]^n}{p}[1 - i^*(p)]$$

Ainsi, si $i(t) = \lambda e^{-\lambda t}$ et $i^*(p) = \frac{\lambda}{\lambda+p}$ alors $P_n^*(p) = \frac{\lambda^n}{(\lambda+p)^{n+1}}$

dont la fonction objet est :

$$P_n(t) = e^{-\lambda t}\, \frac{(\lambda t)^n}{n!}$$

On retrouve l'expression classique de la loi de Poisson (nombre d'évènements survenus entre $0$ $et$ $t$).

### 12.5.3. Espérance mathématique du nombre de défaillances entre 0 et $t$.

Quel est en moyenne le nombre de défaillances, donc de renouvellements, entre $0$ $et$ $t$ ? On conçoit que, dans la pratique, il soit intéressant de connaître un tel élément. Appelons $H(t)$ce nombre moyen.

$$H(t) = E[R(t)]$$

$$H(t) = \sum_{n=0}^{\infty} P_n(t)n$$

En théorie, si l'on connaît $P_n(t), H(t)$ est théoriquement calculable. Là aussi, on passe en général par le calcul opérationnel.

On a, si $H^*(p)$ est la transformée de $H(t)$

$$H^*(p) = \sum_{n=0}^{\infty} n\, P_n^*(p)$$

D'après (18) :

$$H^*(p) = \sum_{n=0}^{\infty} \frac{n}{p}[\phi_n^*(p) - \phi_{n+1}^*(p)]$$

soit, en développant :

$$H^*(p) = \frac{1}{p} \sum_{n=1}^{\infty} \phi_n^*(p) \qquad (19)$$

et encore :

$$H^*(p) = \frac{1}{p} \sum_{n=1}^{\infty} [i^*(p)]^n$$

Comme $i^*(p)$ existe, et que, par ailleurs $|i^*(p)| < 1$ (facile à vérifier) :

$$H^*(p) = \frac{1}{p} \frac{i^*(p)}{1 - i^*(p)} \qquad (20)$$

Par exemple, dans le cas où $i(t) = \lambda e^{-\lambda t}$ et $i^*(p) = \frac{\lambda}{\lambda + p}$

$$H^*(p) = \frac{\lambda}{p^2}$$

La fonction inverse de $H^*(p)$ est : $H(t) = \lambda t$

Là encore, on retrouve un résultat bien connu du processus poissonnien.

**Remarque importante** : dans le cas général, on peut démontrer que le nombre moyen de renouvellements par unité de temps est asymptotiquement $(t \to \infty)$ égal à $\frac{1}{\bar{t}}$ si $\bar{t}$ est la durée de vie moyenne d'un équipement. C'est évidemment un résultat parfaitement intuitif.

### 12.5.4. Densité de renouvellement

La densité de renouvellement a la définition suivante : c'est la fonction $h(t)$ telle que $h(t)dt$est la probabilité qu'il y ait un renouvellement entre $t\ et\ t + dt$. On a :

$$h(t)\,dt = \sum_{n=1}^{\infty} \phi_n(t)\,dt$$

(en effet, la probabilité qu'il y ait un renouvellement entre $t\ et\ t + dt$, est égale à la probabilité que la première défaillance arrive entre $t\ et\ t + dt$, ajoutée à la probabilité que la seconde défaillance arrive entre $t\ et\ t + dt$, et $t + dt$, ajoutée à la probabilité que la seconde défaillance arrive entre $t$ et $t + dt$ etc.)

Soit

$$h(t) = \sum_{n=1}^{\infty} \phi_n(t)$$

et si $h^*(p)$ est la transformée de Laplace de $h(t)$

$$h^*(p) = \sum_{n=1}^{\infty} \phi_n^*(p)$$

$$= \sum_{n=1}^{\infty} [i^*(p)]^n$$

$$h^*(p) = \frac{i^*(p)}{1 - i^*(p)} \qquad (21)$$

Dans le cas poissonnien, on retrouve bien

$$h(t) = \lambda$$

Dans le cas général, ces formules permettent de démontrer que

$$h(t) \to \frac{1}{t} si\ t \to \infty$$

On peut calculer d'autres grandeurs intéressantes telles que la variance du nombre de renouvellements. On peut également partir d'hypothèses moins restrictives, par exemple ne plus supposer que le renouvellement est immédiat, mais faire intervenir au contraire un temps d'intervention ou de réparation entre deux équipements successifs (processus de renouvellement alterné). On peut avoir aussi des durées de vie différentes pour les différents équipements, des durées de vie non indépendantes etc. Dans certaines conditions, ces généralisations peuvent s'introduire dans la modélisation, au prix parfois de complications coûteuses.

## 12.6. PROCESSUS D'APPROVISIONNEMENT

On se pose ici le problème suivant : à l'instant $t = 0$, on met en fonctionnement $N(0)$ équipements identiques, $N(0)$ étant en général grand. À chaque instant $t$, on veut avoir un nombre d'équipements en fonctionnement $N(t)$, ce qui conduit à approvisionner le parc d'équipements, si nécessaire.

On se pose le problème suivant : de combien d'équipements faut-il approvisionner le parc à chaque instant ?

Traitons d'abord le problème en discontinu. Soit $R(t)$ la quantité approvisionnée à l'instant $t$, et soit $v(\tau)$ la fiabilité d'un des équipements, c'est-à-dire la probabilité pour que sa durée de vie dépasse $\tau$. On a de façon évidente :

$$N(t) = R(t) + R(t-1)\ v(1) + R(t-2)\ v(2) + R(1)\ v(t-1) + N(0)\ v(t) \qquad (22)$$

formule qui permet de calculer pas à pas $R(1), R(2) \ldots R(t) \ldots$

En effet :

$N(1) = R(1) + N(0)v(1) \to R(1)$

$N(2) = R(2) + R(1)v(1) + N(0)v(1) \to R(2)$

etc.

Si nous nous plaçons dans le cas particulier, très fréquent dans la pratique, où $N(t) = N$ constante, que peut-on dire lorsque $t$ devient suffisamment grand ? Si $\overline{t}$est la durée de vie moyenne de l'équipement, on démontre que

$$\lim_{t \to \infty} R(t) = \frac{N}{\overline{t}}$$

ce qui semble d'ailleurs bien naturel.

On peut d'ailleurs démontrer ce résultat très facilement dans le cas continu. Pour ce cas, appelons $R(t)$ le nombre d'approvisionnements effectués entre $0\ et\ t\ et\ R'(t)$ le nombre d'approvisionnements effectués entre $t\ et\ t+dt$.

L'équivalent de la relation (22) est :

$$N(t) = N(0)\, v(t) + \int_0^t R'(u)\, v(t-u)\, du \qquad (23)$$

Si l'on passe aux transformées de Laplace, on a, avec des notations évidentes :

$$N^*(p) = N(0)\, v^*(p) + R'^*(p)\, v^*(p)$$

soit

$$R'^*(p) = \frac{N^*(p) - N(0)\, v^*(p)}{v^*(p)}$$

Dans le cas particulier où $N(t) = N = constante$

$$R'^*(p) = \frac{N - pN\ v^*(p)}{p\, v^*(p)}$$

On sait que l'on a l'égalité :

$R'(t) \lim_{n\to\infty} pR'^*(p)_{|p|\to 0}$

$$p\, R'^*(p) = \frac{N}{v^*(p)} - p\, N$$

on a :

$$v^*(p) = \int_0^\infty e^{-pt}\ v(t)\, dt$$

et lorsque $p \to 0 \qquad v^*(p) \to \overline{t}$

Donc

$$\lim_{t\to 0} pR'^*(p) = \frac{N}{\overline{t}}$$

et

$$\lim_{t\to\infty} R'(t) = \frac{N}{\overline{t}}$$

## 12.7. STRATEGIES DE REMPLACEMENT

Revenons au problème de renouvellement : un équipement, de fiabilité connue, est mis en fonctionnement à l'instant $t = 0$ ; lorsqu'il tombe en panne, on le remplace immédiatement par un équipement identique, et ainsi de suite.

Lorsque l'équipement tombe en panne, les coûts consécutifs à cet évènement peuvent être très importants, du fait du caractère aléatoire de l'instant de la défaillance.

Aussi, en particulier si $\lambda(t)$ croît beaucoup avec $t$, il peut être intéressant de mettre en place une politique systématique de remplacement des équipements, c'est-à-dire de changer les équipements avant qu'ils ne tombent en panne, et cela à un instant bien déterminé. Il s'agira alors de ce qu'on pourra appeler une politique d'entretien préventif. Nous allons examiner un cas particulier de ces politiques.

### 12.7.1. Une politique d'entretien préventif

La politique la plus simple consiste à se fixer un intervalle de temps $\theta$ :

- si un équipement donné, mis en fonctionnement à $t$, a une défaillance avant $t + \theta$, on le remplace immédiatement : le coût de défaillance est alors $\gamma_r + \Gamma$, où $\gamma_r$ représente le coût du remplacement proprement dit et $\Gamma$ les coûts supplémentaires dûs au caractère aléatoire de la défaillance (perte de production, de clientèle, etc...).
- si l'équipement, au contraire, est toujours en fonctionnement à $\theta$, alors on le remplace à cette date, et le coût encouru n'est alors que $\gamma_r$.

Il s'agit de savoir si cette politique est plus intéressante que celle de laisser-faire, c'est-à-dire celle qui consiste à ne remplacer un équipement que lorsqu'il tombe en panne. Dans ce cas, le coût de chaque défaillance est $\gamma_r + \Gamma$.

Appelons $\bar{t}$ la durée de vie moyenne de l'équipement, et plaçons-nous en régime permanent, sur un intervalle de temps $T$.

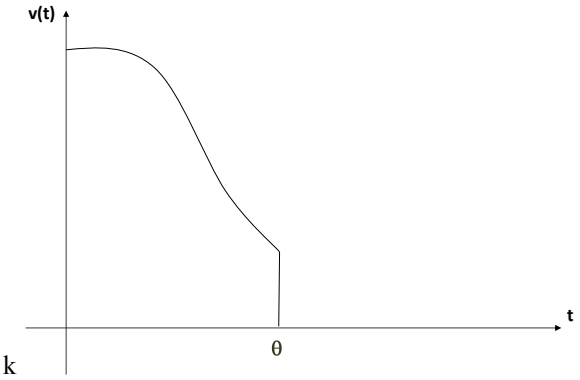
Si l'on choisit la politique du laisser-faire, le nombre de défaillances moyen sur $T$ est $\frac{T}{\bar{t}}$ et le coût moyen total $\frac{T}{\bar{t}} \times (\gamma_r + \Gamma)$.

Par unité de temps, le coût de cette politique est

$$C = \frac{\gamma_r + \Gamma}{\bar{t}} \qquad (25)$$

Le calcul analogue, pour la politique d'entretien préventif, est un peu plus compliqué.

Tout d'abord, quelle est la durée de vie moyenne de l'équipement ? Il faut prendre en compte le fait que la courbe de fiabilité est « tronquée » à partir de $\theta$ :




Les probabilités que la durée de vie de l'équipement soit égale à $\theta$ est $v(\theta)$. Dans ces conditions, la durée de vie moyenne est :

$$\overline{t_\theta} = \int_0^\theta t\, i(t)dt + \theta\, v(\theta)$$

Soit, en intégrant par parties :

$$\overline{t_\theta} = \int_0^\theta v(t)dt \qquad (26)$$

Sur la période $T$ le nombre de remplacements moyens est alors $\frac{T}{\overline{t_\theta}}$. Sur ce nombre, une proportion $v(\theta)$ d'équipements ont été remplacés « systématiquement » avec un coût $\gamma_r$, et une proportion $1 - v(\theta)$ remplacés avec le coût $\gamma_r + \Gamma$. Le coût moyen pour la période $T$ est alors :

$$\frac{T}{\overline{t_\theta}}[\gamma_r\, v(\theta) + (\gamma_r + \Gamma)\,(1 - v(\theta))]$$

Soit

$$\frac{T}{\overline{t_\theta}}[\gamma_r + \Gamma\,(1 - v(\theta))]$$

ou encore, le coût moyen par unité de temps est :

$$C_\theta = \frac{1}{\overline{t_\theta}}[\gamma_r + \Gamma\,(1 - v(\theta))] \qquad (27)$$

Il s'agit donc de comparer $C$ et $C_\theta$ tels qu'ils sont formulés en (25) et (27).

Examinons alors la fonction $C_\theta$.

Pour $\theta = 0$ $C_\theta = \infty$

Pour $\theta = +\infty$ $\overline{t_\theta} = \overline{t}$ $v(\theta) = 0$ $et$ $C_\theta = C$

Par ailleurs, dérivons $C_\theta$

$$\frac{dC_\theta}{d\theta} = \frac{-v'(\theta)\,\overline{t_\theta}\,\Gamma - v(\theta)[\gamma_r + \Gamma(1-v(\theta))]}{t_\theta^{-2}}$$

Pour

$$\theta = 0 \qquad \frac{dC_\theta}{d\theta} < 0$$

Cherchons alors les extremums de $C(\theta)$

$$\frac{dC_\theta}{d\theta} = 0 \rightarrow -v'(\theta)\,\overline{t_\theta}\,\Gamma = v(\theta)[\gamma_r + \Gamma(1-v(\theta))]$$

ce qui donne :

$$\frac{\gamma_r + \Gamma}{\Gamma} = v(\theta) - \overline{t_\theta}\,\frac{v'(\theta)}{v(\theta)}$$

mais

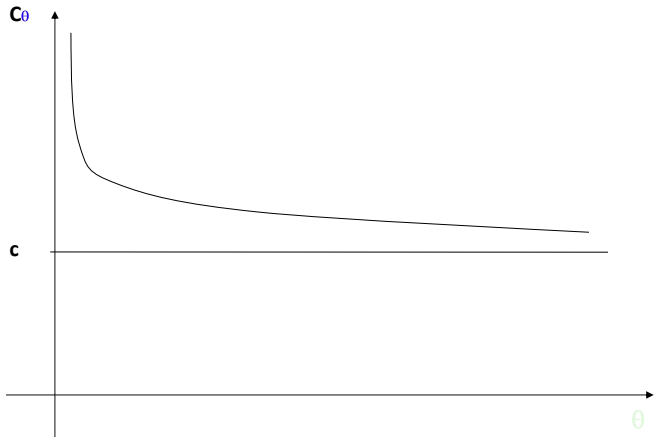
$\frac{v'(\theta)}{v(\theta)} = \lambda(\theta)$ (taux de défaillance)

donc

$$v(\theta) + \overline{t_\theta}\,\lambda(\theta) = \frac{\gamma_r + \Gamma}{\Gamma} \qquad (28)$$

l'équation (28) donne le ou les extremums de $C(\theta)$.

On voit alors que si $\lambda(\theta)$ est décroissant ou constant, il n'y a pas d'extremum pour $C(\theta)$ : en effet, $\overline{t}_\theta = 0$ pour $\theta = 0$ et alors dans ce cas, la fonction $v(\theta) + \overline{t}_\theta\lambda(\theta)$ est constamment égale ou inférieure à 1. Alors, comme $C(\theta)$ est décroissant pour les faibles valeurs de $\theta$, on a la configuration suivante :




Le seul cas intéressant est le cas où $\lambda(\theta)$ est croissant avec θ. Dans ce cas, y a-t-il un extremum unique ? Pour cela, analysons la fonction

$$U(\theta) = v_\theta + \overline{t_\theta}\,\lambda(\theta)$$

Pour

$\theta = 0 \qquad U(\theta) = 1$

Pour $\theta = \infty \qquad U(\theta) = \overline{t}_\theta\,\lambda(\infty)$

Par ailleurs :

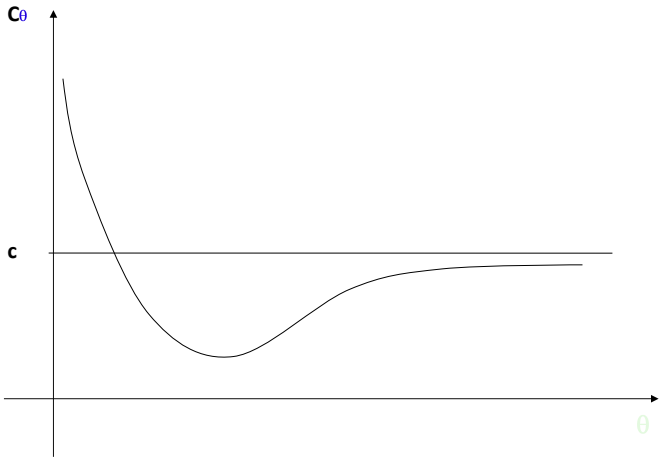
$U'(\theta) = v'(\theta) + v(\theta)\lambda(\theta) + \overline{t}_\theta\lambda'(\theta)$

$U'^{(\theta)} = \overline{t}_\theta\lambda'(\theta) > 0$ puisque l'on a supposé $\lambda(\theta)$ croissant.

On voit alors que la condition nécessaire et suffisante pour que $C_\theta$ ait un extremum est

$$\overline{t}_\theta\lambda(\infty) > \frac{\gamma_r + \Gamma}{\Gamma}$$

et alors, l'extremum est unique : comme $C_\theta$ est décroissante à l'origine, cet extremum est un minimum et l'on a la configuration suivante :



Ce minimum est obtenu par la résolution (parfois compliquée) de l'équation

$$v(\theta) + \bar{t}_\theta \, \lambda(\theta) = \frac{\gamma_r + \Gamma}{\Gamma}$$

Dans beaucoup de cas, on calcule $C_\theta$ systématiquement pour différentes valeurs de $\theta$; on procède souvent à ce calcul en prenant l'optique discontinue.

Si nous reprenons l'exemple du paragraphe $V$ avec $\gamma_r = 1$ et $\Gamma = 3$, on peut, d'une part calculer $C$ :

$$C = \frac{4}{5{,}76} = 0{,}69$$

D'autre part dresser le tableau suivant :

| $\theta$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| v($\theta$) | 0,99 | 0,96 | 0,90 | 0,82 | 0,68 | 0,45 | 0,25 | 0,13 | 0,06 | 0,02 | 0 |
| $\bar{t}_\theta$ | 0,995 | 2,97 | 2,90 | 3,76 | 4,51 | 5,075 | 5,425 | 5,615 | 5,71 | 5,75 | 5,76 |
| $C_\theta$ | 1,04 | 0,515 | 0,45 | 0,41 | 0,434 | 0,52 | 0,60 | 0,64 | 0,668 | 0,685 | 0,69 |

**Remarque** *:* $\bar{t}_\vartheta$ a été calculé par la formule

$$t_\theta = \sum_{t=0}^{\theta-1} v(t) + \frac{1}{2} v(\theta) - \frac{1}{2}$$

tenant compte du fait que les défaillances ont été concentrées aux moitié de périodes.

On voit donc que l'entretien préventif est intéressant ($C_\theta$ comparé à $C = 0{,}69$) pour $2 \leq \theta$ et que la politique optimale est $\theta = 4$ (plus de 40% de gain par rapport au laisser-faire).

### 12.7.2 . Autres politiques de remplacement systématique

Dans certains cas, par raison de commodité, on essaie de remplacer l'équipement de façon régulière, deux visites successives étant séparées par un intervalle de temps fixe $T$. Cela dit, si l'équipement tombe en panne entre deux visites, on le remplace en catastrophe.

On a vu que l'on pouvait calculer le nombre moyen de défaillance $H(T)$ pendant une période $T$ quelconque (voir V. 3). Dans ces conditions, comment évaluer le coût de cette politique ?

Sur une période $T$, en moyenne, il y a :

- un remplacement systématique de coût $\gamma_r$.
- $H(T)$ défaillances, chacune de coût $\Gamma + \gamma_r$

Donc, par unité de temps, le coût de cette politique est :

$$C_T = \frac{\gamma_r + (\gamma_r + \Gamma)\, H(T)}{T}$$

La comparaison de $C_T$ à $C$ (laisser-faire) ou $C_\theta$ (remplacement à l'âge $\theta$) n'est pas toujours facile. Remarquons simplement qu'en général, la politique étudiée (remplacement toutes les périodes $T$) risque fort de ne pas être très intéressante : en effet, il se peut très bien que l'on remplace un équipement alors qu'il vient d'être mis en service.

On peut alors imaginer des stratégies mixtes, tenant compte à la fois des âges des équipements et des commodités des structures mises en place pour les remplacements. Ces développements dépassent le cours de cet exposé.

# Chapitre 13 • Problème de stocks

## INTRODUCTION

Pour des raisons de sécurité, de commodité, ou encore pour des raisons financières, il est très courant que les systèmes de production procèdent à des stockages de matières ou de produits : il s'agit alors de réserves inemployées, momentanément improductives, et qui entraînent des dépenses diverses (surveillance, immobilisation des locaux, détérioration etc.).

Les stocks peuvent être de nature très variée, aussi bien en ce qui concerne l'élément stocké (matière première, liquidités, produits finis, etc.) que la place qu'ils ont dans le système étudié : ils peuvent être en amont du processus de production proprement dit (matières premières ou pièces de rechange pour les machines, par exemple; alors les sorties des stocks sont régies par des commandes internes au système) ou encore en aval (stock de produits finis : alors, les sorties de stocks sont régies par la demande extérieure à l'organisme).

D'une façon générale, un stock est caractérisé par les éléments suivants :

- un ou plusieurs produits,
- une **demande** qui peut être déterministe, mais qui, le plus souvent est aléatoire,
- des **délais de livraison** : délai qui sépare l'instant où une demande arrive au point de stockage et l'instant où les pièces correspondantes sortent effectivement du stock.

Dans tout l'exposé, ces délais seront considérés comme nuls. Cette hypothèse correspond au fait que l'organisme stockeur n'a pas intérêt à retenir le produit stocké, et que ce dernier est sorti le plus vite possible du stock lorsqu'il est demandé. La loi des arrivées des demandes sera confondue alors avec la loi des instants de sorties de stock.

- des **délais de réapprovisionnement** : lorsqu'on veut augmenter le stock, l'approvisionnement commandé est rarement instantané; le délai correspondant est le plus souvent aléatoire.
- des **coûts de stockage :** ces coûts (surveillance, locaux, capital immobilisé, etc.) croissent avec la quantité de matières stockées et le temps pendant lequel il y a stockage.
- des **coûts de réapprovisionnement** ou **coûts de lancement :** ils contiennent (ou peuvent contenir, car on peut les exclure la plupart du temps) les dépenses d'achats, mais, et c'est le fait important, des dépenses de lancement de commandes, qui croissent d'avantage en fonction du nombre de commandes

effectuées qu'en fonction de la quantité totale approvisionnée sur une certaine période de temps.
- des **coûts de rupture de stock,** qui doivent être imputées lorsque l'organisme ne peut faire face à une demande. Ces coûts comprennent des coûts de réapprovisionnement « en catastrophe » mais également des coûts dûs à des pertes de production ou de clientèle.

Une politique de stock consiste alors à se poser les questions suivantes : compte tenu des éléments précédents supposés connus :

- quand faut-il réapprovisionner le stock ?
- quelle quantité faut-il commander ?

Si l'on prend en charge, en vue de la formalisation d'un phénomène réel, tous les éléments précédents, ainsi que leur aspect généralement aléatoire, on se heurte la plupart du temps à de très épineuses difficultés méthodologiques. Nous n'étudierons ici que des modèles fort simples, en faisant tout d'abord la remarque suivante :

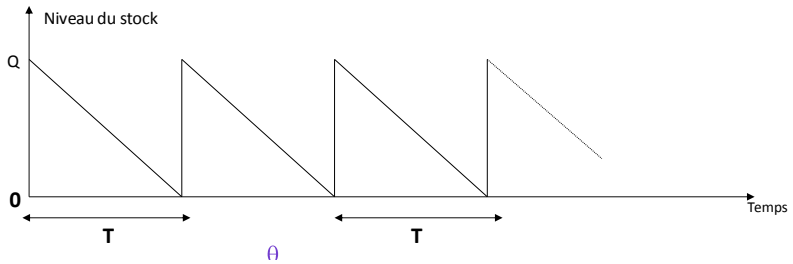
Si l'on ne tient pas compte - pour le moment - des coûts de rupture, il convient de souligner l'alternative suivante :

On peut vouloir diminuer les coûts de stockage en maintenant le stock à un faible niveau; mais cela correspond alors à une politique de réapprovisionnements nombreux, pour faire face a la demande, et les coûts de lancement augmentent. En revanche, une politique de réapprovisionnements peu nombreux, mais massifs, diminuent les frais de lancements, mais augmentent les frais de stockage. Il y a donc un équilibre à trouver entre coûts de stockage et coûts de lancement. C'est sur cette idée que se fonde le modèle de stock le plus ancien (il date de 1929 ) , modèle dit « formule de Wilson », qui , étant déterministe, se situe quelque peu en dehors du champ de cette partie, mais qu'il est nécessaire d'exposer, ne serait-ce qu'à cause de sa notoriété.

## 13.1. FORMULE DE WILSON

Le modèle de Wilson est caractérisé par les éléments suivants :

- un seul produit,
- demande déterministe, constante dans le temps, mesurée par un flux $d$ (quantité du produit stocké demandée par unité de temps),
- délais de réapprovisionnement connus (non aléatoires),
- coûts de lancement $C_l$ par commande indépendante de la quantité commandée,
- coûts de stockage proportionnel au temps et à la quantité stockée, c'est-à-dire que l'on se donne $C_s$ : coûts de stockage par unité stockée et par unité de temps,
- rupture de stock interdite.




Dans ces conditions, on veut réapprovisionner le stock à période fixe, correspondant à un intervalle de temps $T$ et d'une quantité fixe $Q$. Il s'agit donc de calculer $Q$ et T, de façon que le coût total (lancement + stockage) soit minimal.

En effet, le réapprovisionnement étant instantané et la demande déterministe, il est évident qu'une commande doit avoir lieu lorsque le stock s'annule.

On a d'abord évidemment :

$$dT = Q \qquad (1)$$

$d$ étant la demande par unité de temps.

Ensuite, le coût pour une période $T$ se décompose de la façon suivante :

- le coût de lancement $C_l$
- le coût de stockage, qui est : $C_s \frac{Q}{2} T$

Par ailleurs, si $\Theta$ est la durée totale traitée, il y a $n = \frac{\Theta}{T}$ commandes ou périodes. Donc, le coût total est :

$$C = \frac{\Theta}{T}\left[ c_\ell + c_s \frac{Q}{2} T \right] \qquad (2)$$

(1) et (2) donnent :

$$C = \Theta \frac{d}{Q}\left[ c_\ell + c_s \frac{Q^2}{2d} \right]$$

soit

$$C = \Theta\, d\left[ \frac{C_\ell}{Q} + c_s \frac{Q}{2d} \right] \qquad (3)$$

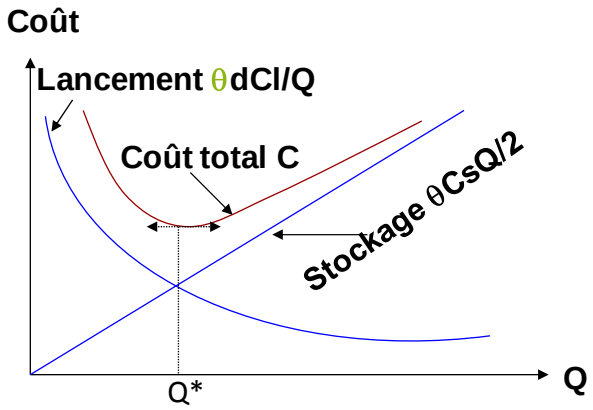
le minimum de $C$ est obtenu pour :

$$-\frac{c_\ell}{Q^2}+\frac{c_s}{2d}=0$$

soit

$$Q^*=\sqrt{2d\,\frac{c_\ell}{c_s}} \qquad (4)$$

Si l'on représente la variation des deux coûts (lancement + stockage) en fonction de $Q$, on a la figure suivante, très classique :




($Q^*$ est trouvé à l'intersection de la droite et de l'hyperbole)

On a aussi les résultats :

$$T^*=\frac{Q^*}{d}=\sqrt{\frac{2}{d}\,\frac{c_\ell}{c_s}}$$

$$n^*=\frac{\Theta}{T^*}=\Theta\sqrt{\frac{d}{2}\,\frac{c_s}{c_\ell}} \qquad (5)$$

$$C^*=\sqrt{2\Theta^2\,d\,c_s\,c_\ell}$$

Ce modèle est extraordinairement fruste. Il a été enrichi par quelques variantes telles que

- introduction de délais de réapprovisionnement (fixes),
- actualisation,

- commandes groupées pour plusieurs articles,
- introduction de prix d'achats dégressifs,
- contraintes de capacités de stockage etc.

Mais il est évident que, même muni de ces raffinements, le modèle de Wilson reste indigent. En particulier, l'hypothèse du déterminisme de la demande est en général assez peu compatible avec les phénomènes réels. Aussi, allons-nous examiner quelques modèles simples où l'aléatoire est pris en charge. Par ailleurs, il sera évidemment nécessaire d'envisager des situations de rupture de stock.

## 13.2. MODELES DE STOCK EN UNIVERS ALEATOIRE

Nous allons d'abord prendre un modèle simple où les instants de réapprovisionnement sont connus et où seule la quantité à commander $Q$ est inconnue.

### 13.2.1. Période de réapprovisionnement connue

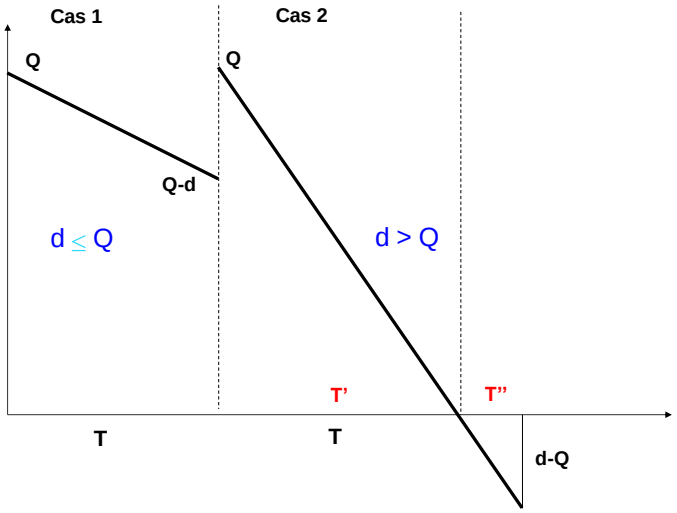
Soit donc le cas suivant :

- un seul produit,
- une demande aléatoire, de loi de probabilité $p(d)$ connue pour la période, $T$ séparant deux approvisionnements successifs. $p(d) = p[demande = d]$,
- délai d'approvisionnement connu (non aléatoire),
- coût de lancement $C_l$ par commande,
- coût de stockage $C_s$ par unité de produit stocké et par unité de temps,
- coût de rupture ou de pénurie de stock $C_p$ par unité manquante et par unité de temps.

À la fin de chaque période $T$, on choisit d'approvisionner de telle façon que le stock soit recomplété à une quantité fixe. Pour calculer $Q$, on choisit alors de minimiser l'espérance mathématique de coût pour une période quelconque.

Calculons cette espérance mathématique.

On a deux situations possibles, suivant que l'on a rupture de stock ou non, résumées par la figure ci-dessous, où l'on a fait par ailleurs l'hypothèse que le stock décroissait linéairement sur les périodes $T$ :




Dans le premier cas $(d \leq Q)$, on a les coûts suivants :

- coût de lancement $C_l$[16]
- coût de stockage; le stock moyen étant pendant la période $T$, $Q - \frac{d}{2}$, le coût de stockage moyen est $(Q - \frac{d}{2})C_s T$
- coût de pénurie : 0

Dans le second cas $(d > Q)$, on a les coûts :

- coût de lancement $C$
- coût de stockage : il est facile de voir qu'il y a eu stockage pendant le temps
- $T' = T\frac{Q}{2}$, le stock moyen étant égal sur cet intervalle de temps à $\frac{Q}{2}$, donc le coût de stockage à $C_s \frac{Q^2}{2d} T$.
- coût de rupture de stock : il y a eu rupture pendant un temps $T'' = \frac{d-Q}{d}\ T$, et la rupture moyenne, en quantité est $\frac{d-Q}{2}$, Donc le coût de rupture est :

$$c_p \frac{(d-Q)^2}{2d} T$$

16 Si l'on voulait être rigoureux, il faudrait faire intervenir l'évènement : demande nulle sur la période précédente, qui éviterait alors de procéder à un lancement. Nous supposerons que les frais fixes de lancement sont fatals, même pour des quantités commandées nulles.

Le premier cas se produit pour $d \leq Q$, le second pour $d > Q$. Dans ces conditions, l'espérance mathématique de coût peut s'écrire :

$$E(c) = C_l + \sum_{d=0}^{Q} \left(Q - \frac{d}{2}\right) C_s T p(d) + \sum_{d=Q+1}^{\infty} \frac{Q^2}{2d} T p(d) + \sum_{d=Q+1}^{\infty} \frac{(d-Q)^2}{2d} C_p T p(d) \quad (6)$$

Si l'on veut minimiser $E(C)$, on voit que $T$ et $C_l$ n'interviennent plus, et que $Q$ est trouvé par la formule complexe :

$$Min \left[ \sum_{d=0}^{Q} \left(Q - \frac{d}{2}\right) C_s p(d) + \sum_{d=Q+1}^{\infty} \left( \frac{Q^2}{2d} C_s + \frac{(d-Q)^2}{2d} C_p \right) p(d) \right] Q$$

**On peut minimiser cette quantité par tâtonnement.** Dans beaucoup de cas (quantité **en** cause importante), on peut passer au cas continu . On a alors à minimiser la quantité, si $f(d)$est la loi de densité de la demande :

$$g(Q) = c_s \int_0^Q f(u)\, du \left(Q - \frac{u}{2}\right) f + \int_0^Q \left( \frac{Q^2}{2u} c_s + \frac{(u-Q)^2}{2u} c_p \right) f(u)\, du$$

fonction qui atteint un extremum pour $Q$ tel que

$$c_s \int_0^Q f(u)\, du + c_s\, Q\, f(Q) - c_s \frac{Q}{2} f(Q) + \int_Q^\infty \frac{Q}{u} c_s\, f(u)\, du - c_s \frac{Q}{2} f(Q)$$

$$- \int_Q^\infty \left( \frac{u-Q}{u} \right) c_p\, f(u)\, du = 0$$

Soit

$$c_s \int_0^Q f(u)\, du + c_s \int_Q^\infty \frac{Q}{u}\, f(u) du - c_p \int_Q^\infty \frac{u-Q}{u}\, f(u)\, du = 0$$

ou encore :

$$(c_s + c_p) \int_0^Q f(u)\, du + (c_s + c_p)\, Q \int_Q^\infty \frac{f(u)}{u}\, du = c_p$$

$$F(Q) + Q \int_Q^\infty \frac{f(u)}{u}\, du = \frac{c_p}{c_s + c_p} \quad (7)$$

La résolution de l'équation (7) n'est pas toujours facile. Lorsque l'on est dans le cas discret, on calcule en général la fonction :

$$g(Q) = \sum_{d=0}^{Q} p(d) + Q \sum_{d=Q+1}^{\infty} \frac{p(d)}{d}$$

et l'optimum est donné par $Q^*$ tel que :

$$g(Q^* - 1) < \frac{c_p}{c_p + c_s} < g\,(Q^*)$$

**Remarque très importante :** la valeur optimale $Q^*$ ainsi calculée n'est valable que si le nombre de périodes pendant lequel se déroule le processus est important.

S'il n'y avait qu'une seule période, on pourrait également prendre le critère de l'espérance mathématique, mais il serait totalement arbitraire : le coût réel pourrait être à la fin de la période, très différent de $E(C)$.

Par contre, si le nombre de périodes est important, la loi des grands nombres nous assure que le coût moyen par période sera bien $E(C)$, le plus faible possible.

Cela dit, il ne faut pas se cacher que le critère « espérance mathématique » reste criticable pour un grand nombre de périodes. Si le coût moyen est certes peu différent de $E(C)$, il se peut fort bien que le coût total réel soit très supérieur à $nE(C)$, si $n$ est le nombre de périodes. Pour s'en assurer, il suffit d'appliquer le théorème central limite : si par période, l'espérance mathématique du coût est $E(C)$ et l'écart type $\sigma_c$, le coût total $\Gamma$ sur $n$ périodes, d'après ce théorème, est une variable aléatoire d'espérance $nE(C)$ et d'écart type $\sqrt{n}\sigma_c$ (donc croissant avec $n$).

Dans ces conditions, le risque pour que $\Gamma$ dépasse le cadre de certaines limites de tolérance peut ne pas être négligeable. Nous entrons là dans un débat qui dépasse le cadre de ce cours.

### 13.2.2. Période de réapprovisionnement inconnue

Nous gardons les mêmes hypothèses qu'en III.I, sauf que nous sommes obligés d'introduire une loi de probabilité $p(T, d)$, $T$ étant la période que l'on cherche (on cherche également $Q$, stock en début de période). Dans ces conditions, la formule (6) reste valable pour une période, à condition de remplacer $p(d)$ par $p(T, d)$. Sur une durée totale $\Theta$ le coût est alors :

$$C(\Theta) = \frac{\Theta}{T} E(C)$$

Soit :

$$C(\Theta) = \Theta\left[ c_\rho + \sum_{d=0}^{Q}\left( Q - \frac{d}{2} \right) c_s\, p(T,d) + \sum_{d=Q+1}^{\infty} \frac{Q^2}{2d}\, c_s\; p(T,d) + \sum_{d=Q+1}^{\infty} \frac{(d-Q)^2}{2d} c_p\, p(T,d) \right] \quad (8)$$

On trouve $Q$ et $T$ optimaux par $Min_{Q,T}\, C(\Theta)$.

On procède en général de la façon suivante : on se fixe d'abord $T_0$ période au-dessous de laquelle on ne veut pas descendre et on calcule (par un ajustement statistique) la distribution de probabilités $p(T_0, d)$. On recherche $Q_0^*$ optimal pour ce $T_0$, par la méthode exposée au III.1. Ensuite, on essaye les périodes $T_1 = 2T_0\,, T_2 = 3T_0 \ldots KT_0$ ...en calculant les nouvelles distributions de la demande par la formule de récurrence :

$$p(K\,T_0, d) = \sum_{i=0}^{d} p[(K-1)\,T_0, i]\; p[T_0, d-i]$$

(On suppose ici que les demandes successives sont indépendantes, ce que l'on faisait d'ailleurs avant implicitement).

Enfin, on prend le couple $(Q^*, T^*)$, minimum minimorum.

Il convient de souligner à nouveau que ce type de calcul n'est valable que :

- pour un grand nombre de périodes,
- pour des données identiques pour chaque période,
- pour une indépendance totale entre deux périodes successives (le stock en fin de période n'intervient pas dans le calcul de la période suivante).

Il se peut évidemment fort bien que ces hypothèses ne puissent être acceptées : dans ce cas, il faut recourir à des modèles plus élaborés. Souvent ces derniers font appel à la programmation dynamique.

### 13.2.3 . Application de la programmation dynamique

Pour montrer comment le principe d'optimalité peut s'appliquer aux problèmes de stocks, nous allons traiter un exemple qui est d'ailleurs assez voisin de celui traité au chapitre I (programmation dynamique discrète aléatoire).

Une machine d'un certain type est demandée suivant la loi de probabilité suivante, concernant un trimestre :

| D | 0 | 1 | 2 | 3 | >3 |
|---|---|---|---|---|---|
| p(d) | 0,1 | 0,2 | 0,5 | 0,2 | 0 |

$d$ est la demande trimestrielle.

$p(d)$la probabilité de cette demande.

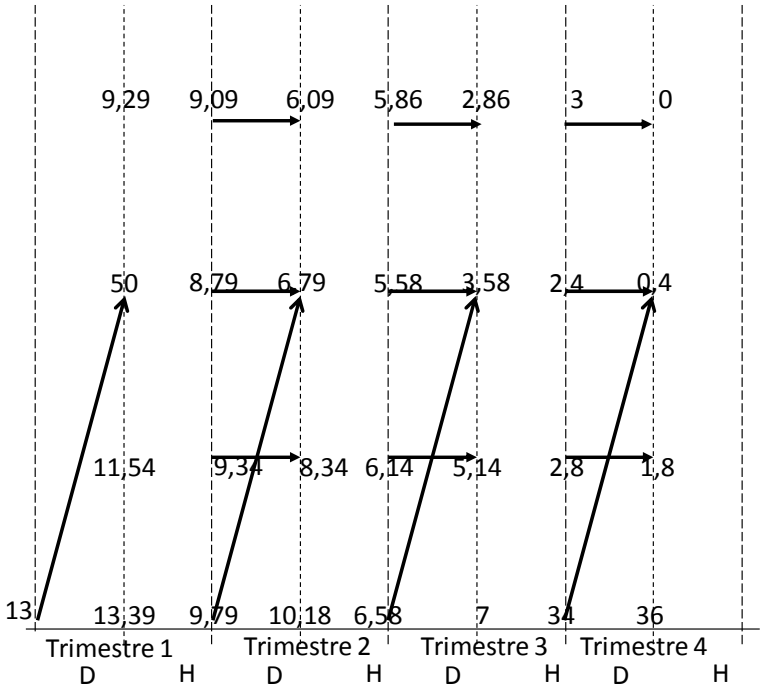
Les frais fixes de réapprovisionnement en machines sont $\gamma_l = 1$ (dans une unité qu'il est inutile de spécifier).

Les frais de stockage sont $\gamma_s = 1$ /machine /trimestre et les coûts de pénurie sont évalués à $\gamma_p = 2$ /demande insatisfaite

L'approvisionnement a lieu au début d'un trimestre, sans délai. Les commandes arrivant au cours d'un trimestre sont livrées à la fin de ce même trimestre.

On se demande quelle est la politique optimale sur un an. Au début de l'année, le stock est nul.

On peut alors diviser l'année en 4 phases, correspondant à chaque trimestre. En début de chaque phase, une décision doit être prise, concernant le nombre de machines à approvisionner. Ensuite, le hasard joue par l'intermédiaire de la loi de la demande. On a un processus DH, représenté ci-dessous :




Nous allons donc déterminer la stratégie à adopter, en appliquant, comme nous l'avons fait au chapitre I, le principe d'optimalité.

Il est tout d'abord évident que le stock de machines n'a jamais à dépasser 3. Nous avons donc les quatre états possibles : $0, 1, 2\ et\ 3$ machines en stock. $(E_0, E_1, E_2, E_3)$.

Plaçons-nous alors en phase 4 (on sait que l'on remonte toujours le temps en programmation dynamique aléatoire) et calculons l'espérance mathématique en chaque état pour la phase hasard de la phase 4 $(H_4)$. C'est le coût de pénurie qui intervient à ce niveau.

**$H_4$**

$E_3$ : $espérance\ coût\ de\ pénurie\ = 0$

$E_2$ : $espérance\ coût\ de\ pénurie\ = 2 \times 0{,}2 = 0{,}4$

$E_1 espérance\ coût\ de\ pénurie = 0{,}5 \times 2 + 0{,}2 \times 4 = 1{,}8$

$E_0\ espérance\ coût\ de\ pénurie = 0{,}2 \times 2 + 4 \times 0{,}5 + 6 \times 0{,}2 = 3{,}6$

(ces chiffres sont reportés sur le graphique)

Quelles décisions doit-on prendre alors en début de la phase 4 (phase $D_4$) ?

On a les transactions possibles suivantes, avec leurs coûts associés; et les politiques optimales correspondant aux coûts encadrés.

**$D_4$**

$E_3 \rightarrow E_3\ coût = 0 + 3 + 0 = 3$ (lancement + stockage + espérance mathématique de l'état d'arrivée)

$E_2 \rightarrow E_3\ coût = 1 + 3 + 0 = 4$

$E_2 \rightarrow E_2\ coût = 0 + 2 + 0{,}4 = \boxed{2{,}4}$

$E_1 \rightarrow E_3\ coût = 1 + 3 + 0 = 4$

$E_1 \rightarrow E_2\ coût = 1 + 2 + 0{,}4 = 3{,}4$

$E_1 \rightarrow E_1\ coût = 0 + 1 + 1{,}8 = \boxed{2{,}8}$

$E_0 \rightarrow E_3\ coût = 1 + 3 + 0 = 4$

$E_0 \rightarrow E_2\ coût = 1 + 2 + 0{,}4 = \boxed{3{,}4}$

$E_0 \rightarrow E_1\ coût = 1 + 1 + 1{,}8 = 3{,}8$

$E_0 \rightarrow E_0\ coût = 0 + 0 + 3{,}6 = 3{,}6$

La stratégie optimale étant calculée pour la phase 4, on passe à la phase 3, et tout d'abord à la phase hasard de la phase 3 $(H_3)$.

Pour cela, il suffit d'ajouter à l'espérance mathématique des coûts de pénurie pour la phase 3, l'espérance des coûts calculés en $D_4$, soit

**$H_3$**

$E_3$: $0 + 0{,}1 \times 3 + 2{,}4 + 0{,}2 \times 2{,}8 \times 0{,}5 + 0{,}2 \times 3{,}4 = 2{,}86$

$E_2$: $0{,}4 + 0{,}1 \times 2{,}4 + 0{,}2 \times 2{,}8 + 0{,}7 \times 3{,}4 = 3{,}58$

$E_1$: $1{,}8 + 0{,}1 \times 2{,}8 + 0{,}9 \times 3{,}4 = 5{,}14$

$E_0$: $3{,}4 + 3{,}6 = 7$

À partir de là, sans commentaires, la phase décision de la phase 3 ($D_3$)

**$D_3$**

$E_3 \to E_3$ $\quad 0 + 3 + 2{,}86 = \boxed{5{,}86}$

$E_2 \to E_3$ $\quad 1 + 3 + 2{,}86 = 6{,}86$

$E_2 \to E_2$ $\quad 0 + 2 + 3{,}58 = \boxed{5{,}58}$

$E_1 \to E_3$ $\quad 1 + 3 + 2{,}86 = 6{,}86$

$E_1 \to E_2$ $\quad 1 + 2 + 3{,}58 = 6{,}58$

$E_1 \to E_1$ $\quad 0 + 1 + 5{,}14 = \boxed{6{,}14}$

$E_0 \to E_3$ $\quad 1 + 3 + 2{,}86 = 6{,}86$

$E_0 \to E_2$ $\quad 1 + 2 + 3{,}58 = \boxed{6{,}58}$

$E_0 \to E_1$ $\quad 1 + 1 + 5{,}14 = 7{,}14$

$E_0 \to E_0$ $\quad 0 + 0 + 7 = 7$

D'où la stratégie optimale pour la phase 3.

***Phase 2***

**$H_2$**

$E_3$ $\quad 0 + 0{,}1 \times 5{,}86 + 0{,}2 \times 5{,}58 + 0{,}15 \times 6{,}14 + 0{,}2 \times 6{,}58 = 6{,}09$

$E_2$ $\quad 0{,}4 + 0{,}1 \times 5{,}58 + 0{,}2 \times 6{,}14 + 0{,}7 \times 6{,}58 = 6{,}79$

$E_1$ $\quad 1{,}8 + 0{,}1 \times 6{,}14 + 0{,}9 \times 6{,}58 = 8{,}34$

$E_0$ $\quad 3{,}6 + 6{,}58 = 10{,}18$

**$D_2$**

$E_3 \to E_3$ $\quad 0 + 3 + 6{,}09 = \boxed{9{,}09}$

$E_2 \to E_3$ $\quad 1 + 3 + 6{,}09 = 10{,}09$

$E_2 \to E_2$ $\quad 0 + 2 + 6{,}79 = \boxed{8{,}79}$

$E_1 \to E_3$ $\quad 1 + 3 + 6{,}09 = 10{,}09$

$E_1 \to E_2$ $\quad 1 + 2 + 6{,}79 = 9{,}79$

$E_1 \to E_1 \qquad 0 + 1 + 8{,}34 = \boxed{9{,}34}$

$E_0 \to E_3 \qquad 1 + 3 + 6{,}09 = 10{,}09$

$E_0 \to E_2 \qquad 1 + 2 + 6{,}79 = \boxed{9{,}79}$

$E_0 \to E_1 \qquad 1 + 1 + 8{,}34 = 10{,}34$

$E_0 \to E_0 \qquad 0 + 0 + 10{,}18 = 10{,}18$

***Pour la phase 1***

$\boldsymbol{H_1}$

$E_3 \quad 0 + 0{,}1 \times 9{,}09 + 0{,}2 \times 8{,}79 + 0{,}5 \times 9{,}34 + 0{,}2 \times 9{,}79 = 9{,}28$

$E_2 \quad 0{,}4 + 0{,}1 \times 8{,}79 + 0{,}2 \times 9{,}34 + 0{,}7 \times 9{,}79 = 10{,}00$

$E_1 \quad 1{,}8 + 0{,}1 \times 9{,}34 + 0{,}9 \times 9{,}79 = 11{,}54$

$E_0 \quad 3{,}6 + 9{,}79 = 13{,}39$

$\boldsymbol{D_1}$

$E_0 \to E_3 \qquad 1 + 3 + 9{,}29 = 13{,}29$

$E_0 \to E_2 \qquad 1 + 2 + 10 = \boxed{13}$

$E_0 \to E_1 \qquad 1 + 1 + 11{,}54 = 13{,}54$

$E_0 \to E_0 \qquad 0 + 0 + 13{,}39 = 13{,}39$

La stratégie optimale est donc la suivante :

- au début du trimestre 1, stocker 2 machines,
- au début des autres trimestres, ne pas s'approvisionner si l'on a 3, 2 ou 1 machines; stocker 2 machines si le stock est nul.

Le petit exemple que nous venons de traiter est évidemment très simple. Cela dit, le principe d'optimalité peut s'appliquer à des problèmes de stocks assez complexes, notamment à ceux qui bénéficient des hypothèses suivantes :

- un seul produit,
- plusieurs périodes de durée fixe,
- délai de réapprovisionnement nul.

Par contre, les coûts de lancement, de stockage et de rupture de stock peuvent avoir des formes très diverses (et en particulier varier avec le temps), ce qui n'est pas le cas ici.
L'utilisation de la programmation dynamique pour résoudre un problème de stock nous amène à faire trois remarques très importantes :

1) Lorsque le phénomène étudié est stationnaire (coûts et lois de probabilités constants avec le temps) et que le nombre de périodes traitées est important, on trouve d'après ce

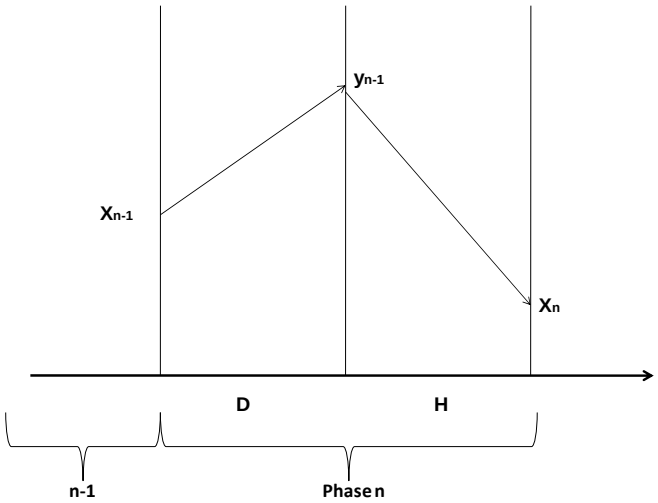
qu'on a vu sur la programmation dynamique une stratégie permanente, c'est-à-dire ne dépendant pas de la période considérée

C'est le cas du petit exemple que nous avons analysé ci-dessus et nul doute que la stratégie permanente que l'on trouverait serait celle que l'on a trouvée pour les périodes $2,3\ et\ 4$.

2) La programmation dynamique se prête fort mal à des délais de réapprovisionnement non nuls. En effet, lorsque l'on se trouve en début d'une phase $t$, les stocks existant dépendent alors des décisions prises en $t-1, t-2$ etc. Or, la programmation dynamique remontant le temps, ces instants n'ont pas été examinés.

3) La programmation dynamique, à un niveau théorique, peut servir de justification à des pratiques fort courantes en matière de stock :

- plaçons-nous en effet dans le cas où les quantités de produit stocké sont assez importantes pour qu'on puisse les considérer comme continues. Dans ces conditions, soit le stock résiduel $x_{n-1}$ à la fin de la période $n-1$. Au début de la période $n$, on décide d'approvisionner à $y_n$.

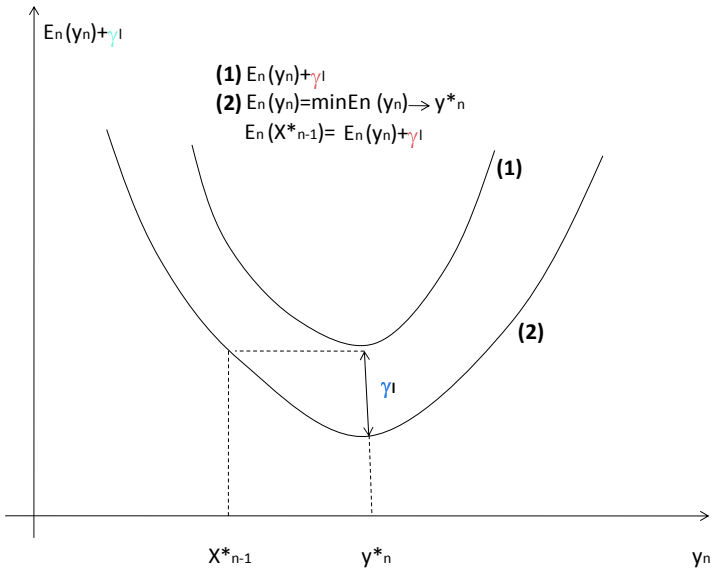



$y_n \geq x_{n-1}$. Par ailleurs, appelons $E_n(y_n)$ l'espérance mathématique du coût total lorsque l'on réapprovisionne à $y_n$ (les coûts de stockage sur la période $n$ sont intégrés dans $E_n(y_n)$. Dans ces conditions, si l'on appelle $\gamma_l$le coût de lancement d'un réapprovisionnement on a , si $E_{n-1}(x_{n-1})$ est l'espérance en fin de période $n-1$.

$$E_{n-1}(x_{n-1}) = E_n(y_n) + \gamma_l \quad si \quad y_n > x_{n-1}$$

$$E_{n-1}(x_{n-1}) = E_n(y_n) \quad si \quad y_n > x_{n-1}$$

Cela dit, on cherche le minimum de $E_{n-1}(x_{n-1})$. En général $E_n(y_n)$ est une fonction convexe de $y_n$. On a alors la figure suivante :




On voit que l'on peut avoir trois situations :

1) $x_{n-1} > y_n^*$ : comme on ne peut effectuer de réapprovisionnement négatif (hypothèse que l'on a faite implicitement dès le départ), on a intérêt à rester en $x_{n-1}$. Réapprovisionnement nul.

2) $x_{n-1}^* \leq x_{n-1} \leq y_n^*$. Là non plus, on n'a pas intérêt à effectuer un réapprovisionnement, car sur cet intervalle $E_n(y_n^*) + \gamma_l > E_n(x_{n-1})$

3) $x_{n-1} < x_{n-1}^*$.
Alors, on a intérêt à réapprovisionner et le stock optimal est alors $y_n^*$.

Donc, la stratégie obtenue est : approvisionner à $y_n^*$ si le stock résiduel est inférieur à une certaine quantité $x_{n-1}^*$, ne pas réapprovisionner si le stock résiduel est supérieur à cette même quantité.

Si le phénomène est stationnaire et le nombre de périodes important, on trouve alors une stratégie du type dit $s - S$ :

- approvisionner à $S$ si le stock résiduel est inférieur à $s$ ; ne pas approvisionner dans le cas contraire.

Ce type de stratégie est tout à fait habituel. A tel point que très souvent, la formalisation d'un problème de stock, au lieu de passer par la recherche de stratégie optimale la plus générale, qui constitue le plus souvent un casse-tête inextricable, est fondée sur la

recherche des meilleures stratégies $s - S$ : on obtient ainsi des solutions sous optimales, mais dont on peut penser en général qu'elles ne sont pas éloignées des solutions optimales.

Cela dit, si l'on examine comment s'énoncent ces stratégies $s - S$, on peut se poser une question : puisque l'on ne réapprovisionne que lorsque le stock résiduel est inférieur à une quantité $s$, pourquoi ne pas attendre que le stock atteigne de lui-même cette quantité pour lancer une commande de réapprovisionnement ?

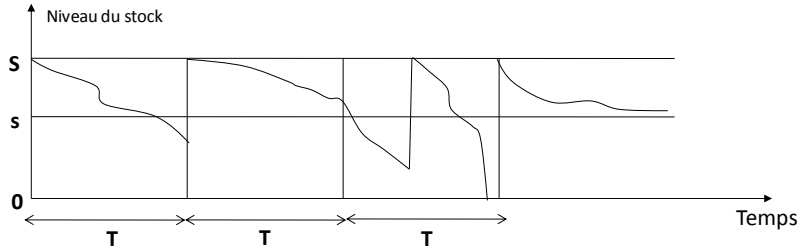
Cette question introduit à l'examen rapide des deux grands types de politique étudiés le plus souvent.

## 13.3. DEUX GRANDS TYPES DE POLITIQUES DE STOCKS

### 13.3.1. Politiques à périodes fixes ou sur calendriers

Cette première catégorie de politique est celle que nous avons examinée jusqu'ici : elle consiste à passer une commande de réapprovisionnement régulièrement, de façon à recompléter le stock à une quantité fixe donnée $S$. Les quantités commandées sont donc variables. Par ailleurs, il peut y avoir (en particulier lorsqu'il y a des coûts fixes de lancement) une quantité $s$ telle que si, en début de période, le stock est supérieur à $s$, il n'est pas intéressant de lancer une nouvelle commande.

L'évolution d'un stock géré sur la base de cette politique peut donc se figurer de la façon suivante, si l'on suppose nuls les délais de réapprovisionnement :




$s, S$ ainsi que la période $T$ peuvent faire l'objet de calculs analogues à ceux que l'on a effectués précédemment.

On peut également introduire un autre élément dans cette politique, qui est le stock de sécurité. En effet, comme nous l'avons fait précédemment, les calculs sur $s, S$ $et$ $T$ font intervenir un coût de pénurie, en général élevé. Mais ce coût est en général très mal connu (sa définition même peut poser problème). Par ailleurs, le gestionnaire du stock peut hésiter à appliquer le critère de l'espérance mathématique, qui lui assure un gain moyen, mais qui peut conduire à des risques non négligeables de pénurie.

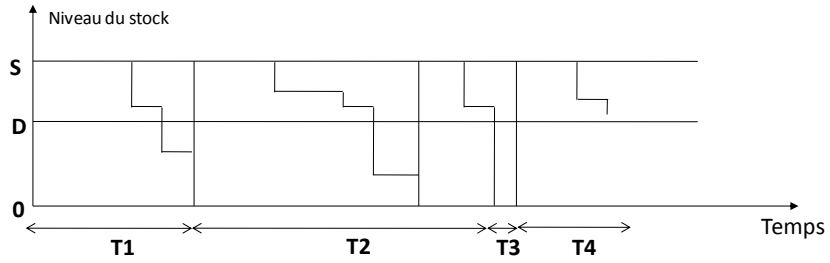
C'est pourquoi, on peut procéder de la façon suivante : on ajoute à $S$ une réserve $R$, dite stock de sécurité, telle qu'il n'existe qu'une probabilité faible que la demande $D$ sur une période dépasse $S + R$, soit

$$P[D > S + R] = \alpha$$

avec $\alpha$ faible ($0{,}001\ par\ exemple$).

### 13.3.2. Politiques à périodes variables ou sur point de commande

Dans ces politiques, on se définit également deux quantités $s\ et\ S$, mais, à la différence des politiques précédentes, le réapprovisionnement ne se fait pas de façon régulière : on laisse le stock baisser jusqu'à ce qu'il tombe en dessous de $s$ et on commande alors $S - stock\ résiduel$. Une telle politique peut alors se figurer de la façon suivante, dans le cas de demandes discrètes :




Sur cette figure, le délai de réapprovisionnement est nul. Si c'est le cas, et si les quantités en cause autorisent des calculs du type continu, une telle politique prémunit bien contre la pénurie (dans le cas continu, théoriquement, on a intérêt à faire $s = 0$). Mais il s'agit là de configuration d'école.

Dans la plupart des cas, non seulement les délais de réapprovisionnement existent, mais ils sont aléatoires. Alors, ils doivent être introduits dans les calculs, en particulier pour les définitions de la réserve de sécurité, qui doit être prévue également pour les politiques sur point de commande. Le calcul de $s$, quantité du stock qui déclenche la commande, est alors fondé en général sur la considération suivante :

Si $d$ est le délai de réapprovisionnement prévu, la prévision de la demande pendant l'intervalle de temps $d$ doit être égale à $s$ (ou plutôt $s - r$, si $r$ est la réserve de sécurité). Ce type de calcul pose le problème de la prévision de la demande et du délai de réapprovisionnement, ce qui renvoie au calcul statistique et plus particulièrement à un chapitre des statistiques assez délicat (analyse des séries chronologiques).

D'une façon générale, les problèmes de stocks deviennent très vite difficiles à formaliser si l'on prend en charge tous les éléments qui interviennent ainsi que leur caractère aléatoire. Là encore, la simulation peut permettre de surmonter l'écueil d'une modélisation inextricable : si l'on connait bien les lois de probabilités en présence (ce qui n'est pas évident en général), on peut reproduire artificiellement, grâce en général à un ordinateur, une séquence d'évènements possibles, et tirer des résultats observés des conclusions en « moyenne » si cette séquence est suffisamment longue. On peut alors

tester et comparer entre elles différentes solutions envisageables (politiques à période fixe, à période variable, niveau de sécurité etc.)

Signalons enfin qu'une des difficultés importantes couramment rencontrée lors d'une modélisation d'un phénomène de stock est la définition et le calcul des éléments de coût que l'on doit prendre en compte. Les limitations à la « prise de décision scientifique » dues à la faiblesse relative de l'outil mathématique ne sont pas les seules.

# BIBLIOGRAPHIE


Berge, C. *« Graphes et hypergraphes* » Dunod 1970

Charon, I Germa, A. Hudry, O. « *Méthodes d'optimisation combinatoire* », Masson 1996

Dantzig, G.B. « *Application et prolongements de la programmation linéaire* », Dunod 1966

Desbazeille G. « *Exercices et problèmes de recherche opérationnelle*, Dunod, 1964.

Faure, R. « *Précis de Recherche Opérationnelle »*, Dunod 1996

Faure, R. « *Précis de Recherche Opérationnelle* », Dunod 1996

Faure, R. Lemaire**,** B**.,** Picouleau, Ch, « Précis *de recherche opérationnelle » : Méthodes et exercices »,* Dunod, 2004.

Gondran, M. et Minoux,M. « *Graphes et algorithmes* », Eyrolles 1979

Gueret, Ch, Prins Ch , Sevaux M. « *Programmation linéaire* », Eyrolles, 2000.

Hêche , JF, Liebling ,Th, de Werra D., *« Recherche opérationnelle pour ingénieurs,* Presses Polytechniques et Universitaires Romandes , 2003.

Kaufmann, A. *« Méthodes et modèles de la recherche opérationnelle* », Dunod 1959

Kaufmann, A. « *Méthodes et Modèles de la Recherche Opérationnelle* », tome II, Dunod 1963

Miller, R.E. «*Optimization ; Foundations and applications*», Wiley 2000

Minoux, M. « *Programmation mathématique ; théorie et algorithmes »*, Dunod 1983

Roseaux *« Exercices et problèmes résolus de Recherche Opérationnelle »*, tome 3, Masson 1985

Roseaux, « *Exercices et problèmes résolus de Recherche Opérationnelle* », Masson 1983

Simonnard, M. *« Programmation linéaire »*, Dunod 1972

# TABLE DES MATIERES

Introduction ..... 5

Chapitre 1 ● Généralités sur la Programmation Linéaire ..... 13
1.1. UN EXEMPLE SIMPLE 13
1.2. ETUDE DU CAS GÉNÉRAL 16
1.3. THÉORÈMES GÉNÉRAUX SUR LES PROGRAMMES LINÉAIRES 20
1.4. INTRODUCTION À L'ALGORITHME DU SIMPLEXE 32
ANNEXE 40

Chapitre 2 ● Algorithme du simplexe ..... 45
2.1. UN EXEMPLE SIMPLE 45
2.2. CAS GÉNÉRAL. RÈGLES DU PIVOTAGE 56
2.3. RECHERCHE D'UNE SOLUTION DE BASE INITIALE 60
2.4. CAS DE LA DÉGÉNÉRESCENCE DU PREMIER TYPE 63

Chapitre 3 ● Dualité ..... 67
3.1. DÉFINITION 67
3.2. SIGNIFICATION ÉCONOMIQUE DU PROGRAMME DUAL 68
3.3. CORRESPONDANCE ENTRE VARIABLES DU PRIMAL ET VARIABLES DU DUAL 69
3.4. THÉORÈMES SUR LA DUALITÉ 71
3.5. PASSAGE DU PRIMAL AU DUAL SUR UN EXEMPLE 78
3.6. INTERPRÉTATION ÉCONOMIQUE DES VARIABLES DU DUAL 80

Chapitre 4 ● Paramétrisation ..... 83
4.1. PARAMÉTRISATION DE LA FONCTION ÉCONOMIQUE 84
4.2. PARAMÉTRISATION DU SECOND MEMBRE 88
4.3. PARAMÉTRISATION D'UN COEFFICIENT DE LA MATRICE DES CONTRAINTES 93

Chapitre 5 ● Compléments et autres algorithmes ..... 95
5.1. AUTRES MÉTHODES 95
5.2. PROLONGEMENTS 103

Chapitre 6 ● Problèmes de chemins ..... 107
6.1. DÉFINITION D'UN GRAPHE 107
6.2. ELÉMENTS DE VOCABULAIRE DE LA THÉORIE DES GRAPHES 109
6.3. PROBLÈMES DE CHEMINS 113
6.4. APPLICATION DE LA NOTION DE CHEMINEMENT DANS UN GRAPHE : LES PROBLEMES D'ORDONNANCEMENT 132
6.5. LES AUTRES TYPES DE PROBLÈME 148

Chapitre 7 ● Arbres et arborescences ..... 149

7.1. ARBRES – DÉFINITION 149
7.2. PROBLÈMES SUR LES ARBRES 152
7.3. ARBORESCENCES – DÉFINITION 156
7.4. LES PROCÉDURES DE RECHERCHE ARBORESCENTS 160

Chapitre 8 ● Complexité des problèmes et heuristiques ..... 179
8.1. RETOUR SUR LA COMPLEXITÉ DES ALGORITHMES 179
8.2. METAHEURISTIQUES 182

Chapitre 9 ● Problèmes de flots ..... 197
9.1. FLOTS – DÉFINTION 197
9.2. PROBLÈME DU FLOT MAXIMAL 198
9.3. PROBLÈME DU FLOT COMPATIBLE 208
9.4. PROBLÈME DU FLOT DE COÛT MINIMAL 211
9.5. LE PROGRAMME DE TRANSPORT 212
9.6. LES PROBLÈMES D'AFFECTATION 223

Chapitre 10 ● Les chaînes de Markov ..... 243
10.1. DÉFINITION D'UNE CHAÎNE DE MARKOV 244
10.2. PROBLÈME DU VECTEUR STOCHASTIQUE LIMITE 247
10. 3. CONDITIONS D'ERGODICITÉ 248
10.4. RECHERCHE DU VECTEUR D'ÉTAT LIMITE 254
10.5. CHAÎNE DE MARKOV AVEC VALEURS DE TRANSITION 257
10.6. PROGRAMMATION DYNAMIQUE DISCRÈTE 261

Chapitre 11 ● Phénomènes d'attente ..... 279
INTRODUCTION 279
11.1. SYSTÈME OUVERT À UNE SEULE STATION. ARRIVÉES POISSONNIENNES ET SERVICE EXPONENTIEL 280
11.2. SYSTÈME OUVERT À PLUSIEURS STATIONS. ARRIVÉES POISSONNIENNES ET SERVICE EXPONENTIEL 288
11.3. RETOUR AU CAS D'UNE SEULE STATION. MÉTHODE INTÉGRALE 293
11.4. SYSTÈMES PLUS COMPLEXES 298

Chapitre 12 ● Problèmes de défaillances d'équipements ..... 303
INTRODUCTION 303
12.1. DÉFINITIONS ET GÉNÉRALITÉS 303
12.2. CLASSEMENT DES APPAREILS SUIVANT LES LOIS DE PROBABILITÉS DE LEUR DURÉE DE VIE 305
12.3. QUELQUES LOIS DE PROBABILITÉS USUELLES POUR LES DURÉES DE VIE 306
12.4. RACCORDEMENT D'UNE CHRONIQUE DE DÉFAILLANCE À UNE LOI CONNUE 311
12.5. PROCESSUS DE RENOUVELLEMENT 313
12.6. PROCESSUS D'APPROVISIONNEMENT 318

12.7. STRATÉGIES DE REMPLACEMENT 320

Chapitre 13 ● Problème de stocks ..... 327
INTRODUCTION 327
13.1. FORMULE DE WILSON 328
13.2. MODÈLES DE STOCK EN UNIVERS ALÉATOIRE 331
13.3. DEUX GRANDS TYPES DE POLITIQUES DE STOCKS 342
BIBLIOGRAPHIE 345

Table des matières ..... 347

Cet ouvrage propose une introduction aux méthodes généralement utilisées dans le domaine de l'optimisation. Son objectif est double : proposer un ensemble de modélisations classiques, à l'aide principalement de la théorie des graphes, la programmation linéaire et les processus aléatoires ; décrire un ensemble de méthodes exactes ou approchées pour résoudre les problèmes d'optimisation ainsi modélisés.

Issu du cours de recherche opérationnelle de l'École Nationale Supérieure des Mines de Paris, l'ouvrage s'adresse aux élèves des écoles d'ingénieurs, aux étudiants de deuxième cycle, ainsi qu'à tous ceux (ingénieurs, chercheurs,...) qui souhaitent se familiariser avec les méthodes d'optimisation en gestion les plus souvent utilisées.