Frédéric Meunier

Aléa et Temps Réel dans la Supply Chain : Outils Mathématiques

---

22 février 2026

Table des matières

Introduction

Avec les progrès récents de l'informatique et des nouvelles technologies (intelligence artificielle, apprentissage automatique, internet des objets, etc.) et l'arrivée du e-commerce, les besoins de la Supply Chain en matière de maîtrise des incertitudes et du temps réel ont fortement cru ces dernières années. La complexité des problématiques qu'un consultant ou ingénieur travaillant dans la Supply Chain peut rencontrer au quotidien ne cesse de grandir et les interactions avec les dernières découvertes techniques ou scientifiques du monde "digital" sont désormais constantes.

L'objectif principal du cours est de former les élèves à la manipulation des outils de la recherche opérationnelle et des probabilités adaptés à une prise de décision optimisée dans la Supply Chain, dans un contexte d'incertitudes et de temps réel. Un des fils conducteurs de ce cours est donc la question : comment prendre des décisions optimisées dans la Supply Chain lorsque les données ne sont pas parfaitement connues ? Un second objectif est d'accroître les capacités des élèves à modéliser les problèmes de décision propres à la Supply Chain et à prendre du recul quand il s'agit d'interpréter des résultats fournis par des algorithmes ou des simulations numériques pour de tels problèmes.

Bien que résolument guidé par les applications, ce cours se veut scientifique. Le contenu mathématique et informatique y est présenté de manière rigoureuse et précise, sous forme de résultats formalisés (comme les théorèmes ou les propositions). C'est la forme la plus sûre permettant la compréhension des conditions exactes d'applications de ces outils et la prise de recul que l'on peut attendre de la part d'ingénieurs généralistes de haut niveau. Les preuves de résultats mathématiques classiques et élémentaires seront en général omises ; de même, les preuves s'appuyant sur des résultats trop avancés par rapport au niveau du cours seront également omises. En revanche, les preuves accessibles de résultats moins standard seront quasiment toujours données dans le détail. Elles permettront aux élèves qui le souhaitent d'acquérir une meilleure compréhension des résultats, voire de pouvoir les étendre à des situations différentes de celles abordées en cours.

CHAPITRE 1

Loi de Little

1.1 Introduction

Un des résultats fondamentaux des files d'attente est la loi de Little. C'est un résultat tellement fondamental qu'on ne l'appelle pas "théorème" ou "principe", mais "loi", comme les lois de la physique. Cette loi, démontrée par Little en 1961 [13], concerne un système quelconque dans lequel des "clients" arrivent et passent un certain temps avant de le quitter. Informellement, cette loi stipule que le nombre moyen de clients dans le système est égal à la fréquence moyenne d'arrivée fois le temps moyen passé dans le système. Cela peut d'ailleurs sembler tellement évident que la nécessité d'une preuve peut paraître inutile. Il suffit cependant d'essayer de formaliser cette propriété pour voir que la question n'est pas si triviale que cela, et avant la preuve de Little, certains pensaient pouvoir construire des exemples où elle ne serait pas satisfaite.

Les applications de cette loi sont extrêmement nombreuses. En pratique, elle peut par exemple servir à dimensionner un magasin ou un entrepôt : connaissant l'intensité du flux entrant et le temps passé en moyenne, on peut en déduire le nombre de clients ou d'unités présents en moyenne. Elle peut aussi servir à analyser une chaîne de production : si l'en-cours de production (nombre d'unités sur la chaîne) reste borné, la fréquence d'arrivée peut être remplacée dans la formule par le taux de production et la loi de Little indique que le délai de production (ou "lead time") est égal au produit du taux de production par l'en-cours. La loi de Little sert aussi en théorie : elle peut par exemple être appliquée à certains sous-systèmes d'une file d'attente ou d'un réseau bien choisi et permettre des calculs bien plus simples que ceux qu'une approche directe aurait imposés.

La loi de Little existe désormais sous de nombreuses formes. Celle que l'on propose dans cette section est l'une des plus générales et va être énoncée dans un contexte déterministe. Nous verrons au chapitre 2 comment cette version déterministe implique une version stochastique.

1.2 Modèle et énoncé

On considère donc un système vide à l'instant $t = 0$ dans lequel arrivent des clients. Soit t_n l'instant d'arrivée du n-ième client et on suppose que la suite des instants d'arrivée, $t_1, t_2, \ldots$, est une suite croissante de $\mathbb{R}_+$ (pas nécessairement strictement) tendant vers $+\infty$.

On note $S_n \geqslant 0$ le temps passé par le n-ième client dans le système. Il convient de souligner que les clients peuvent très bien sortir dans un ordre différent de celui des arrivées, ce qui constitue un autre témoignage de la portée générale de cette loi.

On pose

- Q_t le nombre de clients dans le système à l'instant t, sans compter les éventuels clients le quittant précisément à cet instant.
- N_t le nombre de clients arrivés dans le système jusqu'à l'instant t inclus.

On peut donner une définition mathématique de ces quantités :

$$Q_t = |\{n\colon t_n \leqslant t < t_n + S_n\}| \qquad \text{et} \qquad N_t = \max\{n\colon t_n \leqslant t\}\,.$$

Lorsque les quantités

$$\frac{1}{t}\int_0^t Q_s\,\mathrm{d}s, \qquad \frac{N_t}{t}, \qquad \frac{1}{n}\sum_{k=1}^{n} S_k$$

convergent pour t et n tendant vers $+\infty$, on note leurs limites respectives par $\bar{q}$, par $\bar{\lambda}$ et par $\bar{s}$. Ces quantités peuvent être interprétées respectivement comme le *nombre moyen de clients*, le *fréquence moyenne d'arrivée* et le *temps de séjour moyen*. Chacune de ces limites peut exister indépendamment des autres, mais on a le résultat suivant.

Théorème 1.1 (Loi de Little)**.** *Si $\bar{\lambda}$ et $\bar{s}$ existent et sont finis, alors $\bar{q}$ existe et on a $\bar{q} = \bar{\lambda}\bar{s}$.*

Application 1.1. Le gérant d'un supermarché constate qu'il y a en moyenne 80 clients dans son magasin et que la fréquence d'arrivée est de 120 personnes par heure. Il en conclut qu'en moyenne, un client passe un temps de

$$\bar{s} = \frac{\bar{q}}{\bar{\lambda}} = \frac{80}{120} = 40 \text{ min}$$

dans son magasin.

1.3 Preuve de la loi de Little

Afin de démontrer la loi de Little, nous établissons un petit lemme qui montre que, sous les conditions de la loi de Little, le temps de séjour ne peut pas croître trop vite.

Lemme 1.2. *Si $\bar{\lambda}$ et $\bar{s}$ existent et sont finis, alors* $\displaystyle\lim_{n\to+\infty} \frac{S_n}{n} = \lim_{n\to+\infty} \frac{S_n}{t_n} = 0.$

Démonstration. En écrivant $\frac{1}{n}S_n = \frac{1}{n}\sum_{k=1}^{n} S_k - \frac{n-1}{n}\frac{1}{n-1}\sum_{k=1}^{n-1} S_k$, on voit que $\lim_{n\to+\infty} \frac{S_n}{n} = \bar{s} - \bar{s} = 0$.

Notons maintenant que $\frac{n}{t_n} \leqslant \frac{N_{t_n}}{t_n}$, ce qui implique que $\limsup_n \frac{n}{t_n} \leqslant \bar{\lambda}$. En écrivant $\frac{S_n}{t_n} = \frac{n}{t_n}\frac{S_n}{n}$, on obtient $\lim_{n\to+\infty} \frac{S_n}{t_n} = 0$. □

Démonstration du théorème 1.1. On a

$$\int_0^t Q_s \,\mathrm{d}s = \int_0^t \sum_{n\colon t_n \leqslant t} \mathbf{1}(t_n \leqslant s < t_n + S_n)\,\mathrm{d}s = \sum_{n\colon t_n \leqslant t} \min(S_n, t - t_n)\,.$$

Par conséquent, on a

$$\sum_{n\colon t_n + S_n \leqslant t} S_n \leqslant \int_0^t Q_s \,\mathrm{d}s \leqslant \sum_{n\colon t_n \leqslant t} S_n\,. \tag{1.1}$$

En utilisant le membre de droite, on obtient $\dfrac{1}{t}\displaystyle\int_0^t Q_s \,\mathrm{d}s \leqslant \frac{N_t}{t}\frac{1}{N_t}\sum_{n=1}^{N_t} S_n$. On a donc $\limsup\limits_t \dfrac{1}{t}\displaystyle\int_0^t Q_s \,\mathrm{d}s \leqslant \bar{\lambda}\bar{s}$.

Il reste donc à montrer un résultat semblable pour la limite inférieure, ce que l'on va faire en utilisant le membre de gauche de (1.1). Soit $\varepsilon > 0$. D'après le lemme 1.2, il existe n_0 tel que pour tout $n \geqslant n_0$, on a $S_n \leqslant \varepsilon t_n$, ou encore $t_n + S_n \leqslant (1+\varepsilon)t_n$. Cela implique que

$$\sum_{n\colon t_n + S_n \leqslant t} S_n \geqslant \sum_{\substack{n\colon (1+\varepsilon)t_n \leqslant t \\ n \geqslant n_0}} S_n = \sum_{n=1}^{N_{\frac{t}{1+\varepsilon}}} S_n - \sum_{n=1}^{n_0-1} S_n\,.$$

En divisant par t et en faisant tendre t vers $+\infty$, on obtient, après un calcul en tout point semblable à celui qui a permis d'obtenir la borne supérieure :

$$\liminf_t \frac{1}{t}\int_0^t Q_s \,\mathrm{d}s \geqslant \frac{1}{1+\varepsilon}\bar{\lambda}\bar{s}\,.$$

Comme cette inégalité est vérifiée pour tout $\varepsilon > 0$, la conclusion suit. □

1.4 Fréquence de sortie et chaîne de production

On pose maintenant M_t le nombre de clients ayant quitté le système jusqu'à l'instant t inclus. On a la proposition suivante.

Proposition 1.3. *Supposons que $\bar{\lambda}$ existe et est fini. Alors $\limsup_{t\to+\infty} M_t/t \leqslant \bar{\lambda}$. Si de plus $\lim_{n\to+\infty} S_n/n = 0$, alors M_t/t converge vers $\bar{\lambda}$.*

Démonstration. On a $M_t/t \leqslant N_t/t$ pour tout t, ce qui implique immédiatement la première partie de l'énoncé.

Prouvons maintenant la seconde partie. Tout comme dans la preuve du lemme 1.2, on note que $\frac{n}{t_n} \leqslant \frac{N_{t_n}}{t_n}$, ce qui implique que $\limsup_n \frac{n}{t_n} \leqslant \bar{\lambda}$. En écrivant $\frac{S_n}{t_n} = \frac{n}{t_n}\frac{S_n}{n}$, on obtient alors $\lim_{n\to+\infty} \frac{S_n}{t_n} = 0$. Pour $\varepsilon > 0$, il existe donc n_0 tel que pour tout $n \geqslant n_0$, on a $S_n \leqslant \varepsilon t_n$. Par conséquent,

$$M_t = |\{n\colon t_n + S_n \leqslant t\}| \geqslant |\{n \geqslant n_0\colon t_n + S_n \leqslant t\}| \geqslant |\{n \geqslant n_0\colon (1+\varepsilon)t_n \leqslant t\}| \geqslant N_{\frac{t}{1+\varepsilon}} - n_0\,.$$

Cela implique que $\liminf_t M_t/t \geqslant \frac{\bar{\lambda}}{1+\varepsilon}$ pour tout $\varepsilon > 0$, ce qui permet de conclure. □

Cette proposition montre que si le temps de séjour reste borné ou croît modérément, alors la fréquence moyenne de sortie est égale à la fréquence moyenne d'arrivée. Pour des temps de séjour croissants suffisamment rapidement, le taux de sortie M_t/t peut converger vers une

valeur arbitrairement plus faible que $\bar{\lambda}$. Par exemple, si $S_n = \alpha n$ avec $\alpha > 0$ et $t_n = n$ pour tout n, alors $\bar{\lambda} = 1$ et

$$\lim_{t \to +\infty} \frac{M_t}{t} = \frac{1}{1+\alpha}\,.$$

(Cette dernière limite s'obtient facilement en notant qu'au bout n unités de temps, environ $n/(1+\alpha)$ clients seront sortis.)

Comme annoncé dans l'introduction de chapitre, ce résultat peut être utile pour analyser une chaîne de production. Dans ce contexte, $\bar{q}$ est l'*en-cours* (nombre d'unités en cours de production), $\bar{s}$ est le *délai de production*, ou *lead time*, et $\lim_{t\to+\infty} M_t/t$ est le taux de production. Si la chaîne de production ne présente pas de problème, la formule de Little peut se lire

$$\text{en-cours} = \text{taux de production} \times \text{délai de production}\,.$$

Application 1.2. Si on double le taux de production d'une chaîne de production sans changer le délai de production, l'en-cours va nécessairement doubler.

Terminons ce chapitre par une petite remarque : la formule précédente permet aussi de se rappeler que le taux de production et le délai de production sont deux indicateurs de performance qui peuvent varier de manière indépendante. Selon les contextes, il peut judicieux d'en optimiser un en priorité sur l'autre, mais cela conduit forcément à un ajustement sur l'en-cours.

1.5 Compléments bibliographiques

Cinquante ans après son article fondateur sur la formule qui porte désormais son nom, Little en a écrit un autre sur le sujet [14], où il discute l'histoire de la formule, ses nombreuses applications et quelques généralisations.

CHAPITRE 2

Files d'attente

2.1 Introduction

Les files d'attente forment un domaine à la fois actif et difficile des probabilités théoriques. En même temps, c'est un domaine dont les applications sont innombrables, dans les télécommunications, en informatique et en supply chain. Depuis les années 40, les files d'attente sont utilisées pour dimensionner des services, des équipes d'entretien ou pour analyser des systèmes logistiques. L'un des articles fondateurs du sujet, dû à Jackson, a d'ailleurs été identifié en 2004 comme l'un des dix articles les plus influents des 50 premières années de *Management Science*, une des revues académiques de référence en génie industriel.

Bien que reliée à des questions ardues de mathématiques, la théorie des files d'attente fournit des outils directement applicables pour un ingénieur. Ce chapitre présente quelques résultats fondamentaux dont la portée pratique est établie depuis longtemps. Comme partout dans ce cours, seuls les aspects mathématiques incontournables seront évoqués, parfois sous une forme simplifiée, mais toujours rigoureuse, pour éviter des discussions techniques dépassant le cadre de ce cours.

Ce chapitre est partagé en deux sections. La première section, intitulée "Processus de Poisson", présente l'un des concepts fondamentaux pour la modélisation d'arrivées aléatoires non concertées d'individus ou d'événements : les processus de Poisson. La seconde section, intitulée "Files d'attente simples", aborde les files d'attente les plus élémentaires—un seul serveur, pas de problème de capacité—mais dont les applications sont déjà très nombreuses et permettent d'avoir un premier aperçu des modèles et résultats de ce domaine.

2.2 Processus de Poisson

Il n'est pas rare d'avoir des systèmes dans lequel des clients, des unités, des événements, etc. arrivent de manière aléatoire et indépendamment les uns des autres. L'ingénieur qui souhaite modéliser un tel système cherchera un outil des probabilités qui rendra bien compte d'un tel phénomène.

Il s'avère qu'il en existe un qui réalise cette tâche de manière parfaite, tant en théorie qu'en pratique. Il s'agit des processus de Poisson, que l'on définit comme suit.

Notons T_1 l'instant d'arrivée du premier client, et T_n le temps séparant les arrivées des $(n-1)$-ième et n-ième clients. On dit que les arrivées forment un *processus de Poisson* si les

T_n sont indépendants et suivent chacune une *loi exponentielle* de même paramètre, c'est-à-dire que les T_n sont de support $[0, +\infty[$ et de densité $t \mapsto \lambda e^{-\lambda t}$ pour un certain $\lambda > 0$ (le *paramètre*). L'*intensité* du processus de Poisson est par définition la valeur de ce paramètre λ. Rappelons les propriétés suivantes de la loi exponentielle.

Proposition 2.1. *Soit X une variable aléatoire suivant une loi exponentielle de paramètre λ. Alors*

$$\mathbb{E}[X] = \frac{1}{\lambda} \qquad \text{et} \qquad \mathbb{V}[X] = \frac{1}{\lambda^2}\,.$$

Les phénomènes d'arrivées indépendantes, invariant dans le temps, sont quasiment toujours modélisés par un processus de Poisson. (Lorsqu'il n'y a pas invariance dans le temps, ce qui survient dès que lorsqu'on considère un système réel suffisamment longtemps, on peut encore utiliser des processus de Poisson *inhomogènes*, mais ces derniers ne seront pas étudiés dans ce cours.) Toutes les expériences confirment la pertinence de ce choix, et la raison mathématique est donnée par les propositions 2.2 et 2.3 suivantes. Elle sont classiques et nous omettons les preuves. Une variable aléatoire continue X, à valeurs réelles positives, est *sans mémoire* si pour tous s, t dans $\mathbb{R}_+$ on a $\mathbb{P}(X > t + s \,|\, X > s) = \mathbb{P}(X > t)$.

Proposition 2.2. *Une variable aléatoire continue X, à valeurs réelles positives, est sans mémoire si et seulement si X suit une loi exponentielle.*

Notons N_I le nombre de clients arrivant sur un intervalle de temps I. La proposition suivante est peut-être celle qui constitue la justification théorique la plus convaincante de la pertinence des processus de Poisson. (Une preuve être trouvée ici : [2, Theorem 2.1.14].)

Proposition 2.3. *Supposons que, presque sûrement, les T_n sont strictement croissants et tendent vers $+\infty$. Les T_n forment un processus de Poisson (homogène) si et seulement si les deux propriétés suivantes sont satisfaites.*

- *Pour toute famille finie $\mathcal{I}$ d'intervalles de temps deux à deux disjoints, les $(N_I)_{I\in\mathcal{I}}$ forment une famille de variables aléatoires indépendantes.*
- *Pour tout intervalle de temps I, l'espérance de N_I est proportionnelle à la longueur de I.*

Nous énonçons maintenant quelques propriétés des processus de Poisson, toutes faciles à montrer, et dont nous omettons également les preuves.

Le nom des processus de Poisson vient de la propriété suivante. La *loi de Poisson* de paramètre α est une distribution de probabilité discrète donnant au tirage $k \in \mathbb{Z}_+$ la probabilité $e^{-\alpha}\frac{\alpha^k}{k!}$.

Proposition 2.4. *Considérons un processus de Poisson d'intensité λ. Le nombre d'arrivées dans un intervalle de longueur τ suit une loi de Poisson de paramètre $\lambda\tau$.*

On peut superposer des suites d'arrivées : étant données deux suites d'arrivées (avec des instants d'arrivée tous distincts, ce qui arrive presque sûrement avec des processus de Poisson), on peut considérer la suite des arrivées globalement, en oubliant à quel processus chaque arrivée appartient. On a la propriété suivante, encore une fois très intuitive quand on voit des processus de Poisson comme des arrivées non concertées.

Proposition 2.5. *La superposition de deux processus de Poisson indépendants, d'intensités respectives λ_1 et λ_2, est un processus de Poisson d'intensité $\lambda_1 + \lambda_2$.*

La propriété suivante est un peu la réciproque de la propriété précédente.

Proposition 2.6. *Considérons un processus de Poisson d'intensité λ. Pour chaque arrivée, décidons de la conserver avec probabilité p, indépendamment des autres arrivées. Les arrivées non éliminées forment alors un processus de Poisson d'intensité $p\lambda$.*

2.3 Files d'attente simples : concepts fondamentaux

2.3.1 Notation, terminologie et hypothèses

Une *file d'attente* est formée d'un *espace d'attente* et d'un *espace de service*. Des clients arrivent dans l'espace d'attente selon un *processus d'arrivée* et sont servis dans l'espace de service selon un *processus de service*.

Les intervalles de temps entre deux arrivées successives forment une suite de variables aléatoires. On se place à un instant arbitraire à partir duquel on commence à compter les clients (la file peut très bien être non vide à l'arrivée du premier client). La variable aléatoire T_1 est l'instant d'arrivée du premier client, et le temps séparant les arrivées des $(n-1)$-ième et n-ième clients est noté T_n. Par exemple, lorsque les T_n sont indépendants et suivent tous une loi exponentielle de même paramètre, les arrivées forment un processus de Poisson. Une file d'attente peut avoir une *capacité*, définie comme le nombre maximal de clients qui peuvent attendre simultanément dans l'espace d'attente. Si un client arrive mais n'a plus de place, il disparaît simplement.

La file d'attente possède un certain nombre de *serveurs* qui servent les clients. Lorsqu'un client arrive dans la file, il se rend dans l'espace de d'attente. Dans ce cours, on supposera que le processus de service suit la règle suivante (*hypothèse FIFO*) : Dès qu'un serveur est libre, le client le plus ancien dans l'espace d'attente se rend dans l'espace de service et est servi par un serveur libre, lequel devient alors occupé. Les durées de service sont les réalisations de variables aléatoires. On note U_n la durée de service du n-ième client.

On suppose dans ce cours que les variables aléatoires T_n sont iid, strictement positives et d'espérance finie, et de même pour les variables U_n. Les suites (T_n) et (U_n) sont de plus supposées indépendantes l'une de l'autre. On note $\frac{1}{\lambda}$ l'espérance commune des T_n et $\frac{1}{\mu}$ celle des U_n.

Selon la convention de Kendall, une telle file d'attente est notée $A/S/m/K$, où A donne la nature du processus d'arrivée, S donne celle du processus de service, m est le nombre de serveurs et K est la capacité de la file d'attente. Lorsque $K = +\infty$, on écrit simplement $A/S/m$. Dans ce cours, on considéra pour A et S les processus suivants :

- M, pour markovien : les durées suivent des distributions exponentielles. Quand M concerne les arrivées, cela signifie donc qu'elles forment un processus de Poisson.
- D, pour déterministe : les durées sont des quantités fixées une fois pour toute.
- G, pour général : les distributions sont quelconques.

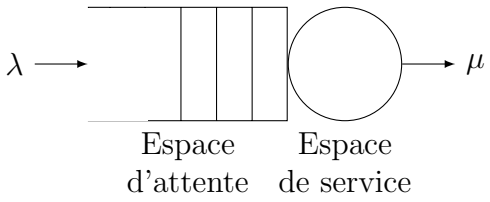

FIGURE 2.1 – Une file d'attente $G/G/1$.

Ce cours se concentrera principalement sur la file $G/G/1$ et certains cas particuliers, comme $M/G/1$.

2.3.2 Les quantités étudiées pour une file $G/G/1$

Étant donnée une file d'attente, différentes quantités peuvent présenter un intérêt. Les principales sont les suivantes :

- le nombre Q_t de clients dans la file d'attente à l'instant t (dans les espaces d'attente et de service), sans compter un client partant à l'instant t.
- le nombre N_t de clients arrivés à l'instant t depuis l'instant 0, instant t inclus.
- le nombre M_t de clients ayant été servis à l'instant t depuis l'instant 0, instant t inclus.
- le *temps d'attente* W_n du n-ième client avant d'être servi.
- le *temps de séjour* S_n du n-ième client, défini comme la somme des temps d'attente et de service : $S_n = W_n + U_n$.

Noter que ce sont des variables aléatoires.

Proposition 2.7. *On a*

$$\lim_{t\to+\infty} N_t = \lim_{t\to+\infty} M_t = +\infty \quad et \quad \lim_{t\to+\infty} \frac{N_t}{t} \overset{p.s.}{=} \lambda\,.$$

Démonstration. Pour tout entier n positif, on a $N_t \geqslant n_0$ pour $t \geqslant \sum_{k=1}^{n_0} T_k$, et on a $M_t \geqslant n_0$ pour $t \geqslant \sum_{k=1}^{n_0}(T_k + U_k)$, ce qui prouve les limites de gauche.

Par ailleurs, on a $\sum_{k=1}^{N_t} T_k \leqslant t < \sum_{k=1}^{N_t+1} T_k$. En divisant par N_t, on obtient $\frac{1}{N_t}\sum_{k=1}^{N_t} T_k \leqslant t < \frac{N_t+1}{N_t}\frac{1}{N_t+1}\sum_{k=1}^{N_t+1} T_k$. Comme N_t tend presque sûrement vers $+\infty$, on obtient par la loi forte des grands nombres : $\frac{1}{\lambda} \overset{p.s.}{\leqslant} \frac{t}{N_t} \overset{p.s.}{\leqslant} \frac{1}{\lambda}$. □

Informellement, une file d'attente est stable si le temps d'attente ne tend pas à devenir arbitrairement grand. Une définition formelle possible est la suivante et que nous prendrons dans ce cours : une file d'attente est *stable* si, lorsque n tend vers $+\infty$, la suite $(W_n)_n$ converge en loi ou la suite $\left(\frac{1}{n}\sum_{k=1}^{n} W_k\right)_n$ converge presque sûrement vers une valeur finie. Ces deux convergences ne sont pas équivalentes : on peut avoir l'une sans l'autre. On choisit juste de dire que l'on a stabilité de la file si l'une au moins de ces convergences se réalise.

L'*intensité* d'une file $G/G/1$ est la quantité $\rho = \dfrac{\lambda}{\mu}$. Lorsque $\rho < 1$, la file est stable, et lorsque $\rho > 1$, la file ne l'est pas. Ce résultat, bien qu'intuitif, nécessite des outils plutôt

avancés pour être prouvé en toute généralité. Lorsque $\rho = 1$, les choses peuvent être plus complexes, et le comportement dépend fortement des lois des processus d'arrivée et de service. Concernant la stabilité de la file $G/G/1$, le résultat partiel suivant admet une preuve simple. Comme une variable aléatoire tendant presque sûrement vers l'infini ne peut converger en loi, ce résultat montre qu'une file n'est pas stable lorsque $\rho > 1$.

Proposition 2.8. *Considérons une file $G/G/1$ avec $\rho > 1$. Alors Q_t tend presque sûrement vers $+\infty$ quand t tend vers $+\infty$ et W_n tend presque sûrement vers $+\infty$ quand n tend vers $+\infty$.*

Démonstration. Prouvons d'abord la convergence presque sûre de Q_t. Définissons $A_n = \sum_{k=1}^{n} T_k$ et $B_n = \sum_{k=1}^{n} U_k$. La loi forte des grands nombres implique que $\frac{1}{n}A_n$ converge presque sûrement vers $\frac{1}{\lambda}$ quand n tend vers $+\infty$. Noter que $t < A_{N_t+1}$ pour tout t. La quantité N_t tend presque sûrement vers $+\infty$. En divisant par N_t l'inégalité précédente, on obtient $\limsup_t \frac{t}{N_t} \overset{p.s.}{\leqslant} \frac{1}{\lambda}$. De manière semblable, on peut noter que $\frac{1}{n}B_n$ converge presque sûrement vers $\frac{1}{\mu}$ quand n tend vers $+\infty$ et que $B_{M_t} \leqslant t$ pour tout t. La quantité M_t tend presque sûrement vers $+\infty$. En divisant par M_t l'inégalité précédente, on obtient $\liminf_t \frac{t}{M_t} \overset{p.s.}{\geqslant} \frac{1}{\mu}$. En combinant les deux résultats de convergence presque sûre précédents et en notant que $Q_t = N_t - M_t$, il vient $\liminf_t \frac{Q_t}{t} \overset{p.s.}{\geqslant} \lambda - \mu$, ce qui permet de conclure car $\lambda > \mu$.

Prouvons maintenant celle de W_n. Posons D_n l'instant de départ du n-ième client. On a $D_n \geqslant \sum_{k=1}^{n} U_k$ et donc $\liminf_n \frac{D_n}{n} \overset{p.s.}{\geqslant} \frac{1}{\mu}$. Par ailleurs,

$$W_n = \left(D_{n-1} - \sum_{k=1}^{n} T_k \right)^+ \geqslant D_{n-1} - \sum_{k=1}^{n} T_k \,,$$

d'où l'on déduit que $\liminf_n \frac{W_n}{n} \overset{p.s.}{\geqslant} \frac{1}{\mu} - \frac{1}{\lambda} > 0$. □

Nous terminons la section par la définition d'une autre quantité, souvent pertinente : l'*utilisation* de la file. Elle est définie comme la fraction du temps (asymptotiquement) où le service est assuré. En notant $\bar{u}$ l'utilisation d'une file d'attente, on a donc quand la limite existe :

$$\bar{u} = \lim_{t \to +\infty} \frac{1}{t} \int_0^t \mathbf{1}(Q_s > 0) \, \mathrm{d}s \,.$$

Remarquer que $\bar{u}$ vaut toujours au plus 1 par définition.

2.3.3 Valeurs "typiques" et régime permanent

La loi de Q_t dépend clairement de t : à l'instant 0 par exemple, on sait qu'il n'y a aucun client dans la file. De même pour les autres quantités mentionnées à la section 2.3.2 : leurs lois dépendent de n. Cependant, en particulier dans un contexte de génie industriel, on est plutôt intéressé par les valeurs "typiques" que peuvent prendre ces quantités dans le temps long. Une file d'attente est souvent dimensionnée pour fonctionner sur des périodes de temps longues. Il s'avère qu'en général, lorsque la file d'attente est stable, on peut donner du sens à "typique".

Considérons une file d'attente et choisissons une des familles de variables aléatoires associées (les W_n, les Q_t, etc.). Supposons que c'est une famille indicée par n et notons-la X_n.

Deux interprétations de valeur "typique" semblent raisonnables pour une file d'attente stable après un long temps de fonctionnement :

1. $\mathbb{E}[X_n]$: si l'on répète l'expérience un grand nombre de fois, la moyenne des réalisations de X_n ne dépend (presque) pas de n.
2. $\frac{1}{n}\sum_{k=1}^{n} X_k$: au cours d'une seule expérience, la moyenne des X_k sur les n premiers clients ne dépend (presque) pas de n et (presque) pas de la réalisation.

Cela motive la définition de régime permanent qui va suivre, pour lequel ces valeurs typiques existent et sont de plus égales. Noter que cette égalité n'est pas vraiment surprenante : la loi forte des grands nombres par exemple assure ce genre d'égalité.

On dit qu'une variable aléatoire X_n (indicée par le n-ième client) est *locale* si elle ne dépend que de quantités qui peuvent être mesurées par le n-ième client lors de son passage dans la file. De même, on dit qu'une variable aléatoire Y_t (indicée par l'instant t) est *locale* si elle ne dépend que de quantités qui peuvent être mesurées par un client qui arriverait à l'instant t. Cette notion n'est absolument pas standard et est spécifique à ce cours. Elle permet d'introduire la notion suivante, qui demande sinon des discussions beaucoup plus techniques. Une file est en *régime permanent* si, pour toute famille de variables aléatoires X_n (resp. Y_t si on indice par le temps) locales, on satisfait simultanément les deux conditions suivantes :

— la suite (X_n) (resp. (Y_t)) est *stationnaire* : en particulier[1], la loi ne dépend pas de l'indice.

— la suite (X_n) (resp. (Y_t)) est *ergodique* : en introduisant une variable aléatoire X (resp. Y) de même loi que les X_n (resp. Y_t), la quantité $\frac{1}{n}\sum_{k=1}^{n} X_k$ (resp. $\frac{1}{t}\int_0^t Y_s \,\mathrm{d}s$) converge presque sûrement vers $\mathbb{E}[X]$ (resp. $\mathbb{E}[Y]$) quand n (resp. t) tend vers $+\infty$.

(Les espérances peuvent être infinies.) En régime permanent, on s'autorisera donc souvent à omettre l'indice des variables aléatoires. Notons qu'en particulier, toute file en régime permanent est stable par définition (et donc $\rho \leqslant 1$) : les W_n sont tous de même loi et donc (W_n) converge trivialement en loi.

Réciproquement, dans un grand nombre de cas, une file stable va converger (dans un certain sens) vers un régime permanent unique, et c'est dans ce dernier état que l'on étudiera la file. Par ailleurs, l'étude des files est très souvent plus simple en régime permanent. Des raisonnements heuristiques et intuitifs conduiront en général à des résultats corrects.

Nous terminons la section avec un résultat intuitif : en régime permanent, si l'espérance du temps d'attente est finie, l'intensité des processus d'arrivées et de départs sont identiques.

Proposition 2.9. *En régime permanent, si* $\mathbb{E}[W_n]$ *est finie, alors* $\lim_{t\to+\infty} \frac{M_t}{t} \overset{p.s.}{=} \lambda$.

Démonstration. Supposons $\mathbb{E}[W_n]$ finie. On a $\mathbb{E}[S_n] = \mathbb{E}[W_n] + \frac{1}{\mu}$ finie également. Comme $\frac{S_n}{n} = \frac{1}{n}\sum_{k=1}^{n} S_k - \frac{n-1}{n}\frac{1}{n-1}\sum_{k=1}^{n-1} S_k$, le fait qu'on soit en régime permanent implique que S_n/n converge presque sûrement vers 0, et on conclut avec la proposition 1.3. □

1. C'est une conséquence de la stationnarité et non une définition.

En régime permanent avec $\mathbb{E}[W_n]$ finie, on peut aussi voir que l'espérance du temps séparant deux départs est égale à $1/\lambda$: en effet, le temps séparant le départ du n-ième client de celui du $(n-1)$-ième client est égal à $T_n + W_n - W_{n-1} + U_n - U_{n-1}$, et, comme les lois sont indépendantes de l'indice (pour les temps inter-arrivées et de service, c'est par définition ; pour les temps d'attente, c'est parce qu'on est en régime permanent), on obtient la conclusion souhaitée.

On a le corollaire intéressant suivant.

Corollaire 2.10. *En régime permanent, si $\mathbb{E}[W_n]$ est finie, l'utilisation est bien définie et est égale à l'intensité de la file : $\bar{u} = \rho$.*

Démonstration. On a $\frac{1}{t}\sum_{n=1}^{M_t} U_n = \frac{M_t}{t}\frac{1}{M_t}\sum_{n=1}^{M_t} U_n$. En passant à la limite quand t tend vers $+\infty$, on obtient $\lim_{t\to+\infty} \frac{1}{t}\sum_{n=1}^{M_t} U_n = \frac{\lambda}{\mu}$ (le numérateur vient de la proposition 2.9 et le dénominateur de la loi forte des grands nombres). L'encadrement

$$\sum_{n=1}^{M_t} U_n \leqslant \int_0^t \mathbf{1}(Q_s > 0)\,\mathrm{d}s \leqslant \sum_{n=1}^{M_t+1} U_n$$

permet de conclure. □

Enfin, la loi de Little prend une forme particulière pour une file en régime permanent pour laquelle $\mathbb{E}[W_n]$ est finie. La version donnée dans le théorème 1.1 devient une version presque sûre et, en utilisant la proposition 2.7, on obtient

$$\mathbb{E}[Q] = \lambda \mathbb{E}[S]\,. \tag{2.1}$$

2.4 Quelques files particulières

2.4.1 Les files $M/G/1$ et $M/M/1$

La file $M/G/1$ forme un modèle naturel pour un grand nombre de files d'attente réelles : les arrivées suivent un processus de Poisson, ce qui est pertinent dès que les arrivées des clients sont non concertées (indépendantes), et le temps de service suit n'importe quelle loi.

Parmi les résultats concernant la file $M/G/1$, le résultat suivant est probablement celui avec la plus grande portée. Il quantifie le temps d'attente moyen en régime permanent. Un résultat général concernant la file $M/G/1$, et dont la preuve dépasse le cadre de ce cours (voir par exemple [1, Chapitre X]), assure qu'une telle file, lorsque elle est stable, c'est-à-dire lorsque $\rho < 1$, converge (dans un certain sens) vers le régime permanent. Cela souligne la pertinence du théorème suivant, dus à Pollaczek et Khinchin : pour une file $M/G/1$ fonctionnant sur une durée suffisamment longue, ce théorème permet d'anticiper assez précisément le temps d'attente moyen. La loi de Little permet alors d'en déduire le nombre moyen de clients présents dans l'espace d'attente.

Théorème 2.11 (Théorème de Pollaczek–Khinchin [11, 16])**.** *Supposons $\rho < 1$. Pour une file d'attente $M/G/1$ en régime permanent, on a alors*

$$\mathbb{E}[W] = \frac{\lambda \mathbb{E}[U^2]}{2(1-\rho)}\,.$$

(C'est valable même si $\mathbb{E}[U^2] = +\infty$.)

Démonstration. (Nous supposerons dans la preuve que $\mathbb{E}[W] < +\infty$. Le cas $\mathbb{E}[W] = +\infty$ est plus technique et utilise des outils qui dépassent le cadre de ce cours.) Notons H'_n le nombre de clients en attente dans la file à l'arrivée du n-ième client, et R_n le temps qu'il reste au client en cours de service. On a $W_n = R_n + \sum_{k=n-H'_n}^{n-1} U_k$. En prenant l'espérance, on a

$$\mathbb{E}[W_n] = \mathbb{E}[R_n] + \mathbb{E}\left[\mathbb{E}\left[\sum_{k=n-H'_n}^{n-1} U_k \,\middle|\, H'_n\right]\right].$$

L'événement $\{H'_n = h\}$ ne dépend pas des U_k pour $k \geqslant n-h$, et donc $\mathbb{E}[W_n] = \mathbb{E}[R_n] + \mathbb{E}[H'_n]\cdot\mathbb{E}[U_n]$. Introduisons maintenant la variable aléatoire H_t, définie comme le nombre de clients en attente dans la file à l'instant t. En utilisant une propriété fondamentale des processus de Poisson—appelée "PASTA" [2]—, le fait qu'on soit en régime permanent implique qu'on a $\mathbb{E}[H'] = \mathbb{E}[H]$. Cela peut se réécrire

$$\mathbb{E}[W] = \mathbb{E}[R] + \frac{1}{\mu}\mathbb{E}[H]\,. \tag{2.2}$$

Par ailleurs, la loi de Little en régime permanent (2.1) implique, en prenant comme "système" la zone d'attente de la file, la relation $\mathbb{E}[H] = \lambda\mathbb{E}[W]$. En substituant $\mathbb{E}[H]$ dans l'équation (2.2), on obtient

$$\mathbb{E}[W] = \frac{\mathbb{E}[R]}{1-\rho}\,. \tag{2.3}$$

Il reste donc à calculer $\mathbb{E}[R]$. On introduit la variable aléatoire R'_t, définie comme la durée de service restant à l'instant t pour le client en cours de service. On a

$$\frac{1}{t}\int_0^t R'_s\,\mathrm{d}s = \frac{1}{t}\sum_{k=1}^{M_t}\frac{1}{2}U_k^2 = \frac{1}{2}\frac{M_t}{t}\frac{\sum_{k=1}^{M_t}U_k^2}{M_t} \tag{2.4}$$

pour tout instant t de départ d'un client. On est en régime permanent. La proposition 2.9 montre qu'en passant à la limite dans (2.4), on a $\mathbb{E}[R'] = \frac{1}{2}\lambda\mathbb{E}[U^2]$. En réutilisant "PASTA", on obtient $\mathbb{E}[R] = \frac{1}{2}\lambda\mathbb{E}[U^2]$, ce qui conduit à l'équation souhaitée, après substitution de $\mathbb{E}[R]$ dans (2.3). □

Application 2.1. Mettons en avant une application qualitative du théorème 2.11. Considérons deux caisses de supermarché, toutes deux étant capables de traiter le même nombre de clients par unité de temps. Ce théorème montre que si l'une à une variabilité du temps de traitement

2. PASTA, pour "Poisson Arrivals See Time Averages". Un cas particulier de cette propriété s'écrit : *Soit $(A_t)_{t\in\mathbb{R}_+}$ un processus réel tel que presque sûrement $t \mapsto A_t$ soit affine par morceaux. Soit $(X_n)_{n\in\mathbb{Z}_+}$ un processus de Poisson. Supposons que $\frac{1}{t}\int_0^t A_s\,\mathrm{d}s$ converge presque sûrement vers une quantité $\bar{a} \in \overline{\mathbb{R}}$. Alors $\frac{1}{n}\sum_{k=1}^{n} A_{X_n^-}$ converge presque sûrement vers $\bar{a}$. (Comme d'habitude, $A_{X_n^-}$ signifie $\lim A_x$ quand $x \to X_n$ par valeurs inférieures.)*

(ici, mesurée par la variance) plus grande que l'autre, son fonctionnement sera plus mauvais. En effet, l'espérance du temps d'attente dans la file (et donc de la longueur de la file, par Little) est linéaire en le moment d'ordre deux du temps de traitement.

Cela montre en particulier que prendre simplement le nombre moyen de clients traités par unité de temps comme métrique d'évaluation de performance d'une file d'attente ne peut pas être suffisant.

La file $M/M/1$ est celle dont le comportement est le mieux compris. Elle forme le cas particulier de la file $M/G/1$ pour laquelle les temps de service U_n suivent une loi exponentielle, tous comme les arrivées. De nombreuses situations peuvent être modélisées ainsi. Le fait que les calculs sont possibles dans ce cas conduit aussi souvent à modéliser une file d'attente réelle par une $M/M/1$ même si les temps de service ne suivent pas complètement des exponentielles.

Le théorème suivant rassemble quelques-unes des nombreuses propriétés de la file $M/M/1$. Noter que l'on a un résultat de convergence, avec l'expression de la distribution limite. Des résultats sur la vitesse de convergence existent aussi.

Théorème 2.12. *Supposons $\rho < 1$. Alors Q_t converge en distribution vers une variable aléatoire Q dont la distribution est donnée par*

$$\mathbb{P}(Q = k) = (1 - \rho)\rho^k \,.$$

En régime permanent, $\mathbb{E}[W] = \frac{\rho}{\mu - \lambda}$ et les départs de la file suivent un processus de Poisson d'intensité λ.

La valeur de $\mathbb{E}[W]$ en régime permanent est une conséquence directe du théorème Pollaczek–Khinchin (théorème 2.11). Le fait que l'intensité du processus des départs est égale à λ n'est pas une surprise : c'est une conséquence de la proposition 2.9. Noter que les départs d'une file $M/G/1$ en régime permanent ne suivent pas en général un processus de Poisson : il suffit de considérer une file $M/D/1$ pour s'en convaincre.

Nous omettons la preuve du reste de ce théorème, laquelle requiert des outils sortant du cadre de ce cours. Notons cependant que ces outils ne sont pas difficiles et que la preuve s'appuie essentiellement sur le caractère "markovien" de la file.

2.4.2 Approximation pour la file $G/G/1$

En l'absence d'information complémentaire, les praticiens n'hésitent pas à utiliser l'approximation de Kingman [12]. En notant c_X le *coefficient de variation* d'une variable aléatoire X (défini comme le rapport de son écart-type avec son espérance), elle affirme qu'on a pour une $G/G/1$ en régime permanent

$$\mathbb{E}[W] \approx \left(\frac{\rho}{\mu - \lambda}\right)\left(\frac{c_T^2 + c_U^2}{2}\right) .$$

Cette approximation est à privilégier lorsque ρ est proche de 1.

On peut vérifier que la formule ci-dessus coïncide avec celle du théorème de Pollaczek–Khinchin (théorème 2.11) quand le processus d'arrivées est un processus de Poisson. Par ailleurs, on voit ici que l'attente dans une file d'attente (ainsi que la longueur de la file) dépend de la variabilité de temps de service, comme souligné dans l'application 2.1, mais aussi des temps séparant deux arrivées successives.

2.5 Compléments bibliographiques

Un ouvrage particulièrement complet, et qui reste accessible sur le plan mathématiques, est le livre de Stewart [19].

CHAPITRE 3

Optimisation en ligne, optimisation temps réel

3.1 Introduction

Traditionnellement, les modèles et méthodes de la recherche opérationnelle considèrent que la totalité de l'instance d'un problème est connue lorsqu'il s'agit de le résoudre. En pratique, il n'est cependant pas rare que des décisions doivent être prises avant que les données du problème soient toutes disponibles. Lorsqu'il s'agit de résoudre un tel problème, on parle d'optimisation en ligne. Un algorithme en ligne prend des décisions séquentielles, en ne tenant compte que de la partie de l'instance qui a été révélée à l'instant courant. La problématique pratique peut parfois être encore plus exigeante et imposer des bornes strictes sur la durée maximale autorisée pour prendre une décision. Un algorithme temps réel est un algorithme en ligne qui prend des décisions en des temps courts.

Dans l'industrie, les transports ou la supply chain, les problèmes en ligne ou temps réel sont fréquents : affecter des objets à usiner aux machines d'un atelier sans connaître à l'avance l'ensemble des pièces à usiner pour la journée, organiser des tournées de collecte alors que l'ensemble des demandes n'est pas connue, faire fonctionner un entrepôt automatique, etc.

Contrairement à d'autres pans de l'optimisation, il y a toujours à ce jour une distance assez grande entre les fondements théoriques de l'optimisation en ligne et temps réel et les applications pratiques. Cependant, ces fondements théoriques possèdent un certain nombre de vertu—ils aident à formaliser les problèmes, à se poser la question de leur évaluation, etc.—et un ingénieur généraliste ne peut que gagner à en connaître les éléments essentiels. L'objectif de ce chapitre est donc double. D'une part, il vise à fournir certains outils théoriques permettant d'appréhender et de discuter ce type de problèmes. D'autre part, il fait certaines recommandations méthodologiques pour la conception et l'évaluation d'algorithmes pratiques de résolution de tels problèmes.

3.2 Quelques notions

3.2.1 Algorithme en ligne, algorithme temps réel

On distingue traditionnellement deux types de problème en ligne.

Dans le modèle *séquentiel*, l'instance comporte une séquence $r_1, r_2, \ldots$ de requêtes. Un algorithme *en ligne* doit choisir la façon de satisfaire la requête r_i avant de connaître la requête

r_{i+1}. L'objectif est de minimiser un coût global dépendant de l'ensemble des décisions prises.

Dans le modèle *temporel*, l'instance comporte également une séquence $r_1, r_2, \ldots$ de requêtes, mais toute requête r_i a un instant t_i à laquelle elle arrive et est révélée (et l'instant t_i lui-même n'est pas connu en avance). Ces instants satisfont $t_1 \leqslant t_2 \leqslant \cdots$. Un algorithme *en ligne* peut choisir d'attendre, ce qui peut induire un coût supplémentaire, avant de prendre une décision concernant la requête r_i. En particulier, l'algorithme n'est plus obligé de satisfaire les requêtes dans leur ordre d'arrivée. L'objectif est encore de minimiser un coût global dépendant de l'ensemble des décisions prises.

Lorsque l'algorithme est autorisé à connaître l'ensemble des requêtes pour prendre ses décisions, on parle de version *hors ligne* du problème. Dans le modèle temporel, la version hors ligne ne peut cependant satisfaire une requête avant son instant d'arrivée : le côté *dynamique*, c'est-à-dire dépendant du temps, n'est pas gommé.

Dans ce chapitre, les problèmes hors ligne seront essentiellement considérés au titre de comparaison, afin de mesurer à quel point devoir prendre les décisions sans connaître le futur peut contribuer à détériorer le coût.

Donner une définition formelle de "temps réel" est plus difficile et les auteurs ne s'accordent pas toujours sur ce point. On peut proposer une définition du style : un algorithme *temps réel* est un algorithme en ligne retournant une décision en des temps courts par rapport au "temps caractéristique" du système considéré (mais cela ne fait que repousser la question terminologique).

3.2.2 Analyse de compétitivité

Une façon d'évaluer théoriquement la qualité d'un algorithme en ligne passe par le ratio de compétitivité.Considérons un algorithme ALG, résolvant un certain problème d'optimisation en ligne. Notons par

- $\mathsf{ALG}(I)$ la valeur du coût obtenu lorsque cet algorithme est appliqué sur une instance I.
- $\mathsf{OPT}(I)$ la valeur optimale du coût pour l'instance I, pour la version hors ligne du problème.

Dans le cas d'un problème de minimisation, l'algorithme ALG est *c-compétitif*, avec c une constante réelle si, pour toute instance I, on a $\mathsf{ALG}(I) \leqslant c\,\mathsf{OPT}(I)$. Dans le cas d'un problème de maximisation, l'algorithme ALG est *c-compétitif* si, pour toute instance I, on a $\mathsf{OPT}(I) \leqslant c\,\mathsf{ALG}(I)$.

Dans les deux cas (minimisation, maximisation), le *ratio de compétitivité* de l'algorithme ALG est l'infimum des c pour lequel il est c-compétitif.

Le ratio de compétitivité est l'outil standard pour évaluer les performances théoriques d'un algorithme en ligne. Il convient de souligner que le temps de calcul n'est pas pris en compte dans l'analyse de compétitivité : seule l'information est considérée comme la ressource limitante.

Cette méthode d'évaluation des algorithmes en ligne s'appelle l'*analyse de compétitivité*.

3.3 Quelques exemples classiques

3.3.1 Location de skis

C'est peut-être l'exemple le plus classique. L'origine exact de ce problème n'est pas connue.

Un skieur décide d'aller skier tous les jours, jusqu'à ce que le mauvais temps arrive. Louer les skis lui revient à 1€ par jour. Acheter les skis lui revient à B€, avec $B \geqslant 1$. Il n'a aucune information sur la probabilité de mauvais temps. On considère qu'il prend sa décision le matin et qu'il connaît le temps qu'il va faire dans la journée. L'objectif est de minimiser le coût total. Si le skieur connaissait parfaitement le temps futur, la décision serait facile à prendre : si le nombre de beaux jours à partir du premier jour est supérieur à B, il vaut mieux acheter les skis ; sinon, il vaut mieux les louer.

C'est un problème en ligne de nature séquentielle. Chaque journée de beau temps forme une requête. Chaque requête peut être satisfaite de trois manières différentes :

— Louer les skis, à 1€.

— Acheter les skis, à B€.

— Utiliser les skis, si le skieur les a déjà achetés, à 0€.

Considérons l'algorithme ALG_k, paramétré par un entier k fixé, satisfaisant une requête comme suit : si le nombre de jours de location précédant le jour courant est au plus k, louer les skis ; sinon, acheter les skis.

C'est bien un algorithme en ligne : pour prendre sa décision, l'algorithme ne tient compte que de l'information passée. Noter qu'il se peut que l'algorithme n'aille pas jusqu'à acheter des skis, si par exemple le mauvais temps arrive avant le k-ième jour.

Théorème 3.1. *L'algorithme* ALG_{B-2} *est* $\left(2 - \frac{1}{B}\right)$*-compétitif.*

Démonstration. Considérons une instance I et notons n le nombre de jours de beau temps. On a

$$\mathsf{ALG}_{B-2}(I) = \begin{cases} n & \text{si } n \leqslant B-1, \\ 2B-1 & \text{si } n \geqslant B, \end{cases} \qquad \text{et} \qquad \mathsf{OPT}(I) = \min(n, B) \ .$$

Si $n \leqslant B-1$, on a $\mathsf{ALG}_{B-2}(I) = n \leqslant \left(2 - \frac{1}{B}\right) n = \left(2 - \frac{1}{B}\right) \mathsf{OPT}(I)$ car $B \geqslant 1$. Si $n \geqslant B$, on a $\mathsf{ALG}_{B-2}(I) = 2B - 1 = \left(2 - \frac{1}{B}\right) B = \left(2 - \frac{1}{B}\right) \mathsf{OPT}(I)$. □

On peut d'ailleurs démontrer qu'aucun algorithme ne peut offre de meilleur ratio de compétitivité comme suit. Étant donné un algorithme en ligne résolvant le problème de location de skis, considérer l'instance qui consiste en un jour de beau temps tant que l'algorithme choisit de louer, et qui consiste en un jour de mauvais temps précisément le lendemain du jour où l'algorithme décide d'acheter les skis. Soit n le nombre de jours de location choisis par l'algorithme. Le coût de la solution proposée par l'algorithme est $n + B$. La solution optimale hors ligne est de coût $\min(n+1, B)$. On vérifie aisément, en distinguant $n \leqslant B-2$ et $n \geqslant B-1$, que

$$\frac{n+B}{\min(n+1, B)} \geqslant 2 - \frac{1}{B} \ ,$$

ce qui montre que l'on ne peut trouver d'algorithme avec un meilleur ratio que celui donné par le théorème 3.1.

Dans la discussion précédente, on a implicitement supposé que l'algorithme était *déterministe* : sur une même instance, l'algorithme prend toujours les mêmes décisions. C'est ce qui permet de construire une instance sur laquelle l'algorithme renvoie une solution de mauvaise qualité. Cette supposition est parfois relâchée et on parle alors d'algorithme randomisé. Cela permet souvent d'obtenir de meilleurs ratios de compétitivité (avec la définition adéquate, car l'aléa rentre alors en jeu).

Ce type de raisonnement, permettant d'assurer l'"optimalité" de certains algorithmes en ligne, est souvent présenté avec un adversaire malveillant et omniscient (un démon), qui construit la pire instance possible, en utilisant les connaissances qu'il a de l'algorithme proposé.

3.3.2 Équilibrage de charge

Cet exemple est du a été proposé par Graham dans un article de 1966 [9]. C'est d'ailleurs peut-être le premier article à avoir considéré un problème en ligne.

On a une séquence de n tâches qui sont révélées les unes après les autres, le nombre n n'étant pas connu. La tâche i a un poids $w_i \geqslant 0$. On dispose de m machines. Chaque tâche doit être affectée à une machine à l'instant où elle est révélée. L'objectif est de minimiser la charge finale de la machine la plus chargée.

C'est un problème en ligne de nature séquentielle.

Considérons l'algorithme "glouton" affectant toujours la tâche courante à la machine la moins chargée.

Théorème 3.2. *L'algorithme glouton a un ratio de compétitivité égal à* $2 - \frac{1}{m}$.

Démonstration. Considérons une instance I. Mettons-nous à l'arrivée de la tâche i. Soit j la machine sur laquelle la tâche i est affectée par l'algorithme, et notons q_j sa charge. On a $q_j \leqslant \frac{1}{m}\sum_{k=1}^{i-1} w_k$ car l'algorithme affecte le tâche i à la machine la moins chargée (dont la charge est donc inférieure à la moyenne des charges). La charge de la machine j devient donc $q_j + w_i$, qui est donc majorée par $\frac{1}{m}\sum_{k=1}^{n} w_k + \frac{m-1}{m} w_i$ (on utilise ici la positivité des poids). On a

$$\frac{1}{m}\sum_{k=1}^{n} w_k \leqslant \mathsf{OPT}(I) \quad \text{et} \quad w_i \leqslant \mathsf{OPT}(I)\,.$$

La première inégalité vient du fait que dans toute affectation des n tâches, l'une des charges de machine au moins est supérieure à la moyenne des charges.

La charge de la machine j après affectation de la tâche i est donc majorée par $\mathsf{OPT}(I) + \frac{m-1}{m}\,\mathsf{OPT}(I) = (2 - \frac{1}{m})\,\mathsf{OPT}(I)$. Cette inégalité est vérifiée pour toute machine dont la charge est mise à jour, et elle reste donc vérifiée quand sa charge n'est pas mise à jour. □

3.3.3 Ordonnancement de tâches

On a une séquence de n tâches qui sont révélées les unes après les autres. La tâche i est révélée à l'instant t_i et a un temps de traitement égal à p_i. On dispose de m machines

identiques. Chaque tâche doit être traitée par l'une quelconque des machines. L'objectif est de minimiser la moyenne des temps d'achèvement des tâches, où le temps d'achèvement d'une tâche est défini comme la différence entre l'instant d'achèvement et l'instant de révélation de la tâche.

C'est un problème en ligne de nature temporelle : il peut d'ailleurs être bénéfique de laisser parfois des machines inactives pour gagner de l'information sur les tâches futures.

Il peut être facilement démontré [8, chapitre 9] qu'on ne peut avoir d'algorithme qui fasse mieux que $\mathsf{ALG}(I) \leqslant (n-1)\,\mathsf{OPT}(I)$ pour toute instance I, où n est la taille de l'instance I. Il n'existe donc aucune constante c pour laquelle l'algorithme est c-compétitif, et en particulier il n'y a pas de ratio de compétitivité.

3.3.4 Voyageur de commerce en ligne

On a un véhicule en o qui doit visiter des points $p_1, p_2, \ldots, p_n$ d'une région donnée. La position du point p_i est révélée à l'instant t_i et on a $t_1 < t_2 < \cdots < t_n$. Le nombre n de points n'est pas connu à l'avance. Pour toute paire de points x, y, la durée $d(x, y)$ pour aller de x à y peut être calculée à la demande, et l'on suppose que la durée $d(x, y)$ est une *distance* au sens mathématique (symétrie, séparation, inégalité triangulaire). L'objectif est de terminer au plus tôt la visite de tous les points.

C'est un problème en ligne de nature temporelle. Il peut être démontré qu'il admet un ratio de compétitivité de 2.5 (voir Exercice 8 de la feuille sur les algorithmes en ligne).

3.4 Approche pratique

Le travail théorique sur les problèmes en ligne, en particulier l'analyse de compétitivité, fournit des outils conceptuels pour appréhender les algorithmes en ligne, discuter leurs qualités, les comparer entre eux, etc. Cela permet aussi de guider la construction d'algorithmes performants en pratique. Cependant, des notions comme le ratio de compétitivité sont souvent d'un intérêt opérationnel limité. Contrairement à l'optimisation hors ligne, où les méthodes théoriques ou académiques sont pertinentes en pratique (programmation mathématique, algorithmes polynomiaux, etc.), le développement d'un cadre théorique pour l'optimisation en ligne pertinent pour les applications reste un problème largement ouvert. Une des explications est probablement la complexité mathématique redoutable à mettre en œuvre pour modéliser fidèlement les problèmes d'optimisation en ligne concrets.

Dans cette section, on discute d'abord brièvement de stratégies algorithmiques possibles dans un contexte pratique puis on procède à une étude détaillée d'une résolution possible d'un problème de job-shop temps réel.

3.4.1 Quelques stratégie algorithmiques possibles

On peut distinguer quatre stratégies pour traiter un problème d'optimisation en ligne concret. Ces stratégies forment plus un guide pour aiguiller la réflexion que des méthodes "clé en main".

Fifo (*first-in-first-out*)

Fifo satisfait les requêtes dans l'ordre d'arrivée. Se donner la possibilité d'utiliser cette stratégie ne fait sens que dans le modèle temporel (le modèle séquentiel étant lui nécessairement FIFO).

Quelques caractéristiques :

— rarement optimal.

— simple à mettre en œuvre, si on fait abstraction de la façon dont la requête est satisfaite.

— compatible avec le temps réel.

— maintient l'ordre d'arrivée.

Greedy

Dans le modèle séquentiel, Greedy choisit de satisfaire la requête de manière à minimiser un coût local (allant dans le sens du critère global à minimiser). L'algorithme "glouton" de la section 3.3.2 est un exemple typique. Dans le modèle temporel, Greedy choisit de satisfaire parmi toutes les requêtes non encore satisfaites celle minimisant un coût local (toujours dans le sens du critère global à minimiser).

Quelques caractéristiques :

— rarement optimal (mais mieux que Fifo ?).

— simple à mettre en œuvre.

— compatible avec le temps réel.

— reste prédictible.

Replan

Cette stratégie ne fait sens que dans le modèle temporel. À chaque arrivée d'une requête, Replan calcule la solution optimale hors ligne sur l'ensemble des requêtes non satisfaites, puis agit selon cette solution, jusqu'à la prochaine arrivée d'une requête.

Quelques caractéristiques :

— peut fournir des solutions de très bonne qualité.

— peut être très coûteux en temps de calcul (dépend de la nature du problème hors ligne).

— peut être très imprédictible.

Ignore

Cette stratégie ne fait sens que dans le modèle temporel. Ignore calcule la solution optimale hors ligne sur l'ensemble des requêtes non satisfaites, puis agit selon cette solution, jusqu'à ce que toutes les requêtes de cette solution soient satisfaites.

Quelques caractéristiques :

— peut fournir des solutions de très bonne qualité (mais moins bien que Replan ?).

— peut être très coûteux en temps de calcul (dépend de la nature du problème hors ligne).

— reste relativement prédictible.

Sur le voyageur de commerce temps réel

Regardons ce que donnent ces quatre stratégies sur le problème du voyageur de commerce en ligne de la section 3.3.4.

FIFO : visite les p_i dans l'ordre d'apparition (très mauvais).

GREEDY : va toujours vers le p_i le plus proche non visité (mauvais, mais peut être acceptable).

REPLAN : recalcule une tournée optimale à chaque arrivée d'un point ; suit cette tournée jusqu'à l'arrivée d'un nouveau point.

IGNORE : se dirige vers p_1 ; une fois en p_1, calcule une tournée optimale sur tous les points arrivés pendant ce premier trajet ; suit cette tournée : etc.

3.4.2 Exemple d'étude : job-shop temps réel

Description du problème

Dans un tel problème, on se donne un atelier regroupant un ensemble M de machines. On a des tâches (penser produits) à réaliser sur ces machines. Une *tâche* j est formée d'une suite de $n(j)$ opérations. La $i$ème *opération* doit être effectuée sur la machine $m_i^j \in M$ et requiert un *temps de traitement* $p_i^j \in \mathbb{R}_+$. On suppose de plus qu'une machine ne peut effectuer au plus qu'une opération à la fois, et qu'une opération, une fois commencée, ne peut être interrompue.

On se met dans un contexte en ligne temporel : les tâches arrivent les unes après les autres, à des instants aléatoires, et leurs caractéristiques ne sont révélées qu'à leurs arrivées. On suppose de plus que l'atelier fonctionne en régime stationnaire et que les arrivées des tâches ne connaissent pas d'interruption. (Il y a donc potentiellement un nombre infini de tâches.)

De nombreux objectifs sont envisageables. Ici, on va considérer deux critères classiques :

— minimiser le temps moyen $\overline{F}$ de complétion d'une tâche.

— minimiser le temps maximum $F_{\max}$ de présence d'une tâche dans l'atelier.

Enjeu

Si les temps d'inter-arrivées sont grands par rapport aux temps de traitement, les machines sont quasiment toujours disponibles et lorsqu'une opération d'une tâche se termine, l'opération suivante de cette tâche peut être immédiatement effectuée. Au pire, il y aura quelques tâches en attente devant certaines machines, mais aucune décision particulière n'aura à être prise.

En revanche, lorsque les temps d'inter-arrivées sont comparables aux temps de traitement, des files d'attente se créent devant les machines, et chaque fois qu'une machine termine une

opération, il doit être décidé, parmi toutes les tâches en attente, la prochaine à être traitée par la machine. Comme le temps de prise de décision contribue directement aux critères évalués, on est clairement dans un contexte temps réel.

Règles de priorité

Une façon naturelle de prendre ces décisions est d'appliquer des "règles de priorité". Une règle de priorité est une méthode heuristique choisissant la prochaine tâche traitée par une machine, parmi celles en attente, par un calcul simple prenant en input les caractéristiques courantes des tâches dans l'atelier. Le temps de calcul d'une telle règle est souvent très faible (constant ou logarithmique). La simplicité et la rapidité du calcul permettent une mise en œuvre aisée de ces règles et les rendent particulièrement adaptées dans un contexte de temps réel.

Voici quelques exemples de ces règles classiques, avec leurs appellations habituelles. Avec la terminologie de la section 3.4.1, elles sont toutes du type FIFO ou GREEDY. Elles associent à l'instant courant une quantité Z_j à toute tâche j. Lorsqu'une machine m se libère, la tâche j en attente avec le plus petit Z_j devient la prochaine tâche traitée par m.

Voici quelques façons de calculer des Z_J. On considère une tâche j en attente devant une machine m. On note T_j son instant d'arrivée dans l'atelier et i l'opération qu'elle doit subir sur m. Enfin, on pose

$$W_j = \sum_{\substack{\text{tâches } k \text{ en attente} \\ \text{devant } m = m_{i+1}^j}} p_{i(k)}^k \,,$$

où $i(k)$ est l'indice de l'opération correspondant au passage de la tâche k sur la machine m. En particulier, si j est en attente de sa dernière opération, on a $W_j = 0$.

- FIFO (*first-in first-out*) : $Z_j =$ instant d'arrivée de la tâche j dans la queue.
- AT (*arrival time*) : $Z_j = T_j$.
- SPT (*shortest process time*) : $Z_j = p_i^j$.
- PT+WINQ (*process time plus work-in-next-queue*) : $Z_j = p_i^j + W_j$.
- PT+WINQ+AT (*process time plus work-in-next-queue plus arrival time*) : $Z_j = p_i^j + W_j + T_j$.
- AT−RPT (*arrival time moins total remaining process time*) : $Z_j = T_j - \sum_{i'=i}^{n(j)} p_{i'}^j$.

Expériences

Pour comparer les performances des différentes règles, on procède ici de manière expérimentale, par simulation. Les résultats sont tirés du travail de Rajendran et Holthaus [17]. Le système simulé est formé d'un atelier de 10 machines ayant à traiter 2500 tâches arrivant selon un processus de Poisson. Les caractéristiques de chaque tâche sont tirées au hasard :

- ordre de visite : tiré uniformément au hasard parmi les permutations possibles.

— durée de chaque opération de la tâche : un nombre entier de secondes tiré uniformément au hasard parmi $1, 2, \ldots, 49$.

Deux durées moyennes d'inter-arrivées $1/\lambda$ sont testées : $\lambda = 0.032$ et $\lambda = 0.038$. On a alors selon les valeurs choisies pour λ :

$$\lambda \times 25 = 0.8 < 1 \qquad \text{et} \qquad \lambda \times 25 = 0.95 < 1\,,$$

où 25 correspond à la durée moyenne d'une opération sur une machine. Avec les notations et les définitions de la section 2.3.2, cela assure heuristiquement la stabilité du système car $\rho < 1$ et une utilisation $\bar{u}$ de chaque machine de 0.8 et de 0.95 selon le choix de λ (voir corollaire 2.10).

Chacune des tables 3.1 rassemble les résultats obtenus pour les six règles présentées ci-dessus pour un des deux taux d'arrivées. Dans chacun des deux tableaux, chaque règle a été testée sur 20 réplications de la simulation. Pour chaque réplication, on évalue le critère sur les 2000 dernières tâches (les 500 premières sont utilisées pour amener le système en régime permanent). C'est la moyenne de la valeur de ce critère sur les 20 réplications qui est donnée.

Règle	$\overline{F}$	$F_{\max}$
FIFO	839.1	1962.0
AT	811.3	1536.8
SPT	633.6	3777.6
PT+WINQ	630.2	2703.2
PT+WINQ+AT	761.3	1479.2
AT−RPT	809.9	1422.7

(a) Pour $\lambda = 0.032$

Règle	$\overline{F}$	$F_{\max}$
FIFO	2418.2	4452.4
AT	1958.0	2978.4
SPT	1466.8	25875.4
PT+WINQ	1428.8	12306.1
PT+WINQ+AT	1785.8	2862.1
AT−RPT	1956.8	2853.4

(b) Pour $\lambda = 0.038$

TABLE 3.1 – Résultats pour les différentes règles de priorité selon la valeur de λ.

Conclusions

Sur ces expériences, les règles SPT et PT+WINQ assurent le meilleur temps moyen $\overline{F}$ de complétion et sont plutôt mauvaises pour le temps maximum $F_{\max}$ de complétion. Les règles PT+WINQ+AT et AT−RPT présentent un comportement opposé. Noter que AT, conformément à l'intuition, se comporte de manière correcte pour $F_{\max}$. Enfin, FIFO est dominé par AT sur les deux critères. Ces conclusions sont connues pour être valables pour de nombreux problèmes de job-shop.

Le message plus général est le suivant. Pour un problème d'optimisation temps réel dont la modélisation est difficile, une approche scientifique est tout à fait possible. Les règles de priorité, et plus généralement les heuristiques, peuvent être conçues à partir d'une bonne intuition du problème (construite par exemple à partir de "modèles-jouets") et évaluées rigoureusement par simulation.

3.5 Compléments bibliographiques

L'article déjà cité de Fiat et Woeginger [8] est plutôt ancien, mais aborde différents aspects de l'optimisation en ligne : pertinence théorique et appliqué, enjeu, notions fondamentales. Un ouvrage classique et très complet, mais de nature résolument théorique, est celui de Borodin et El-Yaniv [5]. L'ouvrage de Jaillet et Wagner [10] prend une approche plus probabiliste et présente de nombreuses applications pertinentes pour l'industrie.

CHAPITRE 4

Gestion de stock sous aléa

4.1 Introduction

Dans de nombreux contextes industriels, l'approvisionnement ou la production de biens étant soumis à des délais ou des incertitudes, la capacité à satisfaire une demande ou un besoin futur dans de bonnes conditions requiert la mise en place de stocks. Les stocks peuvent avoir d'autres intérêts : couverture face à des incertitudes sur les prix à l'achat, lissage de la production, économie d'échelle, etc. Augmenter la taille de stock améliore la prise en compte de ces critères. D'un autre côté, les stocks induisent des coûts (argent immobilisé, assurance, espace de stockage, gestion, suivi, etc.). L'enjeu est donc d'avoir des stocks répondant aux besoins, tout en gardant un coût raisonnable.

On distingue traditionnellement deux types d'examen des stocks : l'examen continu et l'examen périodique. Dans le cadre de l'examen continu, l'approvisionnement peut se faire n'importe quand. Le modèle historique est le *modèle de Wilson*. Dans le cadre de l'examen périodique, l'approvisionnement ne peut se faire qu'à des moments précis, prévus à l'avance. C'est ce dernier type qui sera étudié dans ce chapitre, dans un contexte stochastique. Nous verrons d'abord le cas à une période, connu également sous le nom du problème de vendeur de journaux, puis le cas à N périodes.

4.2 Problème à une période, ou vendeur de journaux

4.2.1 Problème

Formalisation

On considère une entreprise qui dispose d'un stock initial s_0 d'un bien. Elle choisit un niveau de réapprovisionnement, puis satisfait une demande aléatoire dans la limite du stock disponible. Le coût se décompose en coût de commande, coût de stockage et coût de pénurie (équivalent à considérer le gain sur les ventes, voir plus bas). En notant s le niveau de stock choisi par l'entreprise après réapprovisionnement et D la demande (aléatoire), le coût (aléatoire) s'écrit

$$c \cdot (s - s_0) + h \cdot (s - D)^+ + p \cdot (D - s)^+ , \tag{4.1}$$

où $c, h, p \in \mathbb{R}$ sont respectivement les coûts unitaires de commande, stockage et pénurie. On suppose de plus que l'entreprise souhaite minimiser l'espérance de son coût. (En présence

d'aléa, selon le contexte, différentes façons de construire l'objectif sont envisageables ; idéalement, l'objectif doit dépendre des variables aléatoires via une *mesure de risque*.)

On va supposer a priori que le bien est suffisamment "divisible" pour que l'on soit autorisé à voir les quantités comme des nombres réels non nécessairement entiers. Nous discuterons plus loin le cas où cette hypothèse ne peut être faite. Le problème consiste donc à résoudre

$$\begin{array}{ll} \text{Min} & \mathbb{E}[\phi(s,D)] \\ \text{s.c.} & s \geqslant s_0 \\ & s \in \mathbb{R}\,, \end{array} \tag{4.2}$$

avec $\phi(s,D) = c\cdot s + h\cdot(s-D)^+ + p\cdot(D-s)^+$. Ici, on n'a choisi de ne pas faire figurer le terme $-cs_0$ dans la fonction objectif car c'est un terme constant et il ne joue donc pas de rôle dans la résolution du problème. Noter que c'est un problème d'optimisation en dimension 1 : il n'y a qu'une seule variable, s. On peut donc espérer une solution complète du problème, et nous verrons que c'est dans un certain sens le cas. Nous discutons dans les sous-sections suivantes la façon de résoudre ce problème, en fonction de la loi de D, et nous considérerons aussi la version discrète, où les niveaux de stock et la demande ne peuvent prendre que des valeurs entières.

Et s'il y a un gain à la vente au lieu d'un coût de pénurie ?

Nous finissons cette sous-section en discutant un coût alternatif, qui peut sembler plus naturel. Supposons que le coût soit en fait de la forme

$$c\cdot(s-s_0) + h\cdot(s-D)^+ - g\cdot\min(D,s)\,, \tag{4.3}$$

où c et h ont la même interprétation que dans l'équation (4.1), et où g est le prix de vente d'une unité du bien. Dans cette version, on considère qu'il n'y a pas de coût de pénurie mais un gain lié aux ventes. Le coût (4.3) se réécrit

$$c\cdot(s-s_0) + h\cdot(s-D)^+ + g\cdot(D-s)^+ - g\cdot D\,.$$

Le problème d'optimisation que l'on obtient avec ce nouveau coût est de la forme (4.1), à la constante $g\mathbb{E}[D]$ près. L'objectif (4.1) est donc suffisamment général pour couvrir cette version alternative du problème (et d'autres d'ailleurs).

4.2.2 Solution

On fait deux hypothèses :

(i) $\mathbb{E}[D] < +\infty$.

(ii) $-h < c < p$.

Si on lève la première hypothèse, l'espérance du coût vaut $+\infty$ quoi qu'on fasse (sauf si $p = 0$, auquel cas la solution est un peu plus riche). Si l'on est confronté en pratique à un cas où $\mathbb{E}[D] = +\infty$, cela montre qu'il faut peut-être revoir la modélisation du problème, en

particulier le choix de la fonction de coût. La seconde hypothèse est totalement naturelle en pratique : $c + h > 0$ signifie qu'on ne gagne pas d'argent à simplement commander et stocker ; $p - c > 0$ qu'on a intérêt à satisfaire la demande. La résolution du problème lorsque ces hypothèses ne sont pas satisfaites est discutée en exercice.

On a

$$\phi(s, D) = \max((c + h) \cdot s - h \cdot D, (c - p) \cdot s + p \cdot D) \,. \tag{4.4}$$

En effet, si $s \geqslant D$, on a $\phi(s, D) = (c+h)s-hD$ par définition, ce qui est également donné par l'équation (4.4) car $h > -p$ d'après l'hypothèse (ii). Et si $s \leqslant D$, on a $\phi(s, D) = (c-p)s+pD$ par définition, ce qui est également donné par l'équation (4.4), toujours car $h > -p$ d'après l'hypothèse (ii).

Lemme 4.1. *La fonction $s \mapsto \mathbb{E}[\phi(s, D)]$ est convexe et coercive.*

Démonstration. Le maximum de fonctions convexes (*a fortiori* affines) est convexe. L'équation (4.4) montre donc la convexité à D fixé de la fonction $s \mapsto \phi(s, D)$. L'inégalité de convexité est préservée en passant à l'espérance et donc $s \mapsto \mathbb{E}[\phi(s, D)]$ est convexe.

Pour la coercivité, on se sert encore de l'équation (4.4) : l'espérance d'un maximum étant toujours plus grand que le maximum des espérances, on a

$$\mathbb{E}[\phi(s, D)] \geqslant \max((c + h)s - h\mathbb{E}[D], (c - p)s - p\mathbb{E}[D]) \geqslant \max((c + h)s, (c - p)s) - \max(|h|, |p|)\,|\mathbb{E}[D]| \,.$$

Les hypothèses impliquent alors la coercivité : $\lim_{|s|\to+\infty} \mathbb{E}[\phi(s, D)] = +\infty$. □

Le lemme 4.1 montre que le problème (4.2) admet toujours une solution optimale : une fonction continue coercive atteint toujours un minimum global sur un fermé non vide. (Rappelons qu'une fonction convexe est continue sur l'intérieur de son domaine.) Cette solution optimale admet une expression simple.

Théorème 4.2. *Posons*

$$s^* = \max(s_0, S) \qquad \textit{avec} \quad S = \min F^{-1}\left(\left[\frac{p - c}{p + h}, 1\right]\right) \,,$$

où F est la fonction de répartition de D. Alors s^* est solution optimale du problème (4.2). Cette solution est unique si F est strictement croissante sur un voisinage à droite de S.

Notons que s^* est bien défini car $F^{-1}\left(\left[\frac{p-c}{p+h}, 1\right]\right)$ est toujours non vide (c'est une conséquence de la définition des fonctions de répartition). De plus, on peut écrire min et non simplement inf car F est toujours continue à droite (à nouveau par définition des fonctions de répartition). Remarquer que dans le cas particulier où F est inversible, on a $S = F^{-1}\left(\frac{p-c}{p+h}\right)$. C'est un cas qui se rencontre fréquemment, par exemple, si D est une variable aléatoire à densité strictement positive presque partout sur son support. C'est le cas en particulier des variables aléatoires gaussienne, uniforme, exponentielle, triangulaire (voir Section 6.3.1), etc.

La politique optimale consiste donc à commander $S - s_0$ unités si $S > s_0$ et rien sinon. C'est une *politique à une valeur* : la quantité S détermine entièrement ce qu'il faut faire pour agir optimalement ; voir figure 4.1. C'est l'un des principaux messages du théorème 4.2.

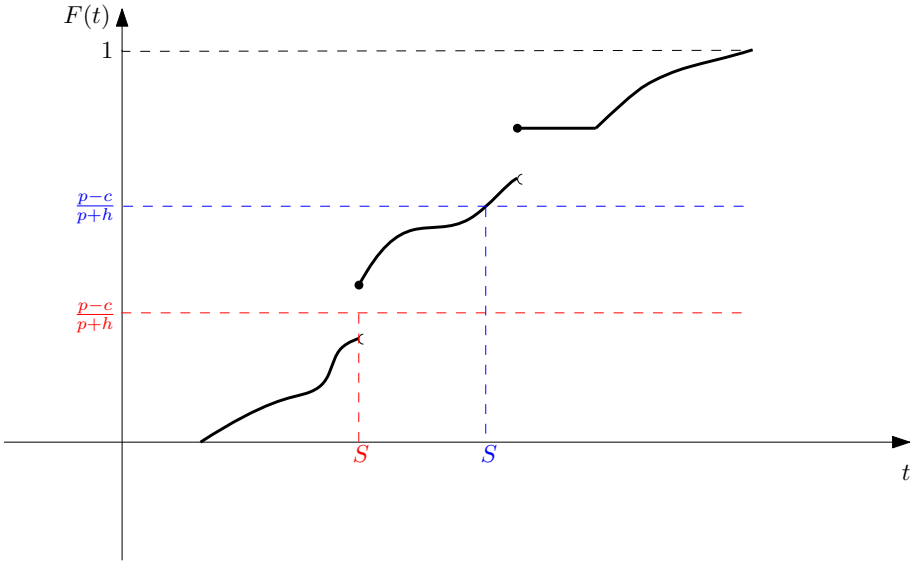

FIGURE 4.1 – Politique à une valeur : La position du stock initial s_0 par rapport au seuil S détermine si l'on amène le niveau de stock à S ou si l'on le laisse tel quel. En rouge et en bleu, deux exemples de situations possibles selon la valeur de $\frac{p-c}{p+h}$.

Même si l'expression donnant S s'écrit sous une forme élégante, il n'est malheureusement pas toujours facile de calculer S : cela dépend de la loi de D.

Un élément clé de la démonstration du théorème 4.2 est l'égalité $\mathbb{E}[(s-D)^+] = \int_0^s F(t)\,\mathrm{d}t$. Cette égalité est immédiate : on a en effet

$$\int_0^s F(t)\,\mathrm{d}t = \int_0^s \mathbb{P}(D \leqslant t)\,\mathrm{d}t = \int_0^s \mathbb{E}[\mathbf{1}(D \leqslant t)]\,\mathrm{d}t = \mathbb{E}\left[\int_0^s \mathbf{1}(D \leqslant t)\,\mathrm{d}t\right] = \mathbb{E}[(s-D)^+]\,.$$

(Le théorème de Fubini s'applique ici et permet d'intervertir l'intégrale et l'espérance.) Cette égalité est même utile au-delà de la preuve car elle donne un façon de calculer l'espérance du coût pour un choix donné du niveau de stock : comme $\phi(s,D)$ peut se réécrire $(c-p)s + pD + (p+h)(s-D)^+$, on a

$$\mathbb{E}[\phi(s,D)] = (c-p)s + p\mathbb{E}[D] + (p+h)\int_0^s F(t)\,\mathrm{d}t\,.$$

Le théorème 4.2 se prouve en explicitant la condition d'optimalité du premier ordre sur cette expression.

Démonstration du théorème 4.2. Même non différentiable, une fonction convexe d'une variable réelle admet toujours des dérivées à droite et à gauche en tout point de l'intérieur de son domaine. Un point est un minimum global d'une fonction convexe si et seulement si c'est l'infimum des points où la dérivée à droite est positive ou nulle. Comme la dérivée à droite de $s \mapsto \mathbb{E}[\phi(t,D)]$ est égale à $c - p + (p+h)F(s)$ (ici, on utilise la continuité à droite de F), le point $S = \min F^{-1}\left(\left[\frac{p-c}{p+h}, 1\right]\right)$ minimise $s \mapsto \mathbb{E}[\phi(t,D)]$. Si $S \geqslant s_0$, le point $s^* = S$ est une solution réalisable et donc optimale du problème (4.2). Si $S < s_0$, le point $s^* = s_0$ est solution optimale (la fonction $s \mapsto \mathbb{E}[\phi(t,D)]$ étant croissante sur $[S, +\infty[$ par convexité).

Le cas d'unicité se déduit selon les mêmes arguments. □

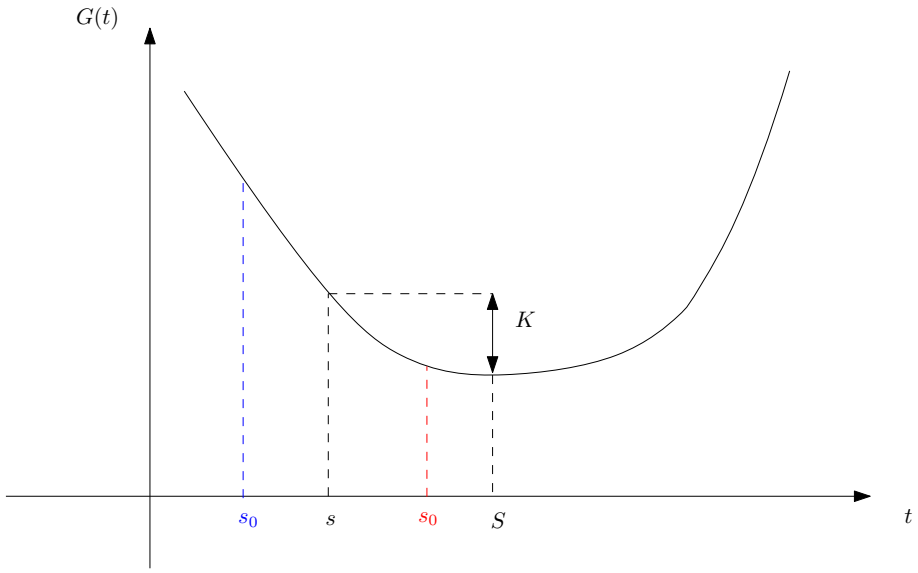

FIGURE 4.2 – Politique à deux valeurs : La position du stock initial s_0 par rapport au seuil s détermine si l'on amène le niveau de stock à S ou si l'on le laisse tel quel. En rouge et en bleu, deux exemples de situations possibles selon la valeur de s_0.

4.2.3 Extension : cas avec coût fixe d'approvisionnement

Sous les hypothèses (i) et (ii), le théorème 4.2 permet de traiter l'extension du problème du vendeur de journaux quand il y a un coût fixe d'approvisionnement : lorsqu'on passe une commande strictement positive, en plus du coût unitaire c, on paye un coût fixe $K > 0$. Amener le stock à un niveau t prend alors la forme [1]

$$K\mathbf{1}(t > s_0) + c(t - s_0) + h(t - D)^+ + p(D - t)^+ . \tag{4.5}$$

Posons $G(t) = \mathbb{E}[\phi(t, D)]$. En reprenant le niveau S défini comme dans le théorème 4.2 et s défini comme $\inf G^{-1}(K + G(S))$, la politique consistant à amener le stock au niveau S si $s_0 < s$ et à ne rien faire sinon est optimale ; voir figure 4.2. (Ici, la convexité et la coercivité permettent de conclure.) C'est une *politique à deux valeurs*. On parle aussi de *politique* (s, S).

4.2.4 Extension : cas avec niveaux de stock entiers

Sous les hypothèses (i) et (ii), le théorème 4.2 permet aussi de traiter l'extension du problème du vendeur de journaux quand les quantités du bien sont contraints à être entiers, c'est-à-dire quand le bien n'est plus "divisible" arbitrairement. En effet, puisque dans ce cas F ne change que sur des entiers, le niveau S donné par le théorème est entier. De plus, le cas avec coût fixe d'approvisionnement est également couvert : s n'est pas nécessairement entier mais cela n'empêche pas son application.

1. On remplace la notation s par t parce que la lettre s a une signification particulière dans ce contexte, comme on peut le voir quelques lignes plus bas.

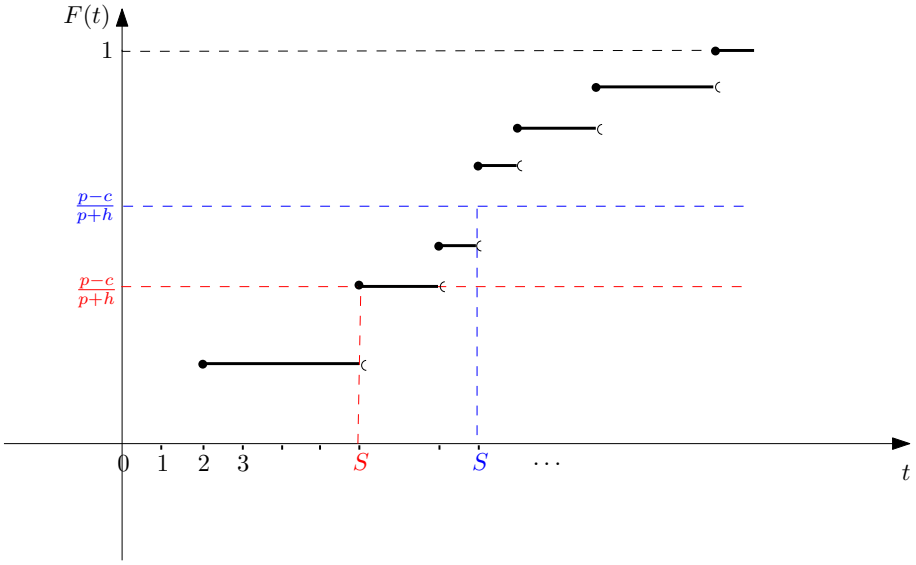

FIGURE 4.3 – Politique à une valeur, cas entier : La position du stock initial s_0 par rapport au seuil S détermine si l'on amène le niveau de stock à S ou si l'on le laisse tel quel. En rouge et en bleu, deux exemples de situations possibles selon la valeur de $\frac{p-c}{p+h}$.

4.2.5 Nombre fini de scénarios

En pratique D est souvent supposée ne prendre qu'un nombre fini de valeurs. En supposant que D prend la valeur d_i avec la probabilité p_i, le problème du vendeur de journaux (toujours sous les hypothèses (i) et (ii)) peut se modéliser sous la forme du programme linéaire (en variables continues) suivant :

$$\begin{array}{lll} \text{Min} & \displaystyle\sum_i p_i v_i & \\ \text{s.c.} & v_i \geqslant (c+h)x - cs_0 - hd_i & \forall i \\ & v_i \geqslant (c-p)x - cs_0 + pd_i & \forall i \\ & v_i \in \mathbb{R} & \forall i \\ & x \geqslant s_0\,. & \end{array}$$

Cela se montre directement à partir de l'équation (4.4).

Dans le cas où D ne prend qu'un nombre fini de valeurs, on peut donc résoudre le problème de manière efficace.

4.3 Problème à N périodes

4.3.1 Problème

On considère à nouveau une entreprise qui dispose d'un stock initial t_1 d'un bien. La gestion du stock est maintenant vue sur N périodes. En début de chaque période i, l'entreprise

doit fixer un niveau u_i de commande. Ensuite, un niveau de demande D_i est révélé. En notant t_i le niveau de stock juste avant le début de période i, on a

$$t_{i+1} = t_i + u_i - D_i \,. \tag{4.6}$$

Dans un tel modèle, la demande non satisfaite pendant une période est reportée sur la période suivante.

Le coût total sur la période i est donné par

$$K\mathbf{1}_{u_i>0} + cu_i + h(t_i + u_i - D_i)^+ + p(D_i - t_i - u_i)^+ \,, \tag{4.7}$$

où $c, h, p, K \in \mathbb{R}$ gardent les mêmes significations qu'en section 4.2. On se place d'emblée dans le cas avec un coût fixe d'approvisionnement. Noter—et c'est un choix de modélisation—que les coûts de stockage et de pénurie sont calculés avec le niveau de stock de la fin de période.

On suppose toujours que l'entreprise souhaite minimiser l'espérance de son coût et que les quantités peuvent être vues comme des nombres réels non nécessairement entiers. De plus, on suppose que l'entreprise connaît le niveau de stock courant et que les demandes D_i sont indépendantes. Une politique (déterministe) de gestion de stock peut donc ne dépendre que du niveau de stock à la fin de la période précédente. En posant

$$\psi(t, u, D) = K\mathbf{1}_{u>0} + cu + h(t + u - D)^+ + p(D - t - u)^+ \,,$$

le problème consiste donc à résoudre

$$\begin{array}{lll} \text{Min} & \mathbb{E}\left[\sum_{i=1}^N \psi(t_i, \mu_i(t_i), D_i)\right] & \\ \text{s.c.} & t_{i+1} = t_i + \mu_i(t_i) - D_i & \forall i \in [N-1] \\ & \mu_i \colon \mathbb{R} \to \mathbb{R}_+ & \forall i \in [N] \,. \end{array} \tag{4.8}$$

Les variables sont les t_i pour $i = 2, \dots, N$ (le niveau de départ t_1 étant fixé *a priori*), ainsi que les fonctions μ_i pour $i = 1, \dots, N$ qui prennent en argument le stock t_i à la fin de la période précédente et renvoie le niveau u_i auquel il faut faire la commande. Dans le cas $N = 1$, on retrouve bien la problème du vendeur de journaux étudié dans la section 4.2.

4.3.2 Solution

On se place toujours sous les hypothèses (i) et (ii) : pour la demande, cela signifie qu'en plus d'être indépendantes, les D_i sont d'espérance bornée. On suppose de plus que les quatre paramètres c, h, p et K sont positifs.

En notant $J_i(t)$ la valeur optimale de l'espérance du coût sur les périodes $i, \dots, N$, on a l'équation de programmation dynamique suivante :

$$J_i(t) = \inf_{u \geqslant 0} \mathbb{E}[\psi(t, u, D_i)] + \mathbb{E}[J_{i+1}(t + u - D_i)] \,. \tag{4.9}$$

Les problèmes d'optimisation convexe en dimension 1 étant faciles à résoudre, au moins numériquement, le lemme suivant montre que la programmation dynamique conduit à une méthode de résolution pratique du problème dans le cas sans coût fixe ($K = 0$). En utilisant la convexité de la fonction objectif, sa preuve s'appuie les résultats de la section 4.2.

Lemme 4.3. *Supposons $K = 0$, et soit $i \in [N]$. La fonction $t \mapsto J_i(t)$ est alors convexe et coercive. De plus, il existe $S_i \geqslant 0$ tel que, pour tout réel t, l'infimum dans l'équation (4.9) est atteint pour $u = (S_i - t)^+$.*

Démonstration. D'après le théorème 4.2, il existe S_N tel que l'infimum dans l'équation (4.9) pour $i = N$ est atteint pour $u = (S_N - t)^+$. En posant

$$L_i(t) = h\mathbb{E}[(t - D_i)^+] + p\mathbb{E}[(D_i - t)^+]\,,$$

on a alors

$$J_N(t) = \begin{cases} c(S_N - t) + L_N(S_N) & \text{si } t < S_N \\ L_N(t) & \text{sinon.} \end{cases} \tag{4.10}$$

La fonction J_N est convexe : c'est vrai pour $t < S_N$ et $t \geqslant S_N$, et une discussion sur les dérivées à gauche et à droite en S_N permet de le vérifier pour tout t. Il est de plus facile de vérifier à partir de l'équation (4.10) que J_N est coercive.

Supposons maintenant que J_{i+1} est convexe et coercive. Comme c, h et p sont positifs, J_{i+1} est toujours positif et en utilisant la coercivité de $s \mapsto cs + L_i(s)$ (conséquence du lemme 4.1), on voit que $s \mapsto cs + L_i(s) + \mathbb{E}[J_{i+1}(s - D_i)]$ est convexe (donc continue) et coercive. Cette dernière fonction admet donc un minimum global en un certain point S_i. L'infimum dans l'équation (4.9) est donc atteint pour $u = (S_i - t)^+$. On a donc

$$J_i(t) = \begin{cases} c(S_i - t) + L_i(S_i) + \mathbb{E}[J_{i+1}(S_i - D_i)] & \text{si } t < S_i \\ L_i(t) & \text{sinon.} \end{cases} \tag{4.11}$$

On peut alors vérifier à partir de ces équations que J_i est convexe et coercive. □

Ce lemme montre de plus que la politique optimale continue dans le cas sans coût fixe ($K = 0$) à prendre une forme simple, très proche du cas $N = 1$. On peut exprimer ce fait sous la forme du théorème suivant, prouvé par Bellman, Glicksberg et Gross en 1955 [4].

Théorème 4.4. *Pour le modèle sans coût fixe ($K = 0$), il existe une suite de valeurs $S_1, S_2, \ldots, S_N$ telle que $\mu_i^*(t) = (S_i - t)^+$ est solution optimale du problème (4.8).*

En d'autres termes, chaque période ressemble au cas à une période : la politique optimale est à une valeur sur chacune des périodes.

Le cas avec coût fixe ($K > 0$) est plus complexe mais il existe un théorème dans le même esprit que le théorème 4.4 et qui affirme que pour la politique optimale, chaque période ressemble au cas à une période, c'est-à-dire que la politique optimale est à deux valeurs sur chaque période. C'est le théorème suivant, prouvé par Scarf en 1959 [18], et qui est un des grands classiques de la théorie de la gestion des stocks. Il avait été conjecturé par la plupart des spécialistes de la gestion de stock, tant théoriciens qu'opérationnels. La preuve du cas sans coût fixe ne s'adapte pas directement car la présence du coût fixe fait disparaître les propriétés de convexité. L'étape essentielle a été l'introduction d'une généralisation de la convexité pour les fonctions d'une variable réelle, appelée *K-convexité*. Cette introduction a permis de réutiliser les autres idées de la preuve du théorème 4.4. La preuve étant difficile, elle est omise.

Théorème 4.5 (Théorème de Scarf)**.** *Pour le modèle avec coût fixe ($K > 0$), il existe une suite de couples $(s_1, S_1), (s_2, S_2), \ldots, (s_N, S_N)$ telle que*

$$\mu_i^*(t) = \begin{cases} S_i - t & \text{si } t < s_i \\ 0 & \text{si } t \geqslant s_i \end{cases}$$

est solution optimale du problème (4.8).

Une question qui ne semble cependant pas avoir été traitée est de savoir si l'équation (4.9) conduit, tout comme dans le cas sans coût fixe, à une méthode de résolution pratique efficace du problème. En effet, alors que minimiser une fonction convexe est aisé à faire numériquement, il n'est pas du tout clair que cela reste le cas pour les fonctions K-convexes.

4.4 Compléments bibliographiques

Un ouvrage classique sur la question de la gestion de stock, déterministe et stochastique, est le livre de Zipkin [22]. Bien qu'un peu vieux—il date de 2000—il reste une référence sur le sujet.

CHAPITRE 5

Revenue management

5.1 Introduction

Le revenue management est l'un des apports les plus spectaculaires de la recherche opérationnelle au monde socio-économique. Pour une entreprise vendant un bien ou un service, le revenue management propose un ensemble de modèles et de méthodes permettant de fixer le prix de vente afin de maximiser le profit. Pour pouvoir être appliqué avec succès, il requiert cependant un certain nombre de conditions, comme une bonne compréhension de la demande. Lorsque ces conditions sont satisfaites, les gains peuvent être importants tout en ne nécessitant que des changement marginaux et quasiment sans coût : essentiellement, il s'agit simplement d'appliquer certaines formules mathématiques pour déterminer les prix.

L'idée de départ est la suivante. Dans de nombreuses situations, le montant auquel un client est prêt à payer le bien ou le service mis en vente dépend du client. Le profit sera d'autant plus grand que l'entreprise aura été capable de vendre en priorité aux clients prêts à payer le plus et au prix maximal acceptable pour ceux-ci. Comment choisir ces prix et comment les faire varier au cours du temps est la question principale à laquelle s'intéresse le revenue management.

Les domaines où les techniques du revenue management sont appliqués avec succès sont principalement l'hôtellerie et le transport. C'est d'ailleurs dans le transport aérien qu'est née cette approche, au cours des années 80. Les entreprises n'ayant pas su rapidement l'intégrer ont fait faillite et toutes les compagnies aériennes du monde l'appliquent désormais. On considère que de l'ordre de 4 à 5% du chiffre d'affaire d'une compagnie aérienne provient directement des applications des techniques du revenue management, ce qui est à peu près le niveau typique de la marge opérationnelle dans ce secteur.

La situation classique étudiée par le revenue management est celle d'une entreprise vendant un produit périssable (une unité non vendue étant définitivement perdue après une date limite). Les exemples typiques (et historiques) sont ceux des places dans un avion ou des chambres dans un hôtel. Une technique classique du revenue management consiste à fixer au préalable un nombre fini de classes tarifaires (c'est-à-dire un certain nombre de prix possibles pour une unité du bien vendu—par exemple, une place en classe économique dans un avion) et de déterminer la quantité maximale vendue pour chaque classe tarifaire. On parle de *contrôle par la quantité*. Il existe également un *contrôle par les prix*, dont nous ne parlerons pas mais qui est assez proche : les résultats d'une approche peuvent souvent être

transférés à l'autre approche par un simple calcul.

5.2 Cas à deux classes tarifaires

Le modèle à deux classes tarifaires, proposé en 1972 par Littlewood [15], est historiquement le premier modèle de revenue management, et reste par bien des aspects fondamental pour la compréhension de ce domaine. Il offre une certaine similarité avec le problème de vendeur de journaux (et la forme de la solution est d'ailleurs assez proche).

Une entreprise met en vente une quantité $q \geqslant 0$ d'un bien. On a deux classes tarifaires, numérotées 1 et 2. Les clients de la classe 1 sont prêts à acheter le bien à un prix p_1 alors que ceux de la classe 2 sont prêts à l'acheter à un prix p_2. Ces prix sont fixés (par exemple, par le marketing de l'entreprise) et vérifient $p_1 > p_2 > 0$. La demande pour la classe 1 est notée D_1 et celle pour la classe 2 est notée D_2. On suppose que les niveaux de demande D_1 et D_2 sont des nombres réels positifs non nécessairement entiers (cette dernière hypothèse sera discutée plus loin). Ici, D_1 et D_2 sont des variables aléatoires supposées indépendantes.

Le problème consiste à déterminer la quantité maximale s_i qu'on s'autorise à vendre à chacune des classes i afin de maximiser l'espérance du revenu, avec $s_1+s_2 = q$. Une hypothèse souvent faite en pratique, et pas complètement irréaliste, est que les clients de la classe 2, avec un pouvoir d'achat plus faible, vont se manifester avant ceux de la classe 1.

Plutôt que s_1 et s_2, l'habitude veut qu'on utilise plutôt la variable y_1, qui est le nombre d'unités réservées à la classe 1. On dit que y_1 est le *niveau de protection de la classe* 1. On ne s'autorise à vendre à la classe 2 que s'il y a plus de y_1 unités disponibles. Pour une réalisation possible de D_1 et D_2, le revenu est égal à

$$R(q; y_1; D_1, D_2) = p_1 X_1 + p_2 X_2, \quad \text{avec } X_1 = \min(D_1, (q - X_2)^+) \text{ et } X_2 = \min(D_2, (q - y_1)^+)\,.$$

Le problème s'écrit donc :

$$\begin{array}{ll} \text{Max} & \mathbb{E}\left[R(q; y; D_1, D_2)\right] \\ \text{s.c.} & y \geqslant 0\,, \end{array} \tag{5.1}$$

dont la seule variable est y, niveau de protection de la classe 1.

Théorème 5.1. *Soit y^* tel que $\mathbb{P}(D_1 > y^*) = \frac{p_2}{p_1}$. Alors y^* est solution optimale du problème (5.1).*

Le résultat est intuitif. Il dit qu'il faut continuer à vendre à la classe 2 tant que le revenu est supérieur ou égal à l'espérance du revenu si l'on vend désormais à la classe 1 : s'il reste y unités à vendre, le montant p_2 est le revenu marginal à rester sur la classe 2 et le montant $p_1\mathbb{P}(D_1 > y)$ est le revenu marginal à basculer sur la classe 1.

Nous ne démontrons pas le théorème ici car il est conséquence d'un théorème plus général vu dans la section suivante.

5.3 Cas à N classes tarifaires

5.3.1 Modèle

On considère la généralisation naturelle du problème de la section 5.2. On a toujours une quantité q d'un bien. On a maintenant N classes tarifaires, numérotées $1, 2, \dots, N$. Les demandes correspondantes sont des variables aléatoires $D_1, D_2, \dots, D_N$. Les tarifs correspondants, et fixés à l'avance, sont $p_1 > p_2 > \cdots > p_N > 0$. L'objectif est de déterminer les *niveaux de protection* $y_1 \leqslant y_2 \leqslant \cdots \leqslant y_{N-1}$: on accepte de vendre à la classe $i+1$ que s'il reste au moins y_i unités disponibles. Pour une réalisation possible de la demande, et pour un choix des niveaux de protection, le revenu est égal à

$$R(q; y_1, \dots, y_{N-1}; D_1, \dots, D_N) = \sum_{i=1}^{N} p_i X_i \,,$$

avec

$$X_i = \min(D_i, (Q_i - y_{i-1})^+), \qquad Q_i = Q_{i+1} - X_{i+1} \qquad \text{et} \qquad Q_N = q \,.$$

Le nombre X_i est la quantité d'unités vendues à la classe i et Q_i est le nombre d'unités qu'il reste au moment d'ouvrir la vente à la classe i (c'est-à-dire au moment de passer au prix p_i).

Le problème d'optimisation que l'on souhaite résoudre est donc le suivant :

$$\begin{array}{ll} \text{Max} & \mathbb{E}\left[R(q; y_1, \dots, y_{N-1}; D_1, \dots, D_N)\right] \\ \text{s.c.} & y_1, \dots, y_{N-1} \geqslant 0 \,. \end{array} \tag{5.2}$$

5.3.2 Résultats

Pour faciliter les énoncés et les preuves, on introduit les fonctions r_i suivantes définies par

$$r_1(q, \boldsymbol{y}; \boldsymbol{D}) = \begin{cases} p_1 q & \text{si } 0 \leqslant q < D_1, \\ p_1 D_1 & \text{sinon,} \end{cases} \tag{5.3}$$

et pour $i \in [N]$ par

$$r_{i+1}(q; \boldsymbol{y}; \boldsymbol{D}) = \begin{cases} r_i(q; \boldsymbol{y}; \boldsymbol{D}) & \text{si } 0 \leqslant q < y_i, \\ p_{i+1}(q - y_i) + r_i(y_i; \boldsymbol{y}; \boldsymbol{D}) & \text{si } y_i \leqslant q < y_i + D_{i+1}, \\ p_{i+1} D_{i+1} + r_i(q - D_{i+1}; \boldsymbol{y}; \boldsymbol{D}) & \text{si } y_i + D_{i+1} \leqslant q. \end{cases} \tag{5.4}$$

Remarquer que l'on a en particulier

$$R(q; y_1, \dots, y_{N-1}; D_1, \dots, D_N) = r_N(q; \boldsymbol{y}; \boldsymbol{D}) \,.$$

En notant ∂_q^- (resp. ∂_q^+) la dérivation à gauche (resp. droite), on a le résultat général suivant, dû à Brumelle et McGill [6].

Théorème 5.2. *Supposons* $\mathbb{P}(D_1 > 0) = 1$. *Alors le problème admet une solution optimale. Plus précisément, toute suite de niveaux* $y_1^*, \dots, y_{N-1}^*$ *satisfaisant*

$$\partial_q^+ \mathbb{E}[r_i(q; \boldsymbol{y}^*; \boldsymbol{D})]\big|_{q=y_i^*} \leqslant p_{i+1} \leqslant \partial_q^- \mathbb{E}[r_i(q; \boldsymbol{y}^*; \boldsymbol{D})]\big|_{q=y_i^*} \tag{5.5}$$

pour $i \in [N-1]$ *forme une solution optimale du problème (5.2), et il existe toujours une solution optimale de cette forme.*

Dans le cas où les D_i sont de lois continues, la condition (5.5) se réécrit

$$p_{i+1} = \frac{\partial}{\partial q} \mathbb{E}[r_i(q; \boldsymbol{y}; \boldsymbol{D})]\big|_{q=y_i^*} .$$

Ce n'est pas complètement immédiat : il faut montrer que lorsque les D_i sont de lois continues, alors la fonction $q \mapsto \mathbb{E}[r_i(q; \boldsymbol{y}; \boldsymbol{D})]$ est différentiable. On a de plus le résultat suivant, dont la preuve, pas difficile, est omise. Cela fournit une formulation plus explicite des y_i^*, et donc plus accessible aux calculs.

Proposition 5.3. *Lorsque les lois des* D_i *sont continues, l'équation (5.5) est équivalente à*

$$\mathbb{P}(D_1 > y_1^*;\ D_1 + D_2 > y_2^*; \dots; D_1 + D_2 + \cdots + D_i > y_i^*) = \frac{p_{i+1}}{p_1} .$$

Dans le cas où les D_i prennent des valeurs entières, on a le résultat suivant, qui démontre la portée générale du théorème 5.2.

Proposition 5.4. *Si les* D_i *sont à valeurs entières, alors il existe toujours des niveaux* y_i^* *optimaux entiers.*

Enfin, la forme de l'équation (5.5) montre que l'on a nul intérêt à changer les niveaux y_i^* au cours de la vente, en tenant compte de la demande passée. Il a même été démontré que des politiques plus complexes—en autorisant la vente non seulement si le nombre de places restantes est au-dessus d'un seuil, mais aussi si elle prend certaines valeurs particulières par exemple—ne peuvent améliorer le revenu.

5.3.3 Démonstration

La démonstration du théorème 5.2 s'appuie sur les lemmes suivants.

Lemme 5.5. *Si* $\boldsymbol{y}^*$ *satisfait l'équation (5.5) pour tout* $i \in [N]$, *alors l'application* $q \mapsto \mathbb{E}[r_i(q; \boldsymbol{y}^*; \boldsymbol{D})]$ *est concave pour tout* $i \in [N]$.

Démonstration. La preuve se fait par récurrence sur i. □

Lemme 5.6. *Si* $\mathbb{P}(D_1 > 0) = 1$, *alors il existe des niveaux* $y_1^*, \dots, y_{N-1}^*$ *satisfaisant*

$$\partial_q^+ \mathbb{E}[r_i(q; \boldsymbol{y}^*; \boldsymbol{D})]\big|_{q=y_i^*} \leqslant p_{i+1} \leqslant \partial_q^- \mathbb{E}[r_i(q; \boldsymbol{y}^*; \boldsymbol{D})]\big|_{q=y_i^*} .$$

pour tout $i \in [N-1]$.

Démonstration. Une récurrence immédiate sur i montre que $\partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y};\boldsymbol{D})]\big|_{q=0} = p_1$ pour tout i, quelque soit $\boldsymbol{y}$.

Montrons par récurrence sur i que $\lim_{q\to+\infty} \partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y}^*;\boldsymbol{D})] = 0$. C'est clairement vrai pour $i = 1$. Considérons maintenant $i \geqslant 2$. À D_i fixé (i.e. conditionnellement à D_i), on a d'après l'équation (5.4) pour $q > y_{i-1}^*$

$$\partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y}^*;\boldsymbol{D})] = p_i \mathbf{1}(D_i > q - y_{i-1}) + \partial_q^+ \mathbb{E}[r_{i-1}(q - D_i;\boldsymbol{y}^*;\boldsymbol{D})] \times \mathbf{1}(D_i \leqslant q - y_{i-1})\,.$$

Posons $f(q) = \partial_q^+ \mathbb{E}[r_{i-1}(q;\boldsymbol{y}^*;\boldsymbol{D})]$. On a $\lim_{q\to+\infty} f(q) = 0$ (d'après l'hypothèse de récurrence). Soit $\varepsilon > 0$. Il existe q_0 tel que pour tout $q \geqslant q_0$ on a $f(q) < \varepsilon$. On a alors

$$\partial_q^+ \mathbb{E}[r_{i-1}(q-D_i;\boldsymbol{y}^*;\boldsymbol{D})]\times\mathbf{1}(D_i \leqslant q-y_{i-1}) = f(q-D)\mathbf{1}(D \leqslant q-y_{i-1})\mathbf{1}(q-D \geqslant q_0)+f(q-D)\mathbf{1}(D \leqslant q-y_{i-1})\mathbf{1}(q-D < q_0)$$

et donc

$$\partial_q^+ \mathbb{E}[r_{i-1}(q - D_i;\boldsymbol{y}^*;\boldsymbol{D})] \times \mathbf{1}(D_i \leqslant q - y_{i-1}) \leqslant \varepsilon + f(y_{i-1})\mathbf{1}(D_i > q - q_0)\,.$$

On a utilisé la décroissance de la fonction f, due à la concavité de $q \mapsto \mathbb{E}[r_{i-1}(q;\boldsymbol{y}^*;\boldsymbol{D})]$ (lemme 5.5). En passant à l'espérance sur D_i, on obtient :

$$\partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y}^*;\boldsymbol{D})] \leqslant \mathbb{P}(D_i > q - y_{i-1}) + \varepsilon + f(y_{i-1})\mathbb{P}(D_i > q - q_0)\,.$$

La décroissance de $q \mapsto \partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y}^*;\boldsymbol{D})]$ (concavité) montre que l'on peut passer à la limite. Comme D_i est fini presque sûrement, on a $\lim_{q\to+\infty} \partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y}^*;\boldsymbol{D})] \leqslant \varepsilon$. Comme cela est vrai pour tout $\varepsilon > 0$, on a la convergence souhaitée pour tout i.

La concavité de $q \mapsto \mathbb{E}[r_i(q;\boldsymbol{y};\boldsymbol{D})]$ permet alors de conclure : pour tout $y_1, \dots, y_{i-1}$, il existe alors nécessairement une valeur y_i pour laquelle $\partial_q^+ \mathbb{E}[r_i(q;\boldsymbol{y};\boldsymbol{D})]\big|_{q=y_i} \leqslant p_{i+1} \leqslant \partial_q^- \mathbb{E}[r_i(q;\boldsymbol{y};\boldsymbol{D})]\big|_{q=y_i}$ car $p_{i+1} < p_1$. □

Démonstration du théorème 5.2. La preuve se fait par récurrence sur N. Pour $N = 1$, c'est immédiat.

Supposons le théorème démontré pour N. Prouvons le théorème pour $N + 1$. D'après l'hypothèse de récurrence, la quantité $\mathbb{E}[r_N(q; y_1^*, \dots, y_{N-1}^*; \boldsymbol{D})]$ est maximale précisément lorsque les y_i^* vérifiant l'équation (5.5) pour $i = 1, \dots, N-1$. L'équation (5.4) montre donc que $\mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_N^*; \boldsymbol{D})]$ est maximal lorsque les y_i^* vérifiant l'équation (5.5) pour $i = 1, \dots, N-1$. Il reste donc à maximiser par rapport à y_N^*. Pour cela, faisons l'étude de la fonction $y \mapsto \mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})]$. On a, en dérivant l'équation (5.4),

$$\partial_y^+ r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})\big|_{y=y_N^*} = \begin{cases} 0 & \text{si } 0 \leqslant q \leqslant y_N^*, \\ -p_{N+1} + \partial_q^+ r_N(q;\boldsymbol{y}^*;\boldsymbol{D})\big|_{q=y_N^*} & \text{si } y_N^* < q \leqslant y_N^* + D_{N+1}, \\ 0 & \text{si } y_N^* + D_{N+1} < q \end{cases}$$

et

$$\partial_y^- r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})\big|_{y=y_N^*} = \begin{cases} 0 & \text{si } 0 < q < y_N^*, \\ -p_{N+1} + \partial_q^- r_N(q;\boldsymbol{y}^*;\boldsymbol{D})\big|_{q=y_N^*} & \text{si } y_N^* \leqslant q < y_N^* + D_{N+1}, \\ 0 & \text{si } y_N^* + D_{N+1} \leqslant q. \end{cases}$$

Par conséquent, on a

$$\partial_y^+ \mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})]\big|_{y=y_N^*} = \left(-p_{N+1} + \partial_q^+ \mathbb{E}[r_N(q;\boldsymbol{y}^*;\boldsymbol{D})]\big|_{q=y_N^*}\right) \times \mathbb{P}(D_{N+1} \geqslant q - y_N^*)$$

et

$$\partial_y^- \mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})]\big|_{y=y_N^*} = \left(-p_{N+1} + \partial_q^- \mathbb{E}[r_N(q;\boldsymbol{y}^*;\boldsymbol{D})]\big|_{q=y_N^*}\right) \times \mathbb{P}(D_{N+1} > q - y_N^*)\,.$$

L'équation (5.5) pour y_N^* implique que pour toute valeur prise par D_{N+1}, on a

$$\partial_y^+ \mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})]\big|_{y=y_N^*} \leqslant 0 \leqslant \partial_y^- \mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})]\big|_{y=y_N^*}\,.$$

La concavité de $q \mapsto \mathbb{E}[r_N(q;\boldsymbol{y}^*;\boldsymbol{D})]$ (lemme 5.5) implique alors que $y \mapsto \mathbb{E}[r_{N+1}(q; y_1^*, \dots, y_{N-1}^*, y; \boldsymbol{D})]$ est croissante sur $[0, y_N^*[$ et décroissante sur $]y_N^*, +\infty[$. Cela permet de conclure. □

5.4 Les heuristiques EMSR-a et EMSR-b

En pratique, lorsque les lois des D_i sont continues, deux heuristiques directement inspirées du théorème pour 2 classes (théorème 5.1) sont souvent utilisées. Elles sont en effet assez intuitives. Bien que non-optimales (le théorème 5.2 est de toute façon une condition nécessaire et suffisante), elles donnent des solutions de bonne qualité, à quelques pourcents de l'optimum.

5.4.1 EMSR-a

La première heuristique est EMSR-a. Pour fixer le niveau y_i, elle raisonne pour chaque classe $j < i+1$ comme s'il n'y avait que deux classes—la classe j et la classe $i+1$—et applique le théorème 5.1. Cela donne un niveau y_i^j avec y_i^j tel que $\mathbb{P}(D_j > y_i^j) = \frac{p_{i+1}}{p_j}$. Le niveau de protection final pour la classe i est alors $y_i = \sum_{j=1}^{i} y_i^j$.

5.4.2 EMSR-b

La seconde heuristique est EMSR-b. Pour fixer le niveau y_i, elle agrège les classes $j < i+1$ en une seule grande classe et applique le théorème 5.1. L'agrégation se fait comme suit (le prix est la moyenne des prix, pondérés par la demande) :

$$\begin{aligned} \widetilde{D}_i &= \sum_{j=1}^{i} D_j \,. \\ \tilde{p}_i &= \frac{\sum_{j=1}^{i} p_j \mathbb{E}[D_j]}{\sum_{j=1}^{i} \mathbb{E}[D_j]} \,. \end{aligned}$$

Le niveau de protection final pour la classe i est alors tel que $\mathbb{P}(\widetilde{D}_i > y_i) = \frac{p_{i+1}}{\tilde{p}_i}$.

5.5 Compléments bibliographiques

Pour aller plus loin, on pourra consulter les revues de littérature dues à van Ryzin et Talluri [21] ou à Strauss et al. [20].

CHAPITRE 6

Simulation à événements discrets

6.1 Enjeux de la simulation

Une méthode naturelle pour décrire ou prédire le comportement de phénomènes ou de systèmes consiste à les modéliser sous forme d'équations mathématiques avec l'idée d'analyser ces dernières. Cependant, dans de nombreuses situations, les équations ainsi obtenues sont trop complexes pour être résolues (comme en mécanique des fluides) ; il se peut même que l'écriture de telles équations soit difficile (comme les mouvements de foule). Une solution alternative, très ancienne, peut être de former une maquette reproduisant le comportement que l'on souhaite comprendre. Avec l'arrivée des ordinateurs, une telle maquette est devenue informatique et est donc plus fiable et infiniment plus facile à produire et à maintenir. La *simulation* est la reproduction informatique d'un phénomène ou d'un système afin de comprendre et d'analyser son comportement.

Dans de nombreux domaines industriels, la simulation est devenue un outil incontournable de l'ingénieur. Que ce soit par exemple pour la conception d'un avion, l'aménagement d'un quartier, ou le comportement d'un matériau, la simulation permet d'étudier plusieurs solutions sans avoir à réaliser physiquement un prototype qui serait coûteux à réaliser, en temps et en argent.

Si la simulation permet de reproduire le comportement d'un phénomène ou d'un système, et donc de mieux les comprendre, elle permet aussi d'expérimenter de nouvelles solutions, éventuellement en grand nombre, sans avoir à procéder à des expérimentations physiques de toute façon impossibles. Elle forme donc un outil essentiel de l'aide à la décision pour un ingénieur. Elle est particulièrement utile pour l'ingénieur en supply chain, où les systèmes à optimiser possèdent souvent des comportements complexes difficiles à modéliser par des équations : un entrepôt automatisé, un port de marchandises, ou un atelier flexible sont des exemples où la simulation peut particulièrement aider à proposer des organisations et des politiques de gestion efficaces.

Les deux types de simulation les plus classiques sont probablement les deux suivants : la *simulation à temps continu* et la *simulation à événements discrets*. La première traite principalement des phénomènes physiques dans lesquels il est crucial de reproduire à tout instant le comportement du phénomène ou du système (l'écoulement d'un fluide est l'exemple typique). La seconde concerne les systèmes dont le comportement peut être décrit sans perte de précision en se limitant à un sous-ensemble d'instants discret (c'est-à-dire, fini ou

dénombrable). C'est cette dernière qui est essentiellement utilisée en génie industriel et que nous allons étudier dans ce chapitre.

6.2 Simulation à événements discrets

La simulation à événements discrets reproduit le comportement de *systèmes* pouvant se trouver dans différents *états* et pour lesquels les instants de *transition* entre états forment un ensemble discret. Un *événement* est ce qui conduit à la réalisation d'une transition d'un état à un autre. On a toujours un événement "début" qui déclenche la simulation et "fin" qui achève la simulation.

Exemple 6.1. Le comportement d'une file $M/G/1$ peut être étudié par la simulation à événements discrets : le système, c'est la file ; un état est caractérisé par les clients présents dans l'espace d'attente (s'il y en a) et par le client présent dans l'espace de service (s'il y en a un) ; une transition est uniquement réalisée quand un client arrive dans la file et quand un service se termine ; ces deux types de transitions sont donc les deux types d'événements à considérer. (Remarquer qu'un début de service coïncide toujours avec l'arrivée d'un client ou avec une fin de service ; il est donc inutile de considérer un tel événement.)

L'idée centrale de la simulation à événements discrets est de ne simuler que l'enchaînement des événements, en passant directement d'un événement au suivant et sans simuler le système entre deux événements successifs. De plus, à part les événements "début" et "fin", tout événement est toujours généré par un autre événement. Notons en passant que programmer une simulation à événements discrets requiert donc de bien analyser les relations causales au sein du système.

Un programme de simulation doit donc disposer d'un objet "événement". Par ailleurs, il doit comporter des *registres de données* qui permettront l'étude quantitative de la simulation. La description d'un objet événement e comporte les informations suivantes :

- son instant de réalisation t_e.
- le changement d'état qu'il implique.
- les nouveaux événements qu'il déclenche dans le futur.
- la mise à jour des registres de données.

Le déroulement de la simulation s'appuie sur un *échéancier* qui contient en permanence la file des événements prévus. Les événements dans l'échéancier sont ordonnés par instants de réalisation croissants : les événements sont rangés dans l'ordre $e_1, e_2, \ldots$, avec $t_{e_1} \leqslant t_{e_2} \leqslant \cdots$. Si deux événements sont simultanés, ils doivent être tels que les réaliser dans un ordre ou l'autre n'a pas d'incidence. Le cœur d'une simulation consiste en la boucle suivante, où $\mathcal{S}$ représente le système étudié, et E l'échéancier.

repeat
  Dépiler premier événement e de E;
  Modifier l'état de $\mathcal{S}$ d'après e;
  Générer de nouveaux événements futurs d'après e et les insérer aux bons endroits de E;
  Mettre à jour les registres de données;
until $e =$ *"Fin"*;

Algorithme 1 : Boucle principale d'une simulation à événements discrets

Exemple 6.1 (Suite). On revient sur la $M/G/1$. Le système $\mathcal{S}$ est composé de l'espace d'attente et de l'espace de service. Cela peut être aisément réalisé avec des structures de données standard. On suppose être intéressé par la durée d'attente moyenne. Pour pouvoir suivre les temps d'attente des clients, on identifie chaque client avec un numéro. On a un registre de données stockant pour chaque client son instant d'arrivée et un autre stockant son instant de début de service. Considérons un événement e dépilé.

- Supposons $e =$"arrivée du client i".
  - Si l'espace de service est vide :
    - On place i dans l'espace de service.
    - On crée un événement e' "arrivée du client $i+1$" avec comme instant de réalisation $t_{e'} = t_e + T'$, où T' est tirée selon une loi exponentielle. On place e' au bon endroit dans E.
    - On crée un événement e'' "fin de service du client i" avec comme instant de réalisation $t_{e''} = t_e + T''$, où T'' est tirée selon la loi de service. On place e'' au bon endroit dans E.
    - On met à jour le registre de données : on indique l'arrivée et le début de service du client i à l'instant t_e.
  - Si l'espace de service est occupé :
    - On place i dans l'espace d'attente.
    - On crée un événement e' "arrivée du client $i+1$" avec comme instant de réalisation $t_{e'} = t_e + T'$, où T' est tirée selon une loi exponentielle. On place e' au bon endroit dans E.
    - On met à jour le registre de données : on indique l'arrivée du client i à l'instant t_e.
- Supposons $e =$"fin de service du client i".
  - Si l'espace d'attente est vide :
    - On retire i de l'espace de service.
    - On met à jour le registre de données : on indique la fin de service du client i à l'instant t_e.
  - Si l'espace d'attente est occupé :
    - On retire i de l'espace de service de la file.
    - On met à jour le registre de données : on indique la fin de service du client i à l'instant t_e.

— On crée un événement e' "fin de service de client" pour le client j le plus ancien de la file (politique FIFO), avec comme instant de réalisation $t'_e = t_e + T'$, où T' est tirée selon la loi de service. On place e' au bon endroit dans E.

— On met à jour le registre de données : on indique le début de service du client j à l'instant t_e.

À la fin de la simulation, il est aisé de parcourir les registres de données pour calculer la moyenne des temps d'attente.

Remarque 1. En pratique, il n'est pas forcément nécessaire de réimplémenter l'échéancier, les événements, etc. Il existe de nombreuses librairies de simulation à événements discrets pour différents langages (comme par exemple SimPy pour Python ou OMNeT++ pour C++) et de nombreux logiciels proposant des fonctionnalités complémentaires variées (comme par exemple Arena ou Witness).

Cela dit, pour comparer des algorithmes de temps réel ou des politiques de gestion, sans prétendre à un réalisme parfait (inutile d'ailleurs pour ce genre de questions), programmer une simulation à événements discrets peut être très rapide et d'utilisation très souple.

6.3 Modéliser l'aléa

L'aléa joue souvent un rôle important dans l'évolution des processus industriels. Les temps de traitement, les arrivées de clients, les niveaux de commande ou les temps de parcours sont autant de quantités pour lesquelles les incertitudes peuvent être substantielles. Les probabilités forment l'outil mathématique indispensable pour appréhender, modéliser et donc simuler ces phénomènes.

Dans cette section, nous allons d'abord rappeler quelques distributions de probabilité utiles dans la simulation des processus industriels. Ensuite, dans un second temps, nous allons donner deux méthodes pour réaliser des tirages selon des distributions arbitraires.

6.3.1 Quelques distributions de probabilité

Les distributions classiques

La plupart des distributions classiques peuvent être utiles dans un contexte industriel. Nous avons vu l'importance de la loi exponentielle et de la loi de Poisson pour modéliser les processus d'arrivées indépendants et non-coordonnés. Les niveaux de demande ou les délais d'approvisionnement sont souvent modélisés de manière pertinente par une loi normale. (Ce sont des situations où les réalisations vont se répartir de manière symétrique par rapport à une moyenne, tant que les valeurs ne sont pas trop grandes ou trop proches de zéro.) La plupart des distributions classiques vues en cours de probabilité peuvent être utiles pour modéliser les phénomènes aléatoires en supply chain. Lorsqu'on veut simplement comparer des heuristiques ou expérimenter des nouvelles politiques, il suffit de donner aux paramètres de ces lois des valeurs raisonnables pour obtenir des simulations pertinentes. Si l'on veut

reproduire de manière fidèle un processus, on peut utiliser des historiques de données pour estimer les valeurs de ces paramètres.

Les distributions empiriques

Si les historiques de données sont suffisamment fournis, on peut aussi directement les utiliser comme distribution de probabilité.

Supposons par exemple que l'on veuille simuler les arrivées de clients dans un restaurant de manière assez fidèle. Un processus de Poisson homogène ne tiendra ni compte des variations d'intensité de la demande selon l'heure, ni compte de l'existence d'arrivées de groupes de clients. Si l'on dispose d'un historique de plusieurs mois, on peut simuler des réalisations en tirant un jour uniformément au hasard dans l'historique et "rejouer" l'histoire des arrivées de clients de ce jour-là.

Pour les commandes d'une activité de commerce en ligne, on peut s'appuyer sur la liste des commandes effectuées dans les derniers mois : une commande aléatoire dans la simulation est obtenue en en tirant une au hasard uniformément dans cette liste.

Les distributions "lack of knowledge"

Il est cependant assez fréquent, en particulier dans un contexte industriel, de ne pas savoir quelles distributions sont pertinentes pour modéliser une quantité aléatoire. Cela peut provenir du fait qu'on ne dispose d'aucun historique ou car le système étudié n'a pas encore connu de réalisation concrète.

Si l'on ne connaît que les valeurs minimale a et maximale b que peut prendre une quantité aléatoire, la distribution uniforme sur $[a, b]$ peut être utilisée ; voir figure 6.1. Si X est une variable aléatoire distribuée selon une telle distribution uniforme, on a

$$\mathbb{E}[X] = \frac{1}{2}(a+b) \quad \text{et} \quad \mathbb{V}[X] = \frac{1}{12}(b-a)^2 .$$

Si l'on connaît la valeur minimale a, la valeur maximale b et la valeur "typique" c que peut prendre cette quantité aléatoire, la *distribution triangulaire* de paramètres a, b, c peut être utilisée ; voir figure 6.2. Sa densité s'écrit :

$$p(t) = \begin{cases} \dfrac{2(t-a)}{(b-a)(c-a)} & \text{si } a \leqslant t \leqslant c, \\ \dfrac{2(b-t)}{(b-a)(b-c)} & \text{si } c \leqslant t \leqslant b, \\ 0 & \text{sinon.} \end{cases}$$

La valeur c est le *mode* de la distribution : c'est le point où cette dernière atteint son maximum. Si X est une variable aléatoire distribuée selon une telle distribution triangulaire, on a

$$\mathbb{E}[X] = \frac{1}{3}(a+b+c) \quad \text{et} \quad \mathbb{V}[X] = \frac{1}{18}(a^2+b^2+c^2-ab-ac-bc) .$$

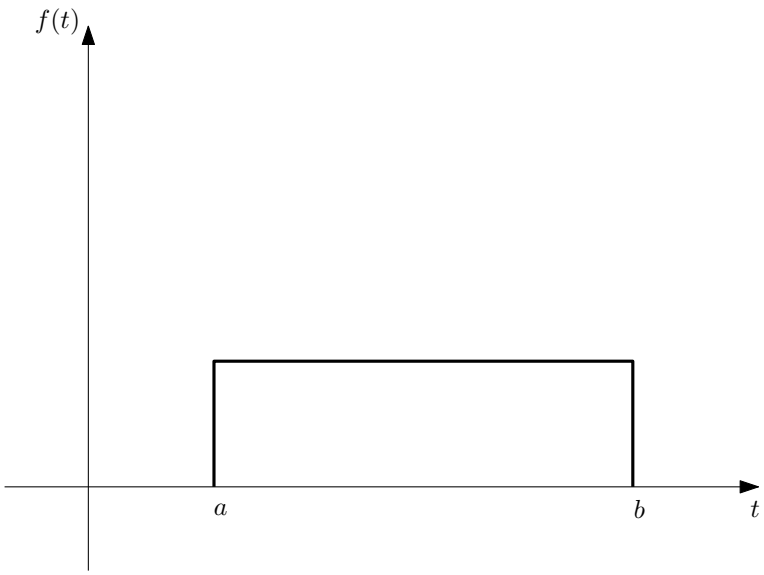

FIGURE 6.1 – La densité d'une distribution uniforme

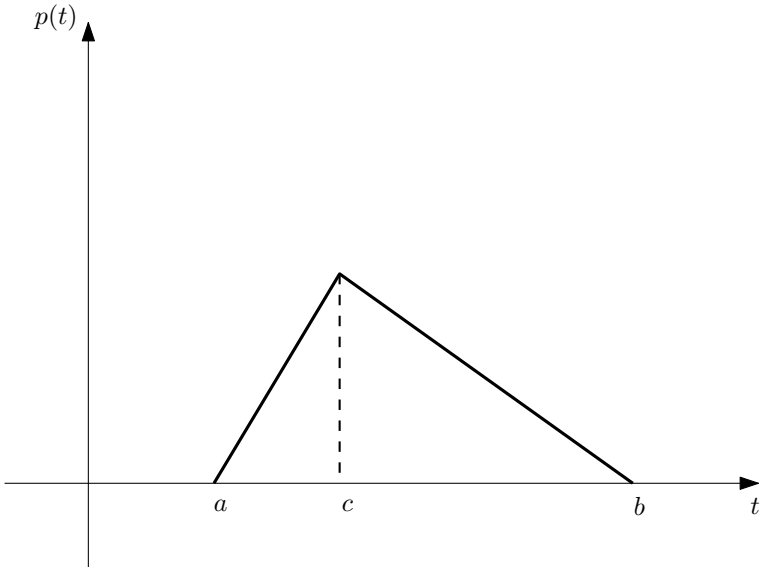

FIGURE 6.2 – La densité d'une distribution triangulaire

Il est à noter que, tant pour la distribution uniforme que pour la distribution triangulaire, l'espérance s'exprime comme la moyenne des paramètres.

Ce qui est intéressant avec les distributions uniforme et triangulaire, c'est l'interprétation aisée de leurs paramètres : un manager sans bagage mathématique particulier mais connaissant bien le système étudié pourra par exemple donner sans difficulté les paramètres de ces distributions.

6.3.2 Tirer selon des lois arbitraires

La plupart des langages de programmation donnent accès à des générateurs de nombres aléatoires. Même si des librairies offrent la possibilité de tirer de tels nombres selon des distributions variées, il arrive parfois que la simulation requiert les tirages selon une distribution qui n'est pas fournie par la librairie. Il existe plusieurs techniques pour pouvoir le faire à partir de tirages aléatoires selon d'autres distributions. On dit alors que l'on *simule* la variable aléatoire.

La méthode de l'inverse

La *méthode de l'inverse* s'appuie sur le résultat suivant, dont la preuve est omise car élémentaire et classique.

Proposition 6.1. *Soit X une variable aléatoire réelle, de fonction de répartition F. Soit U une variable aléatoire uniforme sur $[0,1]$. Alors les variables aléatoires $\inf\{t\colon F(t) \geqslant U\}$ et X ont même loi.*

La fonction $y \mapsto \inf\{t\colon F(t) \geqslant y\}$ est l'*inverse généralisée* de F. Dans le cas particulier où F est inversible (ce qui est le cas quand la loi de X est continue et de densité strictement positive partout sur son support), cette fonction est l'inverse habituelle et cette proposition dit que $F^{-1}(U)$ et X ont la même loi.

Un autre cas particulier intéressant est celui d'une variable aléatoire discrète finie X, de distribution $\mathbb{P}(X = k) = p_k$ pour $k = 1\ldots, n$. Ce que dit la proposition 6.1, c'est que pour simuler X, il suffit de tirer une uniforme U sur $[0,1]$ et de renvoyer l'entier k tel que $U \in [p_0 + p_1 + \cdots + p_{k-1}, p_0 + p_1 + \cdots + p_{k-1} + p_k[$. Noter que ce dernier fait se démontre d'ailleurs directement.

Cependant, même dans le cas où F est inversible, il n'est pas toujours facile de calculer l'inverse de F. Une méthode alternative est disponible. C'est ce que nous allons voir maintenant.

La méthode de rejet

Soit X une variable aléatoire réelle de distribution continue, de densité f. Supposons que l'on connaisse une fonction $g \geqslant f$ et que l'on sache simuler une variable aléatoire de densité $h\colon t \mapsto \frac{1}{\int_{\mathbb{R}} g(s)\,\mathrm{d}s} g(t)$. La *méthode de rejet* consiste alors à effectuer les étapes suivantes pour simuler un tirage de X.

1. Tirer Y selon la distribution h (en normalisant).
2. Tirer U uniformément sur $[0, 1]$, indépendamment de Y.
3. Si $U \leqslant \frac{f(Y)}{g(Y)}$, renvoyer $Z = Y$. Sinon, reprendre en 1.

On a le résultat suivant, dont la preuve est omise car élémentaire et classique.

Proposition 6.2. *Les variables aléatoires X et Z suivent la même loi.*

On peut montrer que la convergence est d'autant meilleure que $\sup_t \frac{g(t)}{f(t)}$ est petit.

6.4 Expérimenter et analyser les résultats

Le but d'une simulation est presque toujours d'évaluer les valeurs prises par certaines quantités (c'est à cela que servent les registres mentionnés dans la section 6.2). En même temps, comme la simulation est souvent utilisée pour étudier des systèmes où l'aléa joue un rôle important, cette évaluation nécessite un traitement statistique et le calcul d'intervalles de confiance. Ces derniers requièrent d'avoir accès à une variance empirique. Une façon de faire cela de manière propre (mais ce n'est pas la seule façon) est de rejouer plusieurs fois la simulation, ce qu'on appelle des *réplications*, en prenant bien garde d'initialiser le générateur de nombres aléatoires sur des graines différentes. Le nombre de réplications est en général assez petit—de l'ordre de quelques dizaines—car le gain en précision devient vite marginal, tout en augmentant le temps de calcul.

L'intervalle de confiance au niveau $1 - \alpha$ d'une quantité x après r réplications est donné par

$$\left[\bar{x} - t^{r-1}_{\alpha/2}\frac{s}{\sqrt{r}}, \bar{x} + t^{r-1}_{\alpha/2}\frac{s}{\sqrt{r}}\right],$$

où

- $\bar{x} = \frac{1}{r}\sum_i x_i$ (moyenne empirique)
- $s^2 = \frac{1}{r-1}\sum_i (x_i - \bar{x})^2$ (variance empirique sans biais)
- t^k_γ est le quantile d'ordre γ à k degrés de liberté de la loi de Student car r est en général assez petit.

Remarque 2. Une méthode alternative au changement de graine pour le tirage de nombres aléatoires dans les différentes réplications consiste à faire tous ces tirages en avance, et à la conserver dans des fichiers utilisés pour les différentes réplications. Par exemple, pour des arrivées aléatoires sur une journée, on peut très bien préparer en avance une petite centaine de fichiers, chacun contenant des arrivées tirées aléatoirement selon la loi simulée. Chaque réplication utilise alors un fichier différent. Un avantage de cette méthode est de diminuer le bruit dans les comparaisons entre différents algorithmes.

6.5 Compléments bibliographiques

Plusieurs ouvrages très complets ont été écrits sur la simulation à événements discrets. Citons par exemple celui de Banks et al. [3] ou celui de Cassandras et Lafortune [7]. L'ouvrage de Stewart [19], cité à la fin du chapitre sur les files d'attente, comporte une partie assez fouillée sur la simulation.

Bibliographie

[1] S. Asmussen, *Applied probability and queues*, vol. 51, Springer Science & Business Media, 2008. 13

[2] François Baccelli, Bartłomiej Błaszczyszyn, and Mohamed Kadhem Karray, *Random measures, point processes, and stochastic geometry*, 2024. 8

[3] J. Banks, *Discrete event system simulation*, Pearson Education India, 2005. 51

[4] R. Bellman, I. Glicksberg, and O. Gross, *On the optimal inventory equation*, Management Science **2** (1955), no. 1, 83–104. 34

[5] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*, Cambridge university press, 2005. 26

[6] S. L. Brumelle and J. I. McGill, *Airline seat allocation with multiple nested fare classes*, Operations research **41** (1993), no. 1, 127–137. 39

[7] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, Springer, 2008. 51

[8] A. Fiat and G. J. Woeginger, *Online algorithms : The state of the art*, vol. 1442, Springer, 1998. 21, 26

[9] R. L. Graham, *Bounds for certain multiprocessor anomalies*, Bell Systems Tech **45** (1966), 1563–1581. 20

[10] P. Jaillet and M. R. Wagner, *Online optimization*, Springer Publishing Company, Incorporated, 2012. 26

[11] A. Y. Khintchine, *Mathematical theory of stationary queue*, Matematicheskii Sbornik **39** (1932), 73–84. 14

[12] J. F. C. Kingman, *The single server queue in heavy traffic*, Mathematical Proceedings of the Cambridge Philosophical Society (1960), 902–904. 15

[13] J. Little, *A proof of the queueing formula : $L = \lambda W$*, Operations Research (1961), 383–387. 3

[14] ———, *Little's Law as Viewed on Its 50th Anniversary*, Operations Research (2011), 536–549. 6

[15] K. Littlewood, *Forecasting and control of passenger bookings*, Airline Group International Federation of Operational Research Societies Proceedings, 1972 **12** (1972), 95–117. 38

[16] F. Pollaczek, *Über eine aufgabe der Wahrscheinlichkeitstheorie. i*, Mathematische Zeitschrift **32** (1930), no. 1, 64–100. 14

[17] C. Rajendran and O. Holthaus, *A comparative study of dispatching rules in dynamic flowshops and jobshops*, European journal of operational research **116** (1999), no. 1, 156–170. 24

[18] H. Scarf, *The optimality of* (s, S) *policies in the dynamic inventories problem*, Mathematical Methods in the Social Sciences, 1959 (K. Arrow, S. Karlin, and P. Suppes, eds.), 1960, pp. 196–202. 34

[19] W. J. Stewart, *Probability, Markov chains, queues, and simulation*, Princeton university press, 2009. 16, 51

[20] A. K. Strauss, R. Klein, and C. Steinhardt, *A review of choice-based revenue management : Theory and methods*, European journal of operational research **271** (2018), no. 2, 375–387. 42

[21] G. J. van Ryzin and K. T. Talluri, *An introduction to revenue management*, Emerging Theory, Methods, and Applications, Informs, 2005, pp. 142–194. 42

[22] P. H. Zipkin, *Foundations of inventory management*, McGraw-Hill Higher Education, 2000. 35