N° d'ordre : 337 N° attribué par la bibliothèque : 05ENSL0 337

École Normale Supérieure de Lyon
Laboratoire de l'Informatique du Parallélisme

THÈSE

présentée et soutenue publiquement le 2 décembre 2005 par

Anne Bouillard

pour l'obtention du grade de

Docteur de l'École Normale Supérieure de Lyon
spécialité : Informatique

au titre de l'École doctorale de mathématiques et d'informatique fondamentale de Lyon

Optimisation et analyse probabiliste de systèmes à événements discrets

Directeur de thèse : Bruno Gaujal

Après avis de : Monsieur Javier Esparza
Monsieur Patrice Moreaux

Devant la commission d'examen formée de :

Monsieur Javier Esparza	Membre/Rapporteur
Monsieur Bruno Gaujal	Membre
Monsieur Jean Mairesse	Membre/Co-encadrant
Monsieur Patrice Moreaux	Membre/Rapporteur
Madame Anca Muscholl	Membre
Monsieur Yves Robert	Membre/Président

Remerciements

Je voudrais tout d'abord à remercier mes directeurs de thèse, sans qui je n'aurais sans doute pas fait cette thèse. J'ai beaucoup appris à leur contact et j'ai vraiment pris du plaisir à faire cette thèse avec eux. J'ai vraiment apprécié leur rigueur et l'intérêt qu'ils ont manifesté pour mon travail. Je tiens à remercier Bruno Gaujal pour m'avoir accordé sa confiance dès mon stage de maîtrise et Jean Mairesse de m'avoir encadrée depuis le DEA. Malgré l'éloignement, il a toujours su me redonner confiance dans les moments difficiles.

Je voudrais aussi remercier Javier Esparza et Patrice Moreaux d'avoir bien voulu être les rapporteurs de ma thèse, Yves Robert et Anca Muscholl d'avoir accepté de participer à ce jury.

Un grand merci à Éric Thierry qui m'a soutenue tout au long de cette thèse par son micro-coaching intensif et en me faisant partager son expérience de jeune docteur. J'ai aussi beaucoup apprécié de travailler avec lui.

Un grand merci à Emmanuel Hyon, le virgulomane fou, pour avoir accepté de relire ma thèse, à Fabrice Rastello pour sa relecture très attentive d'une partie de ce manuscript et les travaux communs que nous avons effectués.

Je voudrais aussi remercier les équipes et laboratoires qui m'ont accueillie ces dernières années : le LIAFA depuis mon stage de DEA et le LIP. En particulier j'aimerais remercier l'équipe MC2 qui m'a accueillie après la disparition de l'équipe TRIO.

La vie du laboratoire aurait été bien triste sans les membres qui le composent. En particulier merci à Daniel, Tom (le Perche[1], ça tue des ours), Stéphane, Victor, Emmanuel, Guillaume, Sylvain, Jingdi, Natacha, Pascal, merci à mes collègues de bureaux successifs : Jean-Chrisophe, Antoine, Hadda, Nicolas, Tri, sans oublier les parisiens Berke, Hugo, Blaise et Olivier pour les discussions autour d'un café, les soirées et les sorties.

Je voudrais aussi remercier tous ceux qui se sont intéressé à l'évolution des mes travaux et on cru à leur aboutissement. En premier lieu, merci à mes parents qui m'ont toujours encouragée dans mes études, merci aux copains du B6, Alexis, Benoît, Lionel, Suzanne et tous les autres, merci à Catherine, à Gérald, pour avoir bien voulu jouer le rôle du relecteur non spécialiste, à Olivier et tous ceux que j'aurais malencontreusement oubliés.

[1]Vincent Malone. *Le Perche*. Les éditions Tarago

Table des matières

Introduction

Les systèmes à événements discrets sont des systèmes dynamiques dans lesquels les changements d'états se font à des instants discrets, et non continuement, comme dans la théorie classique des systèmes dynamiques. L'émergence de systèmes conçus par l'homme dans les dernières décennies, tels que les systèmes informatiques, les systèmes de production, les réseaux de communication et de transport, a mené au développement d'une théorie pour ces systèmes, les systèmes (dynamiques) à événements discrets.

Ces systèmes se définissent par opposition aux systèmes dynamiques classiques [26, 71, 85], dont l'évolution est en général régie par des systèmes d'équations différentielles, éventuellement à dérivées partielles, pour des systèmes en temps continu ou des suites récurrentes pour des systèmes en temps discret. L'évolution d'un système dynamique est représentée par un système d'équations (différentielles, de récurrence...) et par la donnée de l'état initial, ce qui définit le comportement du système au cours du temps. Par exemple le mouvement des planètes, d'un pendule ou les circuits électriques sont des systèmes dynamiques.

Par contre, pour les systèmes à événement discrets, ce sont les événements qui ont lieu dans le système qui modifient l'état du système. L'état du système n'évolue alors qu'en un nombre dénombrable d'instants, mais, contrairement aux systèmes dynamiques discrets, ces instants ne sont pas réguliers et peuvent avoir lieu dans un espace continu, les réels positifs par exemple. Les systèmes à événements discrets se distinguent donc des systèmes dynamiques. C'est pourquoi des outils spécifiques ont été développés pour leur étude. Ce sont ces outils qui ont été étudiés au cours de cette thèse.

Donnons maintenant quelques exemples de systèmes à événements discrets.

Exemples de systèmes à événements discrets

Files d'attente. L'un des exemples les plus simples est celui de la file d'attente : des clients entrent dans un système (un guichet, un feu de circulation, un réseau dans lequel circulent des paquets...) et sont servis dans leur ordre d'arrivée. L'espace d'état est le nombre de clients dans le système. Les événements sont de deux sortes : l'arrivée d'un client et la sortie d'un client. L'état du système varie uniquement lorsqu'un de ces deux événements a lieu : il augmente de un quand un client arrive et diminue de un quand un client sort. La figure 1 représente le schéma d'une file d'attente.

Le modèle présenté ici est modèle le plus simple de file d'attente, mais il existe des variantes :

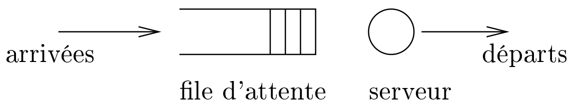


Fig. 1 – File d'attente.

l'ordre dans lequel sont servis les clients peut varier, la capacité de la file d'attente peut être finie (dans ce cas, les clients arrivant dans une file d'attente pleine sont rejetés), il peut y avoir plusieurs guichets pour une seule file d'attente.

On peut aussi relier plusieurs files d'attente : lorsqu'un client sort d'une file d'attente, il entre dans une autre. On parle alors de réseau de files d'attente. Ces réseaux apparaissent naturellement dans les systèmes informatiques et les réseaux de communications. Ils ont été étudiés entre autre dans [13, 9] par exemple.

Ateliers de production. Un produit manufacturé est constitué de plusieurs pièces de base qu'il faut transformer et assembler. Lorsque les pièces de base sont en nombre suffisant, on peut les assembler et une machine est chargée de cet assemblage. En général, elle peut assembler un seul ensemble de pièces données à la fois. L'espace des états de ce système est le nombre de pièces de chaque type (pièces de base, partiellement transformées, finies) et les événements qui ont lieu sont les transformations des pièces.

Les ateliers flexibles permettent de produire plusieurs types de pièces, et de partager les ressources et les machines pour les différents produits.

Ces exemples font apparaître certaines propriétés des systèmes à événements discrets :
- la *concurrence*, lorsque plusieurs actions peuvent avoir lieu en même temps. Dans un réseau de communication, plusieurs clients peuvent être traités en même temps s'ils sont servis par des serveurs différents ;
- la *compétition*, dans un atelier flexible, si certaines ressources sont nécessaires à la fabrication de deux pièces différentes, il faut choisir quelle pièce va effectivement être produite avec ces ressources. Pour chaque unité de ressource, ce choix est exclusif ;
- la *synchronisation*, quand plusieurs ressources sont nécessaires à la fabrication d'une pièce, pour usiner cette pièce, il faut attendre que toutes les ressources soient disponibles.

Les systèmes à événements discrets peuvent aussi être *contrôlés* pour que le système ait le comportement désiré. En effet, les systèmes étant gouvernés par l'occurrence d'événements, on peut vouloir agir sur ces événements en fonction de l'état dans lequel on se trouve. Les actions résultant d'un contrôle donnent lieu à des événements contrôlés. Il peut donc y avoir deux classes d'événements dans un système à événements discrets, les événements contrôlés, et les événements non contrôlés. Par exemple, dans un atelier flexible, ce contrôle peut s'effectuer pour choisir quelle pièce usiner en fonction des ressources disponibles et des pièces déjà produites. Dans un système de files d'attente, si un client sortant d'une file peut choisir entre plusieurs files d'attente, laquelle va-t-on le router ?

Le problème du choix des actions à effectuer dans un système à événements discrets pose aussi la question de savoir ce que l'on veut faire avec. Deux approches complémentaires existent :
- l'approche qualitative étudie les suites d'événements qui peuvent avoir lieu. On voudra par exemple éviter les blocages du système ou on voudra restreindre l'ensemble des suites d'événements possibles parmi toutes celles qui peuvent avoir lieu. Cette approche ne s'intéresse pas aux dates auxquelles ont lieu les événements, mais à la succession logique des événements ;
- l'approche quantitative s'intéresse aux dates auxquelles ont lieu les événements. Principalement, on voudra évaluer le nombre d'événements d'un certain type ayant lieu dans un certain intervalle de temps ou en moyenne par unité de temps (ce qu'on appelle aussi *débit*). Par exemple, calculer le nombre de clients servis par unité de temps dans une file d'attente

pour le comparer au nombre moyen de clients arrivant par unité de temps est utile afin de savoir si un système est stable (le nombre de clients dans la file n'explose pas). Dans un atelier flexible, on voudra aussi pouvoir calculer et ajuster le nombre de pièces d'un certain type fabriquées, pour pouvoir répondre à la demande sans être en surproduction...

Modélisation des systèmes à événements discrets

Les systèmes à événements discrets sont des systèmes concrets, mais pour les étudier et les analyser, des modèles ont été développés.

L'outil le plus simple pour modéliser les suites d'événements sont les automates finis. En effet, partant d'un état, si un événement a lieu, on rejoint un nouvel état. La théorie des automates a été amplement étudiée et, pour tenir compte du temps, on a introduit les automates temporisés. Malheureusement, les automates ne sont pas bien adaptés à la modélisation de la synchronisation et de la concurrence des événements. Les *traces* ont été introduites pour modéliser la concurrence ([37, 36, 64]) : les traces sont des mots sur un alphabet fini dont certaines lettres commutent. Si deux lettres commutent dans un mot, c'est qu'elles représentent des actions concurrentes. Toutefois, ceci ne permet pas de représenter la synchronisation.

Les réseaux de Petri. Reprenons plus précisément l'exemple d'un atelier de production. On veut fabriquer des pièces c en transformant des pièces a et des pièces b, à l'aide d'une machine. Pour ce faire, on doit stocker les pièces a et les pièces b en un lieu accessible par la machine tandis que les pièces c doivent être placées dans un endroit après fabrication. Symboliquement, plaçons les pièces a, b et c dans trois cercles distincts. Schématiquement, la machine peut avoir deux états. Soit elle est en train de fabriquer une pièce c, elle est « occupée », soit elle ne fait rien, elle est « inactive ». On peut aussi représenter ces deux états par des cercles, et un marqueur indique dans quel état est la machine. Deux types d'événements sont possibles : la machine commence une transformation, et la machine termine une pièce c. Représentons ces deux actions par des rectangles. Il faut maintenant relier les événements aux ressources et états de la machine : le début d'une transformation pour la machine ne peut avoir lieu que si la machine est inactive et si les ressources a et b sont disponibles. Relions alors les cercles représentant les ressources et celui représentant la machine inactive au rectangle représentant le début d'une transformation. Pendant la transformation, la machine change d'état : elle devient occupée. On relie donc le rectangle « début de la transformation » au cercle « machine occupée ». Lorsque la machine termine son travail, elle change à nouveau d'état, elle devient inactive. On relie donc le cercle « machine occupée » au rectangle « fin de transformation » et ce rectangle au cercle « machine inactive » et à un cercle représentant le nombre de pièces c fabriquées. On obtient le schéma de la figure 2.

Ce type de schéma est un *réseau de Petri*. Sur cet exemple, on peut voir que l'on peut représenter la synchronisation. S'il y avait deux tels systèmes indépendants l'un de l'autre, on pourrait représenter la concurrence d'événements : les deux machines fonctionnent indépendamment l'une de l'autre. De même, si la pièce a était utilisée par plusieurs machines, on aurait une compétition entre les deux machines pour savoir laquelle utiliserait chaque pièce a.

Modélisé de cette manière, le système peut être étudié de manière logique. Il est aussi possible d'ajouter une dimension temporelle à ce système : les actions, représentées par les rectangles, peuvent prendre un certain temps et la machine rester dans l'état « occupé » un certain temps également, c'est le temps de fabrication de la pièce c.

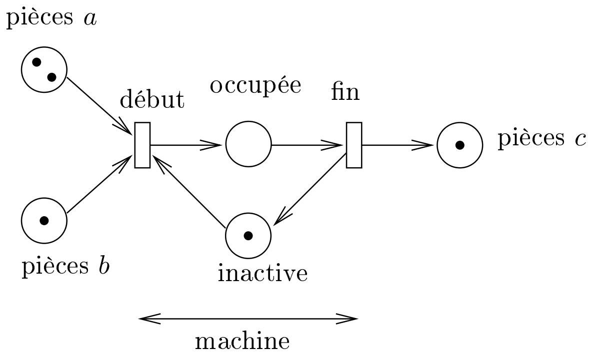


FIG. 2 – Modélisation d'un atelier de production.

Les réseaux de Petri sont beaucoup utilisés pour la modélisation des systèmes de production et des ateliers flexibles. Ils font le lien entre les ressources ou états des composants du système (représentés par des cercles) et les événements ou actions qui ont lieu dans le système (représentés par des rectangles).

On vient de voir que les réseaux de Petri peuvent modéliser des ateliers de production. Ils sont en effet beaucoup utilisés dans ce domaine, mais aussi dans beaucoup d'autres : la modélisation des réseaux, le diagnostic et la détection d'erreurs ([53]), la bio informatique.

L'algèbre (max,plus) Supposons maintenant que les pièces a et b sont aussi des pièces produites. Elles arrivent dans les emplacements accessibles par la machine aux dates $(a_n)_{n\in\mathbb{N}^*}$ pour les pièces a et $(b_n)_{n\in\mathbb{N}^*}$ pour les pièces b. Le temps de production d'une pièce c est τ. On veut savoir à quelles dates les pièces c sont produites. On note c_n la date de production de la n-ième pièce c. À l'instant initial, aucune pièce a ou b n'est disponible et la machine est oisive. Le début de production de la première pièce c est alors la date à laquelle il y a au moins une pièce a disponible et une pièce b disponible. C'est donc $\max(a_1, b_1)$. La date de fin de production de la première pièce c est donc $\max(a_1, b_1) + \tau$. La n-ième pièce c commence à être produite quand la $(n-1)$-ième pièce c est terminée, et que n pièces de a et de b ont été produites. Cette date est donc $\max(a_n, b_n, c_{n-1})$ et on a $c_n = \max(a_n, b_n, c_{n-1}) + \tau$. Sur cet exemple, les deux opérations utilisées sont le maximum, quand il y a synchronisation, et l'addition pour représenter les durées des actions.

Dans cet exemple, les opérations apparaissant dans les calculs sont le maximum et l'addition, alors que les opérations classiques de l'algèbre sont l'addition et la multiplication. Ceci a amené les théoriciens des systèmes à événements discrets à étudier des structures algébriques non classiques, les algèbres (max,plus) ([57, 11, 48]).

Simulation des systèmes à événements discrets Parfois, la modélisation des systèmes à événements discrets n'en permet pas une analyse précise. On utilise alors des techniques de simulations : on peut faire évoluer le système sur une longue période de temps et regarder le comportement asymptotique.

De nombreux logiciels ont été développés pour la simulation des événements discrets, parmi lesquels OMNET++, C++SIM. Les réseaux de Petri eux aussi ont fait l'objet de plusieurs logiciels d'analyse et de simulation : GREATSPN, PROTOS, TINA.

Objectifs de la thèse

Cette thèse est consacrée à l'étude du comportement temporel des systèmes à événements discrets, à travers l'étude des réseaux de Petri et des réseaux de files d'attente.

Une première problématique est l'évaluation des performances des réseaux : on se pose en particulier la question de l'évaluation de son débit, du temps passé par un client dans un système, de la longueur maximale des files d'attente. Les systèmes étudiés peuvent être déterministes : les temps de service ou d'exécution sont constants et les conflits entre plusieurs actions sont résolus à l'avance. Les systèmes peuvent aussi être stochastiques : les temps de service sont aléatoires et les résolutions des conflits pas nécessairement déterministes. Suivant les cas, on cherchera à calculer les performances du système de manière exacte, en moyenne ou à obtenir des garanties sur les performances. Des études ont été menées sur les mesures des performances dans des réseaux de files d'attente. Pour certains types de réseaux, les réseaux de Jackson, aussi dits à forme produit, on peut déduire du comportement stationnaire de chacune des files le comportement stationnaire global du réseau ([55, 63]). Pour des réseaux plus généraux, le comportement est beaucoup plus difficile à appréhender. Par exemple, dans [7, 15], il est montré que si chaque file d'attente d'un réseau prise séparément est stable (le nombre moyen de paquets en attente est fini), le réseau global n'est pas nécessairement stable. Aucune caractérisation simple n'existe pour prouver la stabilité des réseaux. Les performances de certaines classes de réseaux de Petri ont été étudiées. En utilisant des techniques algébriques ([11]) ou probabilistes ([3, 52, 60]), l'existence d'un comportement stable sous certaines conditions a été montrée. Les classes de réseaux privilégiées lors de ces études sont les graphes d'événements et les réseaux à choix libres. Ces résultats vont nous servir de base pour l'évaluation des performances de systèmes.

Une deuxième problématique est l'amélioration des performances. Lorsqu'il y a un choix entre plusieurs actions, comment choisit-on l'action qui maximise les performances du système ? C'est là qu'intervient le contrôle dans les systèmes à événements discrets : un contrôleur peut choisir quelles actions doivent avoir lieu en vue d'améliorer les performances du système. Cette problématique a été abordée dans divers domaines de l'informatique. Une première approche est l'ordonnancement de tâches : quand plusieurs actions doivent être effectuées, on tente de trouver un ordre qui optimise l'utilisation des ressources. Ces problèmes ont été étudiés par exemple dans [56, 53]. Une autre approche est celle de l'admission dans les réseaux : pour garantir de bonnes performances on rejette certains paquets dans les réseaux de sorte que ceux qui sont acceptés sont traités dans de bonnes conditions. Ces problèmes ont été étudiés dans [62, 6] dans le cas des réseaux ouverts sans cycle dans le cadre des réseaux files d'attente. Le problème d'admission se généralise naturellement en un problème de routage : lorsqu'un paquet a été traité ou arrive dans un réseau, le routage consiste à diriger ce paquet selon un chemin. L'optimisation du routage vise alors à minimiser le temps passé dans le système ou la longueur des files d'attente du réseau. Le problème de routage peut aussi être étudié dans des cadres plus larges que celui des files d'attente, comme les ateliers de production. Ainsi, le problème du routage dans les réseaux de Petri permet de considérer le routage pour de grandes classes de systèmes à événements discrets. Mais ces problèmes ont principalement été étudiés pour des réseaux en boucle ouverte. Ici, nous nous proposons d'aborder le problème pour des réseaux de Petri fortement connexes, donc des réseaux fermés.

Plan de la thèse

Nous présentons maintenant la structure du mémoire de thèse avec une sélection de quelques théorèmes pour donner un aperçu des résultats obtenus.

Le premier chapitre de la thèse présente les principaux outils mathématiques utilisés dans cette thèse, à savoir l'algèbre (max,plus) et les chaînes de Markov.

La première partie s'intéresse aux traces. Le monoïde de traces a beaucoup été étudié pour modéliser la concurrence des événements ([37]). Nous étendons la structure de monoïde en groupe en ajoutant des lettres inverses. Dans le chapitre 2, nous étudions les séries génératrices des traces dans les groupes de traces. Nous montrons l'existence d'une formule d'inversion semblable à la formule d'inversion de Möbius dans le monoïde de traces.

> Soient (Σ, D) un graphe fini non-orienté, $\bar{\Sigma}$ une copie de Σ et $(\tilde{\Sigma}, \tilde{D})$ le graphe construit à partir de (Σ, D) tel que $aDb \Rightarrow a\tilde{D}b$, $\bar{a}\tilde{D}b$, $a\tilde{D}\bar{b}$ et $\bar{a}\tilde{D}\bar{b}$.
> On définit $\mathbb{M}(\tilde{\Sigma}, \tilde{D})$ et $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ le monoïde et le groupe de traces par les représentations de monoïde suivantes :
> – $\mathbb{M}(\tilde{\Sigma}, \tilde{D})\{\tilde{\Sigma}^* \mid ab = ba \ \forall (a, b) \notin \tilde{D}\}$;
> – $\mathbb{G}(\tilde{\Sigma}, \tilde{D})\{\tilde{\Sigma}^* \mid ab = ba \ \forall (a, b) \notin \tilde{D}, \ a\bar{a} = \bar{a}a = 1 \forall a \in \Sigma\}$.
>
> **Théorème** *La série génératrice de $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ est*
>
> $$L_{\mathbb{G}}(x) = \sum_{n \in \mathbb{N}} \#\{g \in \mathbb{G}(\tilde{\Sigma}, \tilde{D}) | \ |g| = n\} x^n = \frac{1}{\mu_{\mathbb{M}}(1/(1+x))},$$
>
> *où* $\mu_{\mathbb{M}}(x) = \sum_{n \in \mathbb{N}} \#\{\textit{cliques de taille } n \textit{ dans } (\tilde{\Sigma}, \tilde{\Sigma} \times \tilde{\Sigma} - \tilde{D})\}(-1)^n x^n$.

Ce résultat généralise la formule de Möbius pour le monoïdes de traces de [28]. Une partie des résultats de cette partie sont publiés dans [23], les actes de la conférence DLT'03 et dans une note au CRAS, [24].

La deuxième partie concerne l'étude des réseaux de Petri temporisés. Le chapitre 3 présente les réseaux de Petri et les réseaux de Petri temporisés de manière plus formelle que dans cette introduction et donne une définition du débit, que nous allons étudier dans la suite de cette partie.

Dans le chapitre 4, nous étudions les réseaux de Petri à choix libres avec des temporisations constantes. Nous présentons des algorithmes de calculs du débit pour ces réseaux et étudions les politiques de résolution de conflit qui maximisent ou minimisent le débit. En particulier, nous montrons que dans certains cas, les débits extrémaux sont obtenus pour des routages périodiques.

> **Théorème** *Considérons un réseau de Petri à choix libres vivant 1-borné temporisé avec des temporisations réelles strictement positives. Le débit minimal est obtenu pour un routage périodique. Pour chaque place, la période de ce routage est bornée par le nombre maximal de jetons dans le circuit.*
>
> La preuve de ce théorème nécessite l'introduction de plusieurs outils originaux, comme les TT-invariants et les circuits critiques. La période est toujours bornée par le nombre de jetons dans le réseau, ce qui n'est pas vrai pour les réseaux non 1-bornés. Cela permet une algorithmique plus efficace pour trouver le débit minimal.

Ces résultats sont publiés dans [22], les actes de la conférence ICATPN'05 et acceptés pour publication dans [20].

Dans le chapitre 5, nous étudions le modèle plus général des réseaux stochastiques : les temporisations sont maintenant des suites de variables aléatoires. Nous comparons différentes politiques de résolution des conflits et établissons une hiérarchie de ces politiques.

> **Théorème** *Pour les politiques de compétition, de routage bernoulli et routage périodique, si les proportions de routages sont les mêmes, alors les débits des transitions sont proportionnels.*
> La meilleure politique, celle qui maximise le débit, peut consister à mélanger ces politiques classiques. En effet, on exhibe un exemple de réseau avec une place de choix pour lequel la meilleure politique « non-anticipative » est de choisir tantôt la politique de compétition, tantôt un routage déterministe.

Une version condensée de ce chapitre est publiée dans [21].

Dans le chapitre 6, nous étudions la simulation exacte (*exact sampling* ou *coupling from the past*) des réseaux de Petri. Nous adaptons l'algorithme de Propp et Wilson [80] au cas des réseaux de Petri à choix libres avec des temporisations exponentielles afin d'échantillonner de manière exacte les états du réseau.

> Dans un réseau à choix libres vivant borné, à chaque transition b, on peut canoniquement associer un marquage de blocage M_b.
> **Théorème** *Dans un graphe d'événements, ces marquages de blocage sont extrémaux au sens suivant : l'algorithme de simulation parfaite peut être exécuté en partant de ces seuls marquages.*
> Ceci permet d'utiliser l'algorithme de couplage arrière de Propp et Wilson de manière particulièrement effective. En effet, le nombre de marquages atteignables peut être exponentiellement grand par rapport à la taille du graphe, et ce théorème nous perment de ne calculer qu'un nombre linéaire de trajectoires.

Nous présentons une deuxième méthode de simulation exacte, plus générale dans les distributions des temporisations, mais sur la classe plus réduite des graphes d'événements, en utilisant les propriétés de l'algèbre (max,plus). Nous comparons ces deux méthodes. Ce chapitre a donné lieu au rapport de recherche [19].

La troisième partie est consacrée à l'étude des garanties de performance dans les réseaux. Cette étude a été faite en collaboration avec Éric Thierry.

Le chapitre 7 est consacré à l'étude du *network calculus*, un ensemble de techniques pour calculer des garanties de performances sur les réseaux. Les arrivées dans les réseaux et les courbes de service des serveurs sont données par des fonctions croissantes en fonction du temps, et des opérations sur ces fonctions définissent les relations entre les éléments du système. Nous étudions la stabilité des classes de fonctions les plus usuelles par ces opérations issues de l'algèbre (min,plus).

> **Théorème** *À une fonction $f : \mathbb{N} \to \mathbb{R}$ ultimement pseudo-périodique, on associe de façon canonique une matrice (min,plus) A telle que f^*, la clôture sous-additive de f, s'exprime comme $f^*(t) = A^t_{00}$.*
> Ceci permet en particulier de montrer que f^* est aussi ultimement pseudo-périodique.

De plus, nous développons des algorithmes efficaces de calculs de ces opérations.

Nous présentons et étudions dans le chapitre 8 une méthode pour éviter les blocages dans les réseaux : l'interdiction d'angles. En empêchant les flots de prendre certaines directions sachant leur provenance, on peut empêcher les blocages. Il s'agit ici de minimiser les contraintes (angles interdits), afin de préserver des garanties sur le réseau et de conserver la connexité du réseau. Dans un premier temps, nous décrivons des heuristiques pour trouver le nombre d'angles minimal à interdire pour casser les cycles. Ensuite, nous donnons ce nombre minimal pour certaines classes de graphes.

> **Théorème** *Le nombre minimal d'angles à interdire pour casser tous les cycles dans un graphe 3-régulier est $n/2 + c$, où n est le nombre de nœuds du graphe et c est le nombre de composantes 2-arêtes connexes.*

Ce chapitre est issu d'un travail en collaboration avec Fabrice Rastello.

Dans le chapitre 9 nous faisons la synthèse entre les chapitres 7 et 8. Nous faisons le lien entre l'étude des réseaux avec angles interdits et les méthodes du *network calculus*.

Chapitre 1

Rappels sur l'algèbre (max,plus) et les chaînes de Markov

Dans ce chapitre, nous rappelons la définition de l'algèbre (max, plus) et certaines de ses propriétés. Nous rappelons aussi quelques notions sur les chaînes de Markov, à temps discret et à temps continu ainsi que sur les processus de décision de Markov. Ces deux outils mathématiques sont ceux sur lesquels nous allons nous appuyer tout au long de ce mémoire.

1.1 Les algèbres (max,plus) et (min,plus)

Dans le chapitre d'introduction, nous avons vu que les opérateurs min, max et addition sont primordiaux pour l'étude des systèmes à événements discrets. En général, on utilise les opérations max et plus ou les opérations min et plus, ce qui a amené à l'étude algébrique de ces opérations, et à définir les algèbres dites tropicales, qui doivent leur nom au pays d'origine de Imre Simon, du Brésil, vers la fin des années 1970. Les algèbres tropicales ont été étudiées dans différents domaines de l'informatique : la théorie des automates ([79, 49, 57]), les réseaux de communication ([13]), le contrôle d'admission ([5, 6]), les réseaux de Petri ([50, 10])... Dans ce chapitre, nous présentons les algèbres (max,plus) et (min,plus), ainsi que les automates (max,plus). Pour plus de détails sur ces structures algébriques, on pourra se référer à [11].

1.1.1 Le semi-anneau (max,plus)

Définitions

Définition 1.1 *Un* monoïde *est un ensemble M muni d'une loi associative et d'un élément neutre.*

Un semi-anneau *est un ensemble A muni de deux lois internes, $\oplus$ et $\otimes$, telles que*
- *$(A, \oplus)$ est un monoïde commutatif d'élément neutre ε ;*
- *$(A, \otimes)$ est un monoïde d'élément neutre e ;*
- *la loi $\otimes$ est distributive par rapport à $\oplus$: $\forall a, b, c \in A$, $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ et $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$;*
- *$\forall a \in A$, $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.*

Si $(A, \otimes)$ est un monoïde commutatif, alors le semi-anneau $(A, \oplus, \otimes)$ est également commutatif.

Définition et proposition 1.2 *Soit $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$. Munissons $\mathbb{R}_{\max}$ de deux opérations internes, le maximum, noté $\oplus$ et l'addition, notée $\otimes$. Alors $\mathbb{R}_{\max}$ est un semi-anneau commutatif* *appelé* semi-anneau (max,plus) *d'élément nul* $\varepsilon = -\infty$ *et d'élément unité* $e = 0$.

De manière duale, on peut définir le semi-anneau (min,plus) $(\mathbb{R}_{\min}, \oplus, \otimes)$ où $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$, $\oplus$ est l'opérateur minimum, $\otimes$ est toujours l'addition, et $\varepsilon = +\infty$.

Tout ce qui suit est valable aussi bien pour (max,plus) que pour (min,plus), bien qu'ils ne soient présentés que pour l'algèbre (max,plus). Comme au cours de cette thèse les parties où l'on utilise l'algèbre (max,plus) (parties 1 et 2) sont bien distinctes de celles pour lesquelles on utilise l'algèbre (min,plus) (partie 3), on note de la même manière ces deux structures. Le tableau suivant donne la traduction de (max,plus) vers (min,plus).

Notations	$\mathbb{R}_{\max}$	$\mathbb{R}_{\min}$
$\oplus$	max	min
$\otimes$	$+$	$+$
ε	$-\infty$	$+\infty$
e	0	0
	$\leq$	$\geq$
	poids négatif	poids positif
	de poids le plus lourd	de poids le plus léger

On peut étendre le semi-anneau (max,plus) en la semi-algèbre des matrices (max,plus).

Les opérations dans cette semi-algèbre sont des adaptations directes des opérations sur les matrices de l'algèbre usuelle : si $\forall A, B \in \mathbb{R}_{\max}^{k\times k}$, $\forall \lambda \in \mathbb{R}_{\max}$,

$$\begin{aligned}
(A \oplus B)_{ij} &= A_{ij} \oplus B_{ij} && \text{« addition » interne,}\\
(A \otimes B)_{ij} &= \bigoplus_{\ell=1}^{k} A_{i\ell} \otimes B_{\ell j} && \text{« multiplication » interne,}\\
(\lambda \otimes A)_{ij} &= \lambda \otimes A_{ij} && \text{« multiplication » externe.}
\end{aligned}$$

La matrice nulle $\mathcal{E}_k \in \mathbb{R}_{\max}^{k\times k}$ est la matrice dont tous les coefficients sont ε, et la matrice identité $E_k \in \mathbb{R}_{\max}^{k\times k}$ celle dont tous les coefficients sont ε sauf les éléments diagonaux, qui sont égaux à e. La matrice $\mathbf{1} \in \mathbb{R}_{\max}^{k\times \ell}$ est la matrice dont tous les coefficients sont égaux à e.

Les opérations $\oplus$ et $\otimes$ sont aussi valables pour des matrices non carrées, dès que les matrices ont les bonnes dimensions (les règles sont les mêmes que pour l'algèbre classique). En particulier on peut multiplier des vecteurs, ligne ou colonne, par des matrices.

On peut représenter une matrice $A \in \mathbb{R}_{\max}^{k\times k}$ par un graphe orienté et pondéré à k sommets. Il y a un arc du nœud i vers le nœud j si $A_{ij} \neq \varepsilon$ et cet arc est alors pondéré par A_{ij}. On dit que A est irréductible si son graphe associé est fortement connexe.

Par convention $A^0 = E_k$, et $A^n = A^{n-1} \otimes A$. Soit $A \in \mathbb{R}_{\max}^{k\times k}$. Sous réserve d'existence, on définit A^* par $A^* = E_k \oplus A \oplus A^2 \oplus \cdots = \bigoplus_{n\in\mathbb{N}} A^n$.

Proposition 1.3 *Soit* $A \in \mathbb{R}_{\max}^{k\times k}$. *Alors,* A^* *existe si et seulement si tous les circuits du graphe associé à* A *sont de poids négatifs ou nuls.*

Théorème 1.4 *Soit l'équation*

$$X = (X \otimes A) \oplus B,$$

avec $A \in \mathbb{R}_{\max}^{k\times k}$, $X, B \in \mathbb{R}_{\max}^{1\times k}$. *La plus petite solution de cette équation est* $X_0 = B \otimes A^*$. *Si* A *est irréductible, cette solution est finie si et seulement si* A^* *existe.*

Analyse spectrale des matrices (max,plus)

Le résultat du théorème suivant est facile à interpréter en terme de graphes. C'est un théorème spectral qui permet de déterminer les valeurs propres des matrices (max,plus).

Soit $A \in \mathbb{R}_{\max}^{n\times n}$. Le vecteur $u \in \mathbb{R}_{\max}^{n}$ est un *vecteur propre* de A associé à la *valeur propre* λ si $u \neq (\varepsilon,\ldots,\varepsilon)$ et $A \otimes u = \lambda \otimes u$.

Théorème 1.5 *Une matrice irréductible A de $\mathbb{R}_{\max}^{k\times k}$ admet une unique valeur propre*

$$\rho(A) = \max_{\ell=1}^{k} \frac{\max_{i=1}^{k} A_{ii}^{\ell}}{\ell}.$$

Si A est une matrice de $\mathbb{R}_{\max}^{k\times k}$ non nécessairement irréductible, alors la plus grande valeur propre de A est

$$\rho(A) = \max_{\ell=1}^{k} \frac{\max_{i=1}^{k} A_{ii}^{\ell}}{\ell}.$$

L'interprétation en terme de graphes est que la valeur propre de A est égale au poids moyen du circuit de poids moyen le plus grand du graphe associé à A. En effet, le poids du chemin de i à j de longueur ℓ le plus lourd dans A est A_{ij}^{ℓ}.

Définition 1.6 *Soit $A \in \mathbb{R}_{\max}^{n\times n}$ une matrice irréductible, de graphe associé $\mathcal{G}(A)$. Soit $\mathcal{G}_c(A)$ le sous-graphe de $\mathcal{G}(A)$ obtenu en ne gardant que les arcs qui appartiennent à un circuit de poids moyen maximal. On appelle $\mathcal{G}_c(A)$ le* graphe critique *de A. On appelle nœuds critiques de A les nœuds de $\mathcal{G}_c(A)$, dont l'ensemble est noté crit(A). Soit $\mathcal{C}$ une composante fortement connexe de $\mathcal{G}_c(A)$. La cyclicité de $\mathcal{C}$, notée $c_{\mathcal{C}}$, est le plus grand diviseur commun des longueurs des circuits de $\mathcal{C}$. La* cyclicité *de A est $c = \vee_{\mathcal{C}} c_{\mathcal{C}}$, le plus petit multiple commun des cyclicités des composantes fortement connexes de $\mathcal{G}_c(A)$.*

Un autre théorème fondamental montre que le comportement de la suite $(A^n)_{n\in\mathbb{N}}$ est ultimement périodique :

Théorème 1.7 *Soit $A \in \mathbb{R}_{\max}^{k\times k}$ une matrice irréductible de cyclicité c et de valeur propre λ. Alors il existe un entier $n_0 \in \mathbb{N}$ tel que $\forall n \geq n_0$,*

$$A^{n+c} = \lambda^c \otimes A^n.$$

Le théorème ne dit rien sur l'entier à partir duquel le comportement de (A^n) est régulier. En effet, trouver la valeur exacte pour le comportement régulier est un problème difficile. Néanmoins, nous avons montré dans [18] comment calculer une borne supérieure de ce rang.

Théorème 1.8 ([18], théorème 2) *Soit $A \in \mathbb{R}_{\max}^{k\times k}$ une matrice irréductible, de cyclicité 1 et de valeur propre λ. Le régime périodique de la suite $(A_n)_{n\in\mathbb{N}}$ est atteint à partir du rang*

$$n_1 = k + \max(\frac{|B^* - M'|}{-\beta}, 2k + k^2),$$

où

- *M est la matrice définie par $M_{ij} = \bigoplus_{\ell \in crit(A)} (A-\lambda)_{i\ell}^* \otimes (A-\lambda)_{\ell j}^*$,*
- *B est la sous-matrice des nœuds non critiques de $A - \lambda$,*
- *M' est la sous-matrice des nœuds non critiques de M, et*
- *β est la plus grande valeur propre de B.*

Le calcul de cette borne peut se faire en temps cubique en la dimension de la matrice.

1.1.2 Automates (max,plus)

Définissons maintenant les automates (max,plus) et les empilements de pièces. Ces modèles sont en particulier utilisés pour l'ordonnancement de tâches ([25, 61]), les pièces représentant les tâches à effectuer, et les colonnes occupées par une pièce, les ressources utilisées pour effectuer la tâche.

Définition 1.9 *Soit A un alphabet. Un automate (max,plus) est un quadruplet $\mathcal{A} = (Q, I, F, \mathcal{M})$ où*
- *Q est un ensemble fini d'états,*
- *$I \in \mathbb{R}_{\max}^{1\times Q}$ et $F \in \mathbb{R}_{\max}^{Q\times 1}$ sont les vecteurs initial et final,*
- *$\mathcal{M}$ est un morphisme multiplicatif : $A^* \to \mathbb{R}_{\max}^{Q\times Q}$.*

Le morphisme $\mathcal{M}$ est uniquement défini par la famille $\mathcal{M}(a)$, $a \in A$. Pour un mot $w = a_1 \dots a_n$, on a

$$\mathcal{M}(w) = \mathcal{M}(a_1) \dots \mathcal{M}(a_n),$$

où le produit de matrices est effectué dans le semi-anneau (max,plus).

Les automates (max,plus) sont un cas particulier des automates avec multiplicité (voir [84] pour un développement de cette théorie).

On dit qu'un mot w est *reconnu* si $I\mathcal{M}(w)F \neq -\infty$. On retrouve les mêmes notions que pour les automates sans multiplicité : un mot est reconnu si et seulement s'il a un chemin acceptant ; si l'automate est déterministe, chaque mot a au plus un chemin acceptant. On appelle *série reconnue* par l'automate l'application $S : A^* \to \mathbb{R}_{\max}$ définie par $(S|w) = I\mathcal{M}(w)F$. On note $S = \sum(S|w)w$.

Définition 1.10 *Un empilement de pièces est un quintuplet $\mathcal{H} = (T, R, \mathcal{R}, l, u)$ où*
- *T est un ensemble de pièces,*
- *R est un ensemble de colonnes,*
- *$\mathcal{R} : T \to \mathcal{P}(R)$ associe à chaque pièce t l'ensemble des colonnes qu'elle occupe,*
- *$l, u : T \times R \to \mathbb{R}_{max}$ sont les contours inférieur et supérieur de chaque pièce.*

Partant d'un sol plat, pour un mot w si on empile les pièces correspondant aux lettres de w dans l'ordre, on obtient un tas, et chaque colonne a une certaine hauteur. La hauteur de l'empilement w, notée $h(w)$, est alors le maximum de la hauteur sur toutes les colonnes. On peut modéliser cette fonction hauteur par un automate (max,plus).

En effet, si $\mathcal{H} = (T, R, \mathcal{R}, l, u)$, l'automate $\mathcal{A} = (R, \mathbf{1}, \mathbf{1}, \mu)$ sur l'alphabet T avec

$$\mathcal{M}(a)_{sr} = \begin{cases} 0 & \text{si } s = r \notin \mathcal{R}(a), \\ u(a,r) - l(a,s) & \text{si } r, s \in \mathcal{R}(a), \\ -\infty & \text{sinon.} \end{cases} \tag{1.1}$$

calcule la hauteur du tas de pièces.

Exemple 1.11 Considérons un empilement de pièces, avec trois ressources, $R = \{1, 2, 3\}$ et deux pièces, $T = \{a, b\}$, avec :
- $\mathcal{R}(a) = \{1, 2\}$, $l(a) = (0, 0, \varepsilon)$ et $u(a) = (3, 1, \varepsilon)$,
- $\mathcal{R}(b) = \{2, 3\}$, $l(b) = (\varepsilon, 0, 0)$ et $u(b) = (\varepsilon, 1, 2)$.

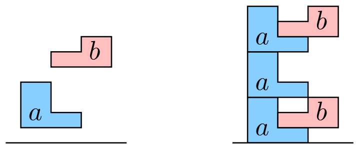


FIG. 1.1 – Empilement de pièces.

La forme des pièces et l'empilement du mot *abaab* est alors représenté dans la figure 1.1.

Les matrices de l'automate (max,plus) correspondant à ce tas de pièces sont

$$\mathcal{M}(a) = \begin{bmatrix} 3 & 1 & \varepsilon \\ 3 & 1 & \varepsilon \\ \varepsilon & \varepsilon & 0 \end{bmatrix} \qquad \mathcal{M}(b) = \begin{bmatrix} 0 & \varepsilon & \varepsilon \\ \varepsilon & 1 & 2 \\ \varepsilon & 1 & 2 \end{bmatrix}$$

Graphiquement, l'automate est représenté en figure 1.2.

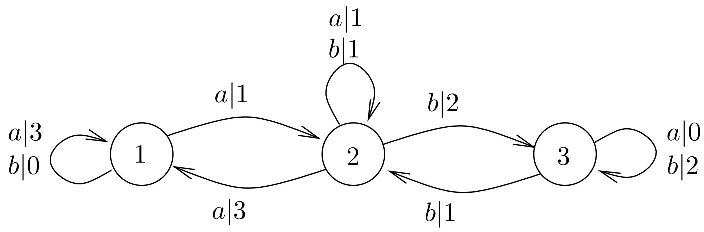


FIG. 1.2 – Automate (max,plus).

La matrice de l'empilement de la figure 1.1 est

$$\mathcal{M}(abaab) = \begin{bmatrix} 9 & 8 & 9 \\ 9 & 8 & 9 \\ 7 & 6 & 7 \end{bmatrix}.$$

1.2 Rappels sur les chaînes de Markov

Dans cette section, nous rappelons certaines définitions et quelques propriétés des chaînes de Markov. Nous nous intéressons aux chaînes de Markov à temps discret, puis aux chaîne de Markov à temps continu avant de définir les processus de décision de Markov.

1.2.1 Chaînes de Markov à temps discret

Soit $(X_n)_{n\in\mathbb{N}}$ une suite de variables aléatoires (v.a.) à valeurs dans S, ensemble fini ou dénombrable, l'*espace des états*. On dit que $(X_n)_{n\in\mathbb{N}}$ est une *chaîne de Markov (homogène)* si pour tout $n \geq 0$, pour tous $x_0, \ldots, x_{n+1} \in S$,

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, \ldots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$$

et si la probabilité de transition de l'état x à l'état y, $\mathbb{P}(X_{n+1} = y | X_n = x)$ ne dépend pas de n quels que soient les états x et y. On note $P_{x,y}$ la probabilité $\mathbb{P}(X_{n+1} = y | X_n = x)$.

La matrice $P = (P_{x,y})_{x,y\in S}$ est appelée *matrice de transition* de la chaîne de Markov. C'est une matrice stochastique : $P_{x,y} \geq 0$, $\forall x, y \in S$ et $\sum_{y\in S} P_{x,y} = 1$ $\forall x \in S$. On a, pour tous $x, y \in S$ et tous $n, m \in \mathbb{N}$, $n > m$,

$$\mathbb{P}(X_n = y | X_m = x) = P^{n-m}_{x,y},$$

où P^{n-m} est la puissance $(n-m)$-ième de P.

Représentations des chaînes de Markov

La relation de récurrence $X_{n+1} = f(X_n, \varepsilon_{n+1})$, $n \geq 0$, où $f : S \times F \to S$, $(\varepsilon_n)_{n \in \mathbb{N}^*}$ est une suite de variables aléatoires i.i.d. indépendantes de X_0 et à valeurs dans F, un ensemble dénombrable, définit une chaîne de Markov homogène. Une telle représentation d'une chaîne de Markov est appelée *représentation fonctionnelle*.

On peut aussi représenter une chaîne de Markov par un graphe orienté dont les nœuds sont les états de la chaîne et où il y a un arc de x vers y étiqueté par $P_{x,y}$ si $P_{x,y} > 0$. La chaîne de Markov est dite *irréductible* si son graphe est fortement connexe. Pour chaque *classe de communication* (*i.e.* composante fortement connexe du graphe), on définit la période de cette classe comme le plus grand diviseur commun des longueurs des circuits élémentaires de cette composante. La période de la chaîne est le plus petit multiple commun des périodes de ses classes de communication. Si une chaîne de Markov est irréductible et a une période de 1, on dit aussi qu'elle est *apériodique*.

Distribution stationnaire

Soit $\pi = (\pi_x)_{x \in S}$ une distribution de probabilité sur S. On dit que π est une *distribution stationnaire* de (X_n) si $\pi = \pi P$. Si X_0 est distribuée selon π, alors le processus (X_n) est *stationnaire* : la distribution de X_n est la même que celle de X_0 pour tout entier n.

On distingue plusieurs classes d'états : les *états récurrents* et les *états transients*. Un état x est récurrent si la probabilité de retourner dans cet état sachant qu'on en est parti est 1. Sinon, cet état est transient. Parmi les états récurrents, on distingue les états récurrents *positifs*, pour lesquels le temps moyen de retour est d'espérance finie, des autres états, les états récurrents *nuls*. À l'intérieur d'une même classe de communication, les états sont du même type. En particulier, pour une chaîne de Markov irréductible à espace d'état fini, tous les états sont récurrents positifs. Pour une chaîne irréductible, on dit qu'elle est récurrente positive (resp. récurrente nulle, resp. transiente) si ses états sont récurrents positifs (resp. récurrents nuls, resp. transients).

Théorème 1.12 *Soit $(X_n)_{n \in \mathbb{N}}$ une chaîne de Markov irréductible. Cette chaîne est récurrente positive si et seulement si elle a une distribution stationnaire. Dans ce cas, cette distribution stationnaire est unique.*

Théorème 1.13 *Soit (X_n) une chaîne de Markov irréductible, apériodique et récurrente positive (une telle chaîne est dite* ergodique*). Soit π sa distribution stationnaire. Alors, $\forall x, y \in S$,*

$$P^n_{x,y} \to_{n \to \infty} \pi_y.$$

On en déduite aisément que

$$\mathbb{P}(X_n = y) \to_{n \to \infty} \pi_y$$

et

$$\frac{1}{n} \sum_{k=1}^{n} \mathbb{P}(X_k = y) \to_{n \to \infty} \pi_y.$$

La dernière limite signifie que partant d'une distribution quelconque la moyenne du temps passé dans un état y est égale à la probabilité de se trouver dans cet état sous la distribution stationnaire.

1.2.2 Chaînes de Markov à temps continu

Soit $X = (X(t))_{t\in\mathbb{R}_+}$ une famille de variables aléatoires à valeurs dans S, ensemble fini ou dénombrable. Le processus X est une chaîne de Markov à temps continu si

$$\mathbb{P}(X(t_{n+1}) = x_{n+1}|X(t_1) = x_1, \ldots, X(t_n) = x_n) = \mathbb{P}(X(t_{n+1}) = x_{n+1}|X(t_n) = x_n),$$

quels que soient $x_1, \ldots, x_{n+1}$ et $0 \le t_1 < t_2 < \cdots < t_n$ et si $\mathbb{P}(X(t) = y|X(s) = x)$ ne dépend que de la différence entre t et s Une telle chaîne est dite à temps continu car le processus est défini en tout point de $\mathbb{R}_+$. Par opposition, ce que nous avons défini plus haut comme chaîne de Markov est aussi appelé chaîne de Markov à temps discret car défini sur les naturels, des indices «discrets».

On a $\mathbb{P}(X(t) = y|X(s) = x) = \mathbb{P}(X(t-s) = y|X(0) = x)$. On note $p_{x,y}(t-s)$ cette quantité. On a donc défini une famille de matrices $P(t) = (p_{x,y}(t))_{x,y\in S}$ telle que :

- $P(0)$ est la matrice identité ;
- $\forall t \in \mathbb{R}_+$, $P(t)$ est une matrice stochastique ;
- $\forall s, t \in \mathbb{R}_+$, $P(t+s) = P(t)P(s)$.

On supposera toujours que $p_{x,y}$ est continue. Donc pour tous $x, y \in S$, $p_{x,y}(t) \to_{t\to 0}= 0$ si $x \neq y$ et $p_{x,x}(t) \to_{t\to 0}= 1$.

Regardons ce qui peut se passer dans un petit intervalle de temps, $[t, t+h[$, quand $X(t) = x$. Ou bien rien ne s'est passé : la chaîne reste pendant tout cet intervalle dans le même état. Ceci a lieu avec une probabilité $p_{x,x}(h) + o(h)$. Le terme $o(h)$ est un terme correcteur qui prend en compte la probabilité que la chaîne ait changé plusieurs fois d'état et soit revenue en x. Soit la chaîne a changé d'état et se trouve dans un nouvel état y, avec une probabilité $p_{x,y}(h)+o(h)$. On a supposé ici que la probabilité que deux changements d'état aient lieu dans un petit intervalle de temps est négligeable. Ceci peut être prouvé, ainsi que le fait que les termes $p_{x,y}(h)$ sont presque linéaires quand h est petit. Il existe alors des constantes $q_{x,y}$, $x, y \in S$ telles que

$$p_{x,y}(h) = q_{x,y}h + o(h) \text{ si } x \neq y \qquad \text{et} \qquad p_{x,x}(h) = 1 + q_{x,x}h + o(h).$$

On a $q_{y,x} \ge 0$ si $x \neq y$ et $q_{x,x} \le 0$. La matrice $Q = (q_{x,y})$ est appelée *générateur infinitésimal* de la chaîne et joue un rôle similaire à celui de P pour les chaînes de Markov à temps discret. Comme $\sum_{y\in S} p_{x,y} = 1$, on a $1 + (\sum_{y\in S} q_{x,y})h = 1$ et $\sum_{y\in S} q_{x,y} = 0$.

Soit x un état tel que $X(s) = x$. Soit $U = \inf\{t \ge 0 \mid X(s+t) \neq x\}$, le temps passé par la chaîne dans l'état x à partir de la date s. Alors, U est exponentiellement distribuée de paramètre $-q_{x,x}$. La probabilité que l'état atteint juste après x soit y est $-q_{x,y}/q_{x,x}$. De plus, l'état atteint et U sont indépendants.

On suppose que $\sup_x -Q_{xx} < \infty$. La chaîne X est irréductible si pour toute paire d'états x, y, il existe t tel que $p_{x,y}(t) > 0$. La distribution π est stationnaire si $\pi = \pi P(t)$ pour tout t. Alors, une distribution π est stationnaire si et seulement si $\pi Q = 0$. Si cette distribution stationnaire existe et que la chaîne est irréductible, alors cette distribution est unique et

$$\forall x, y \in S \quad p_{x,y}(t) \to \pi_y \text{ quand } t \to \infty.$$

Uniformisation

À partir d'une chaîne de Markov à temps continu X, on peut construire une chaîne de Markov à temps discret ayant la même distribution stationnaire. Cette transformation est appelée *uniformisation* et définie comme suit. Soit $q > \sup_{y \in S} -q_{y,y}$. La chaîne de Markov à temps discret qui a pour matrice de transition P définie par

$$P_{x,y} = \begin{cases} q_{x,y}/q & \text{si } x \neq y \\ 1 + q_{x,x}/q & \text{si } x = y \end{cases} \tag{1.2}$$

a la même probabilité stationnaire que X. La constante q correspond à la discrétisation du temps. Plus la constante q d'uniformisation est grande, plus la probabilité de rester dans le même état est grande.

Cette transformation peut s'avérer très utile pour appliquer des théorèmes valables pour les chaînes de Markov à temps discret à celles à temps continu.

1.2.3 Processus de décision de Markov

Les chaînes de Markov que nous venons de définir servent à modéliser des systèmes à événements discrets stochastiques et à évaluer certaines performances de ces systèmes. Si l'on a affaire à des systèmes que l'on peut contrôler en prenant des décisions, on cherche à améliorer les performances du système en prenant les meilleures décisions. Ceci peut se faire au moyen de processus de décision de Markov, encore appelés chaînes de Markov contrôlées.

Étant donné un système à événements discrets décrit par l'équation de récurrence

$$x_{k+1} = f_k(x_k, u_k, w_k),$$

où x_k est l'état à l'étape k, u_k est une décision prise à l'étape k ($u_k \in U$, où U est l'ensemble des décisions possibles), $(w_k)_{k \in \mathbb{N}}$ est une suite de variables aléatoires i.i.d. et f_k est l'équation d'évolution à l'étape k, et une fonction de gain à l'étape k, $g_k(x_k, u_k, w_k)$, on cherche les décisions qui minimisent le coût moyen. C'est-à-dire, on veut trouver une suite $u_1^*, u_2^*, \ldots$ qui maximise le gain moyen à chaque étape, c'est-à-dire,

$$J^* = \lim \inf_{N \to \infty} \frac{1}{N} \mathbb{E} \sum_{k=1}^{N} g_k(x_k, u_k^*, w_k),$$

où l'espérance est prise sur les w_k. Les décisions sont prises alors que l'état du système est connu. On dit alors qu'on est en *boucle fermée*, et on cherche une fonction π qui à un état x_k associe une décision u_k qui maximise l'espérance du gain moyen par étape.

On suppose dans toute la suite que la dynamique est stationnaire ($\forall k \in \mathbb{N},\ f_k = f_1$), que la fonction de coût est stationnaire ($\forall k \in \mathbb{N},\ g_k = g_1$). On note alors $P_{i,j}(u)$ la probabilité que $f_1(i, u, w) = j$ et $g(i, u)$ l'espérance de gain en une étape dans l'état i quand la décision u est prise.

On dit qu'une politique de choix est *stationnaire* si la décision u_k ne dépend que de x_k et pas de k.

Le gain total maximal en N étapes que l'on peut obtenir peut se calculer par programmation dynamique. À l'étape N, le gain ne varie plus. Posons alors $J^N = (0, \ldots, 0)$. Si l'on connaît l'espérance maximale du gain entre l'étape $k+1$ et l'étape N en partant de l'état i, notée

$J^{k+1}(i)$ pour chaque état i, on peut calculer l'espérance maximale du gain à partir de l'étape k. C'est

$$J^k(i) = \max_u(g(i,u) + \sum_j P_{i,j}(u)J^{k+1}(j)).$$

Soit T l'opérateur tel que

$$TJ(i) = \max_u(g(i,u) + \sum_j P_{i,j}(u)J(j)).$$

On note e le vecteur dont toutes les coordonnées sont égales à 1.

Proposition 1.14 ([16], Vol. II, Chap. 4, Proposition 2.1) *S'il existe un réel λ et un vecteur h satisfaisant*

$$\lambda + h(i) = \max_u(g(i,u) + \sum_j P_{i,j}(u)h(j)),$$

soit

$$\lambda e + h = Th,$$

alors λ est le gain moyen optimal par étape, et s'il existe une politique π satisfaisant le maximum pour tout i, alors cette politique est optimale. Le gain moyen maximal est indépendant de l'état initial.

Il y a plusieurs manières de prouver l'existence de λ et h. Une condition suffisante et facile à prouver dans les cas qui nous intéresseront est que pour tout couple (i,j) d'états, il existe une politique stationnaire π et $k \in \mathbb{N}$ tels que $\mathbb{P}(x_k = j | x_0 = i, \pi) > 0$.

La politique optimale et le gain moyen optimal peuvent s'obtenir par itération des valeurs : on a

$$J^* = \lambda = \lim_{k\to\infty} \frac{T^k(J_0)(i)}{k} \quad \forall i.$$

Première partie

Groupes de traces

Chapitre 2

Étude des groupes de traces

Les systèmes concurrents sont des systèmes où plusieurs événements peuvent avoir lieu en même temps. Le temps d'exécution (exécution parallèle) est inférieur à la somme des temps d'exécution (exécution séquentielle) de toutes les actions. On modélise les événements par des lettres et un mot correspond à une suite d'actions. Pour représenter la simultanéité de plusieurs actions, on autorise les lettres les représentant à commuter. Ainsi plusieurs mots sur l'alphabet des actions représentent en fait la même suite d'événements, à l'ordre près des événements simultanés, qui bien sûr n'a pas d'importance. La structure qui permet cette modélisation est le monoïde de traces, qui a été étudié par exemple dans [36, 28, 37, 49, 38], en particulier pour l'étude de la concurrence dans les réseaux de Petri. Une mesure importante pour les systèmes concurrents est le rapport moyen entre temps d'exécution parallèle et temps d'exécution séquentielle ([64]).

La structure de monoïde peut facilement être étendue en une structure de groupe : on ajoute pour chaque lettre une lettre inverse. Pour des systèmes concurrents, une interprétation pourrait être qu'à chaque tâche correspond une tâche annulation.

D'un point de vue algébrique, ces groupes ont été étudiés par de nombreux auteurs sous le nom de *graph groups* et *right-angled Artin groups*. La motivation pour les étudier vient de ce qu'ils sont une des classes simples des groupes d'Artin.

Tout d'abord, Carl Droms a étudié d'un point de vue algébrique les groupes de traces, qu'il appelle *graph groups*, car ces groupes sont construits à partir d'un graphe d'indépendance, comme nous le verrons dans la suite. Il a montré certaines propriétés de ces groupes : les groupes de traces dont tous les sous-groupes sont aussi des groupes de traces sont les groupes construits à partir des graphes d'indépendance qui ne contiennent ni un carré, ni une ligne de quatre sommets comme sous-graphe ([39]). Il a aussi étudié les sous-groupes des groupes construits à partir du groupe à un générateur, du produit libre et du produit cartésien (qui sont des groupes de traces). Enfin, il s'est intéressé à des généralisations des groupes de traces, comme les groupes d'Artin.

Ces groupes ont également été étudiés en combinatoire, avec des applications en physique théorique, car elle permet d'obtenir des résultats quant à la structure de chaînes de molécules. Dans [90], les auteurs étudient les groupes de traces construits à partir des graphes lignes à n nœuds, pour n grand, et en déduisent des résultats pour les groupes de tresses.

Dans ce chapitre, nous étudions les groupes de traces d'un point de vue combinatoire. C'est une généralisation d'une étude faite pour les monoïdes de traces dans [64]. Dans une première partie, nous allons définir plus formellement les différentes structures citées dans cette introduction, puis nous étudierons les propriétés combinatoires des fonctions génératrices des éléments de ces ensembles. Enfin, nous nous attacherons au comportement asymptotique des séries génératrices, qui permettent de calculer la quantité moyenne de parallélisme. De plus, nous montrerons une application de ces résultats à des modèles tâches-ressources.

La principale contribution nouvelle est le théorème 2.22, qui a été publiée dans [23] et [24]. Ce théorème donne une formule simple de la série génératrice d'un groupe de traces. Les autres résultats de ce chapitre découlent directement de cette formule.

2.1 Monoïdes et groupes de traces

2.1.1 Définition et notations

Monoïde de traces

Soit Σ un alphabet fini, D une relation réflexive et symétrique sur Σ et I son complémentaire sur $\Sigma \times \Sigma$. On appelle D relation de dépendance et I relation d'indépendance. On définit une structure de monoïde à partir du monoïde libre sur Σ. Deux lettres en relation dans I commutent : elles représentent des événements qui peuvent avoir lieu en parallèle. On a donc une relation d'équivalence sur les mots : deux mots sont équivalents si l'un peut être obtenu à partir de l'autre en faisant commuter des lettres adjacentes en relation dans I. On a ainsi la définition plus formelle :

Définition 2.1 *Un* monoïde de traces $\mathbb{M}(\Sigma, D)$ *est le quotient du monoïde libre* Σ^* *par la plus petite congruence générée par les relations* $ab \sim ba$, $\forall aIb$. *Les éléments de ce monoïde, aussi appelé monoïde libre partiellement commutatif, sont appelés les traces.*

Les graphes non-orientés (Σ, D) et (Σ, I) sont appelés graphe de dépendance et graphe d'indépendance de $\mathbb{M}(\Sigma, D)$. Les deux cas extrêmes, pour un alphabet Σ fixé, sont le monoïde libre ($D = \Sigma \times \Sigma$), et le monoïde commutatif libre d'alphabet Σ ($D = id_\Sigma$, et $\mathbb{M}(\Sigma, D)$ est isomorphe à $\mathbb{N}^{|\Sigma|}$).

On définit aussi $\pi_{\mathbb{M}} : \Sigma^* \to \mathbb{M}(\Sigma, D)$ le morphisme surjectif canonique qui à chaque mot associe la trace qu'il représente.

Deux mots représentant la même trace sont de la même longueur. On dit alors que la longueur de la trace t est la longueur d'un mot qui la représente et on la note $|t|$.

Soit w un mot. Le support de w, noté $\mathrm{alph}(w)$, est l'ensemble des lettres qui apparaissent dans w. Soit u une trace. Tous les mots représentant cette trace ont le même support. On note par $\mathrm{alph}(u)$ le support d'un représentant de u.

Une *clique* est une trace non vide dont toutes les lettres sont mutuellement indépendantes. Les cliques sont en fait les cliques (c'est-à-dire les sous-graphes complets) de (Σ, I). On note $\mathfrak{C}$ l'ensemble des cliques d'un monoïde de traces $\mathbb{M}$.

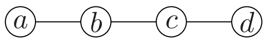
Définition 2.2 *Un couple* $(u, v) \in \mathfrak{C} \times \mathfrak{C}$ *est* Cartier-Foata (CF-) admissible *si pour toute lettre* b *apparaissant dans* v *il existe* a *apparaissant dans* u *tel que* $(a, b) \in D$:

$$\forall b \in \mathrm{alph}(v)\ \exists a \in \mathrm{alph}(u) \textit{ tel que } (a, b) \in D.$$

Proposition 2.3 *La décomposition de Cartier-Foata (CF) d'une trace* t *est uniquement définie par la suite de cliques* $(c_1, c_2, \ldots, c_n)$ *telle que* $t = c_1 c_2 \ldots c_n$ *et pour tout* $i \in \{1, \ldots, n-1\}$, (c_i, c_{i+1}) *est CF-admissible. L'entier* n *est appelé hauteur de la trace* t*, et noté* $h(t)$.

Cette définition permet de construire le graphe des cliques du monoïde $\mathbb{M}(\Sigma, D)$: c'est le graphe orienté dont les nœuds sont les cliques, et (u, v) est un arc si (u, v) est CF-admissible.

Exemple 2.4 On considère $\Sigma = \{a, b, c, d\}$, et le graphe de dépendance :



Soit $w = a\,c\,d\,b\,a\,b\,d\,c\,b$ un mot et soit $t = \pi_{\mathbb{M}}(w)$ la trace qu'il représente. On a alors $|t| = 9$, et la décomposition CF est : $\{a, c\}, \{b, d\}, \{a, d\}, \{b\}, \{c\}, \{b\}$. Donc la hauteur est $h(t) = 6$.

Le graphe des cliques est représenté par la figure 2.1 (les boucles, sur chaque nœud, n'ont pas été représentées).

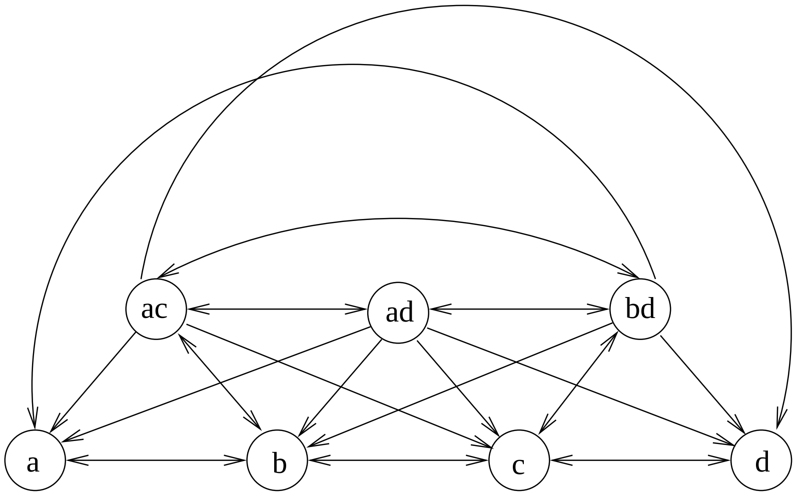


FIG. 2.1 – Graphe des cliques du graphe de l'exemple 2.4

Groupe de traces

La structure de monoïde s'étend en une structure de groupe : à chaque lettre a de Σ, on associe une lettre, notée $\bar{a}$. On note aussi l'alphabet $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$, avec $\Sigma \cap \bar{\Sigma} = \emptyset$.

On construit un groupe de traces $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ à partir d'un monoïde de traces $\mathbb{M}(\Sigma, D)$ de la manière suivante :

- si l'alphabet de $\mathbb{M}$ est Σ, alors celui de $\mathbb{G}$ est $\tilde{\Sigma} = \Sigma \cup \bar{\Sigma}$;
- si dans $\mathbb{M}$ on a aDb, alors dans $\mathbb{G}$, on a $a\tilde{D}b$, $a\tilde{D}\bar{b}$, $\bar{a}\tilde{D}b$ et $\bar{a}\tilde{D}\bar{b}$;
- $\forall a \in \Sigma$, $a\bar{a} = \bar{a}a = \varepsilon$.

En particulier, la deuxième condition implique que $\forall a \in \Sigma$, on a $a\tilde{D}\bar{a}$.

Les éléments de $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ sont aussi appelés des *traces*.

On dit que $\tilde{D}$ est la relation de dépendance de $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$, et la relation d'indépendance du groupe $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ est $\tilde{I}$, le complémentaire de $\tilde{D}$.

On peut définir un groupe de traces de façon plus formelle :

Définition et proposition 2.5 *Un groupe de traces est le quotient de $\tilde{\Sigma}$ par les relations $ab = ba$ si $a\tilde{D}b$ et $a\bar{a} = \bar{a}a = \varepsilon$, $\forall a \in \Sigma$. Ce quotient définit bien un groupe.*

Remarque 2.6 Cette définition ne fait pas apparaître les lettres barrées, car si $ab = ba$, on a $\bar{a}ab = b = \bar{a}ba$ donc $b\bar{a} = \bar{a}ba\bar{a} = \bar{a}b$: les autres relations se déduisent de celles qui sont données.

Remarque 2.7 On sait que a et $\bar{a}$ commutent puisqu'ils sont inverses l'un de l'autre, mais par convention ils ne sont pas indépendants.

Notations : Dans toute la suite, on note $\tilde{\Sigma} = \Sigma \cup \bar{\Sigma}$ l'alphabet du groupe, $\tilde{D}$ et $\tilde{I}$ les relations de dépendance et d'indépendance du groupe. On garde les notations Σ, D et I pour l'alphabet et les relations du monoïde associé.

De même que pour les monoïdes, on définit un morphisme surjectif de monoïde canonique $\pi_{\mathbb{G}} : \tilde{\Sigma}^* \to \mathbb{G}(\tilde{\Sigma}, \tilde{D})$. On remarque que la restriction de $\pi_{\mathbb{G}}$ à Σ^* est exactement le morphisme canonique π_m sur $\mathbb{M}(\Sigma, D)$.

Tous les mots représentant la même trace n'ont pas la même longueur. On a donc une définition légèrement différente de la longueur d'une trace t pour le groupe : c'est la longueur d'un plus petit mot représentant la trace t.

$$|t| = \min\{|w|, w \in \pi_{\mathbb{G}}^{-1}(t)\}$$

De même, tous les mots représentant la même trace n'ont pas le même support. On définit alors le support d'une trace comme le support d'un mot la représentant et de longueur minimale. On note le support de u par $\mathrm{alph}_G(u)$ et on a $\mathrm{alph}_G(u) = \cap_{w \in \pi_{\mathbb{G}}^{-1}(u)} \mathrm{alph}(w)$.

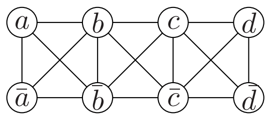
Nous allons aussi donner une nouvelle définition des cliques CF-admissibles : en effet, la superposition de a et $\bar{a}$ pose problème, puisque si ces deux lettres sont dans des cliques adjacentes dans un mot, elles s'annulent.

Définition 2.8 *Soient u et v deux cliques. On dit que (u, v) est CF-admissible si pour tout $b \in \mathrm{alph}_G(v)$ il existe $a \in \mathrm{alph}_G(u)$ tel que $(a, b) \in D$ et si $a \in \mathrm{alph}_G(v)$, alors $\bar{a} \notin \mathrm{alph}_G(u)$:*

$$\forall b \in \mathrm{alph}_G(v),\ \Big((\bar{b} \notin \mathrm{alph}_G(u)) \text{ et } (\exists a \in \mathrm{alph}_G(u)\ a\tilde{D}b) \Big).$$

Cette définition permet ainsi de définir la hauteur d'une trace et de construire le graphe des cliques de la même manière que pour les monoïdes.

Exemple 2.9 On construit le groupe de traces à partir du monoïde de l'exemple 1. On a donc le graphe de dépendance ci-dessous (on omet les boucles).



Soit $w = c\,d\,\bar{c}\,a\,c\,b\,\bar{a}\,d$. La lettre c commute avec a, donc le mot $w' = cd\bar{c}cab\bar{a}d = cdab\bar{a}d$ représente la même trace que w. Les lettres a et b ne commutent pas, donc on ne peut plus trouver de mots de longueur inférieure qui représente la même trace. Donc la longueur de la trace est : $|\pi_g(w)| = 6$ et la décomposition en cliques CF-admissibles est $\{a, c\}, \{b, d\}, \{\bar{a}, d\}$: $h(\psi(t)) = 3$.

Relations entre les différentes structures. Il existe différentes fonctions qui relient ces ensembles ainsi que les monoïdes libres Σ^* et $\tilde{\Sigma}^*$ et le groupe libre sur Σ noté $F(\Sigma)$. Tout d'abord, on note $\imath : \tilde{\Sigma}^* \to \Sigma^*$ le morphisme défini par : $\forall a \in \Sigma$, $\imath(a) = \imath(\bar{a}) = a$. On a aussi le diagramme commutatif suivant :

$$\begin{tikzcd}
\tilde{\Sigma}^* \arrow[r, "\pi_{\mathbb{M}}"] \arrow[d] \arrow[dr, "\pi_{\mathbb{G}}"] & \mathbb{M}(\tilde{\Sigma}, \tilde{D}) \arrow[d, "\varphi"] \\
F(\Sigma) \arrow[r] & \mathbb{G}(\tilde{\Sigma}, \tilde{D})
\end{tikzcd}$$

2.1.2 Représentants particuliers des traces

Chaque trace admet plusieurs représentations, on peut en extraire certaines qui sont particulières.Un ensemble de *formes normales* de traces est un sous-ensemble de Σ^* (ou $\tilde{\Sigma}^*$) tel que toute trace a un représentant et un seul dans cet ensemble.

Forme normale de Cartier-Foata

Une trace est uniquement représentée par la suite CF-admissible des cliques qui la compose. Prenons un ordre total sur Σ et associons à chaque clique le mot sur Σ obtenu en ordonnant les lettres suivant cet ordre. On obtient la forme normale de Cartier-Foata en mettant bout à bout les cliques de la suite CF-admissible.

Proposition 2.10 *L'ensemble des représentants en forme normale de Cartier-Foata d'un monoïde de traces (resp. d'un groupe de traces) est un langage rationnel de Σ^* (resp. de $\tilde{\Sigma}^*$).*

Démonstration. À partir du graphe des cliques on peut construire un automate reconnaissant le langage des formes normales de Cartier-Foata. On explicitera cet automate par la suite. □

Forme normale lexicographique

Dans le monoïde de traces, les mots w de Σ^* représentant la même trace ont tous la même longueur, et donc sont en nombre fini. Définissons un ordre total sur l'alphabet Σ. Alors on peut ordonner de façon totale Σ^* à l'aide de l'ordre lexicographique noté $\leq_{lex}$. Le représentant en forme normale lexicographique est le plus petit élément pour cet ordre.

Pour le cas du groupe de traces, il y a un nombre infini de mots qui représentent la même trace : si w représente une trace t, alors les mots de la forme $a^n\bar{a}^n w$ représentent la même trace, et si a est la plus petite lettre, c'est une suite décroissante dans l'ordre lexicographique. On va donc prendre un autre ordre sur $\tilde{\Sigma}^*$:

$$u \leq v \text{ si } \begin{cases} |u| < |v| \text{ ou} \\ |u| = |v| \text{ et } u \leq_{lex} v \end{cases} .$$

Ainsi, le représentant en forme normale lexicographique est le plus petit dans l'ordre lexicographique parmi les représentants les plus courts. Dans la suite, on prendra pour l'ordre sur $\Sigma \cup \bar{\Sigma}$ l'ordre induit par celui du monoïde : $a < \bar{a}$ et si $a < b$, alors $a < \bar{a} < b < \bar{b}$.

On a la proposition suivante (voir [36]) :

Proposition 2.11 *Un mot w est la forme normale lexicographique d'une trace de $\mathbb{M}(\Sigma, D)$ si et seulement si pour chaque facteur de w, aub, tel que $a, b \in \Sigma$, si $u \in \Sigma^*$, $(au, b) \in I$ alors $a < b$. On note $(u, a) \in I$ si $(b, a) \in I$, $\forall b \in \mathrm{alph}(u)$.*

On a aussi le même résultat pour les groupes :

Proposition 2.12 *Un mot w est la forme normale lexicographique d'une trace de $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ si et seulement si pour chaque facteur de w, aub tel que $a, b \in \Sigma$, $u \in \Sigma^*$, si $(au, b) \in I$ alors $a < b$.*

Et le corollaire :

Corollaire 2.13 *L'ensemble des formes normales lexicographiques d'un monoïde de traces ou d'un groupe de traces est un langage rationnel.*

Exemple 2.14 Soit l'ordre $a < b < c < d$. Le représentant en forme normale lexicographique des mots des exemples précédents sont :
- pour $w = acdbabdcb$, le représentant est : $acbabddcb$,
- pour $cd\bar{c}acb\bar{a}d$, le représentant est : $adab\bar{a}d$.

2.1.3 Représentation des traces par tas de pièces

On peut représenter les éléments du monoïde de traces et du groupe de traces par des tas de pièces. Cette représentation permet d'avoir une idée de la structure du mot par une représentation graphique, et donc intuitive. Soit $\mathbb{M}$ le monoïde $\mathbb{M}(\Sigma, D)$. Soit $\mathcal{C}ol$ un ensemble fini de colonnes. On modélise chaque lettre a par une pièce rectangulaire $p(a)$, de hauteur 1, et de base $col(a) \subset \mathcal{C}ol$. Les colonnes occupées par les pièces doivent satisfaire les conditions suivantes :

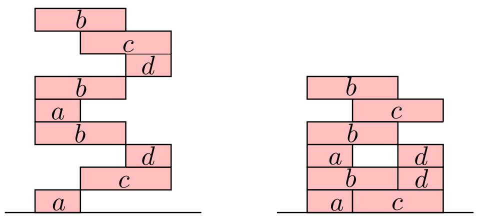
1. Si aIb, alors $col(a) \cap col(b) = \emptyset$,
2. Si aDb, alors $col(a) \cap col(b) \neq \emptyset$.

Soit $w = a_1 a_2 \ldots a_n$ un mot de Σ^*. On construit le tas de pièce correspondant à w en "faisant tomber" $p(a_1)$, puis $p(a_2)$ jusqu'à $p(a_n)$. Chaque tranche du tas ainsi construit représente une clique de la représentation CF de la trace représentée par le mot. De plus, on visualise aisément la hauteur d'un mot : c'est la hauteur du tas de pièces.

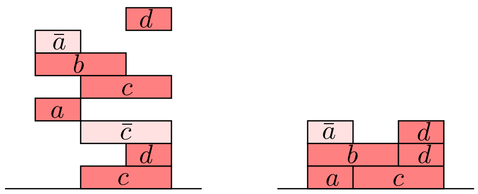
Si deux mots génèrent le même tas de pièces, alors ces mots sont équivalents : on a une bijection entre les traces du monoïde et les tas de pièces. De plus, on peut définir formellement le produit de deux tas de pièces, et donc le monoïde des tas de pièces. Viennot [91] a démontré l'isomorphisme entre le monoïde des tas de pièces et le monoïde de traces.

On peut représenter les traces du groupe par un tas de pièces colorées : a et $\bar{a}$ sont des pièces identiques (elles occupent les mêmes colonnes, mais sont de couleurs différentes : par exemple les pièces de Σ sont rouges et celles de $\bar{\Sigma}$ sont bleues). Les pièces tombent sur le tas de manière identique à celle du monoïde, mais si une pièce tombe sur la même pièce de couleur différente, alors ces deux pièces disparaissent. Ainsi, la hauteur du tas peut diminuer, et même, le tas peut disparaître.

Exemple 2.15 On reprend les deux exemples précédents. Voici le tas de pièces obtenu pour le monoïde :



et pour le groupe. Pour plus de lisibilité, les lettres barrées sont représentées par des pièces plus claires.



2.2 Propriétés combinatoires

On note $K[A]$ l'ensemble des polynômes sur A à coefficients dans K, $K[[A]]$ l'ensemble des fractions rationnelles sur A à coefficients dans K et $K\langle\langle A\rangle\rangle$ l'ensemble des séries sur A à coefficients dans K.

2.2.1 Séries génératrices des traces

L'un des problèmes naturels liés aux monoïdes de traces et aux groupes de traces est de compter les traces. On veut dénombrer les éléments qui ont la même longueur ou ceux qui ont la même hauteur, par exemple. Pour répondre à ces questions, on introduit les séries génératrices. Définissons donc les séries respectivement sur $\mathbb{N}\langle\langle x,y\rangle\rangle$, $\mathbb{N}\langle\langle y\rangle\rangle$ et $\mathbb{N}\langle\langle x\rangle\rangle$:

$$\begin{aligned} F(x,y) &= \sum_{t\in S} x^{h(t)}y^{|t|} \\ L(y) &= \sum_{t\in S} y^{|t|} \text{ (Série génératrice des longueurs)} \\ H(x) &= \sum_{t\in S} x^{h(t)} \text{ (Série génératrice des hauteurs)} \end{aligned}$$

où S peut représenter un monoïde de traces ou un groupe de traces, selon le contexte.

On a bien sûr $L(y) = F(1,y)$ et $H(x) = F(x,1)$. Les résultats suivants ont déjà été établis dans le cas du monoïde de traces [36], et la preuve s'adapte dans le cas des groupes :

Proposition 2.16 *Soit $\mathbb{G}$ un groupe de trace, et $\mathfrak{C}$ l'ensemble des cliques de I. Soit la matrice $A(x,y) \in \mathbb{N}[x,y]^{\mathfrak{C}\times\mathfrak{C}}$ définie par $A(x,y)_{i,j} = xy^{|i|}$ si (i,j) est CF-admissible, et 0 sinon. Soit le vecteur $u = (1,\ldots,1) \in \mathbb{N}[x,y]^{1\times\mathfrak{C}}$ et le vecteur $v \in \mathbb{N}[x,y]^{\mathfrak{C}\times 1}$ défini par $v(x,y)_i = xy^{|i|}$. Alors la série génératrice de la longueur et de la hauteur est donnée par :*

$$F(x,y) = 1 + \sum_{n\in\mathbb{N}} uA(x,y)^n v(x,y) = 1 + u(I - A(x,y))^{-1}v(x,y),$$

où I est la matrice identité.

Démonstration. Une trace est associée à son unique décomposition CF. On associe à un chemin p dans Γ, le graphe des cliques, la suite de nœuds qu'il parcourt $(c_1,\ldots,c_k)$. Par construction, la décomposition CF de $t = c_1\ldots c_k$ est $(c_1,\ldots,c_k)$. Donc les décompositions CF sont en bijection avec les chemins dans Γ. La contribution de la trace t à la série F est $x^{h(t)}y^{|t|}$. Le poids du chemin p dans l'automate de représentation $(u, A(x,y), v(x,y))$ est :

$$u_{c_1}\left(\prod_{i=1}^{k-1} A(x,y)_{c_ic_{i+1}}\right)v(x,y)_{c_k} = \left(\prod_{i=1}^{k-1} xy^{|c_i|}\right)xy^{|c_k|} = x^{h(t)}y^{|t|}.$$

$\square$

Et cette propriété a pour conséquence immédiate :

Corollaire 2.17 *$F(x,y)$, $L(y)$ et $H(x)$, sont des séries rationnelles, et*

$$\begin{aligned} L(y) &= 1 + u(I - A(1,y))^{-1}v(1,y), \\ H(x) &= 1 + u(I - A(x,1))^{-1}v(x,1). \end{aligned}$$

Partition équitable des cliques : La dimension de la matrice A est égale au nombre de cliques de I, qui devient vite très grand. Pour réduire la taille de la matrice, on partitionne l'ensemble des cliques en une partition équitable.

Définition 2.18 *Soit $\pi = \{\mathfrak{C}_1, \ldots, \mathfrak{C}_k\}$ une partition de $\mathfrak{C}$. On dit que π est une partition équitable si pour tout $i, j \in \{1, \ldots, k\}$, tout élément de $\mathfrak{C}_i$ a dans Γ le même nombre $n_{i,j}$ de successeurs dans le graphe des cliques et dans $\mathfrak{C}_j$.*

Soit π une partition équitable, telle que tous les éléments de chaque élément $\mathfrak{C}_i$ de la partition soient de la même longueur ℓ_i. On a alors :

$$F = 1 + \sum_{n\in\mathbb{N}} u_\pi A_\pi(x,y)^n v_\pi(x,y) = 1 + u_\pi(I - A_\pi(x,y))^{-1} v_\pi(x,y),$$

où $u_\pi = (|\mathfrak{C}_i|)_{i\in\mathbb{N}_k}$, $A_\pi(x,y) = (n_{i,j} x y^{|\ell_i|})_{i,j\in\mathbb{N}_k}$ et $v_\pi(x,y) = (xy^{|\ell_i|}))_{i\in\mathbb{N}_k}$.

Exemple 2.19 Toujours avec le même exemple, on a la partition équitable

$$\pi = \{\{\{a\},\{d\}\},\{\{b\},\{c\}\},\{\{a,c\},\{b,d\}\},\{\{a,d\}\}\}.$$

On obtient la matrice $A(1,1)$ (on ne fait pas apparaître x et y) et les vecteurs u_π et v_π :

$$u_\pi = \begin{pmatrix} 2 & 2 & 2 & 1 \end{pmatrix} \qquad A_\pi(1,1) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 1 \end{pmatrix} \qquad v_\pi = \begin{pmatrix} xy \\ xy \\ xy^2 \\ xy^2 \end{pmatrix}$$

Pour le groupe de traces, on peut toujours trouver une partition équitable non triviale : il suffit de regrouper les cliques qui ont la même image par $\imath$. S'il existe $u \in \mathfrak{C}_i$ et $v \in \mathfrak{C}_j$ tel que (u,v) est CF-admissible, alors $A_{\pi\ i,j} = A^k$, où k est le nombre de lettre de $\imath(u)$ n'apparaissant pas dans $\imath(v)$ (et k ne dépend pas de u et v). Par le même argument, on peut déduire une partition équitable pour le groupe de traces à partie de celle trouvée pour le monoïde de traces.

$$u_\pi = \begin{pmatrix} 4 & 4 & 8 & 4 \end{pmatrix} \qquad A_\pi(1,1) = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 2 & 3 & 4 & 0 \\ 3 & 3 & 5 & 2 \\ 2 & 4 & 4 & 1 \end{pmatrix} \qquad v_\pi = \begin{pmatrix} xy \\ xy \\ xy^2 \\ xy^2 \end{pmatrix}$$

2.2.2 Série génératrice des longueurs et fonction de Möbius

Pour obtenir le corollaire 2.17, on utilise la représentation matricielle d'un automate reconnaissant les suites de cliques CF-admissibles. Mais il existe un procédé beaucoup plus élégant pour calculer la série génératrice L qui, de plus, permet d'obtenir la fraction rationnelle explicitement : L est en fait l'inverse de la fonction de Möbius.

Définition 2.20 *Soit $\mathbb{M}(\Sigma, D)$ un monoïde de trace, (Σ, I) son graphe d'indépendance et $\mathfrak{C}$ l'ensemble des cliques de (alf, I). Alors le polynôme de Möbius $\mu_\mathbb{M} \in \mathbb{Z}[[X]]$ est défini par :*

$$\mu_\mathbb{M}(x) = 1 + \sum_{c\in\mathfrak{C}} (-1)^{|c|} x^{|c|}$$

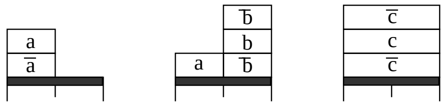


FIG. 2.2 – Exemples de traces alternées.

Théorème 2.21 *La série génératrice des longueurs des traces dans un monoïde de trace $\mathbb{M}$ est donnée par :*

$$L_{\mathbb{M}} = \frac{1}{\mu_{\mathbb{M}}}$$

Ce résultat est dû à Cartier et Foata [28] et est un résultat fondamental pour l'étude des monoïdes de traces. Ce résultat est aussi à l'origine de certaines études combinatoires basées sur les tas de pièces. On démontre maintenant un analogue dans le cas du groupe de traces.

Étant donnés deux mots u et v de $\tilde{\Sigma}^*$, tels que $\pi_{\mathbb{G}}(u) = \pi_{\mathbb{G}}(v)$ et $|u| = |v| = |\pi_{\mathbb{G}}(u)|$, on a $\pi_{\mathbb{M}}(u) = \pi_{\mathbb{M}}(v)$. Mais alors, une trace de $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$ admet un unique représentant de longueur minimale dans $\mathbb{M}(\tilde{\Sigma}, \tilde{D})$. On note

$$\phi : \mathbb{G}(\tilde{\Sigma}, \tilde{D}) \to \mathbb{M}(\tilde{\Sigma}, \tilde{D})$$

la fonction correspondante. En terme de tas de pièces, si t est un tas coloré, alors $\phi(t)$ est le tas obtenu en oubliant les couleurs.

On définit l'ensemble des *traces alternées* comme un sous-ensemble $\mathfrak{D}$ de $\mathbb{M}(\tilde{\Sigma}, \tilde{D})$ donné par l'expression régulière

$$\mathfrak{D} = \sum_{c \in \mathfrak{C}} [\prod_{x \in \mathrm{alph}(c)} x(\bar{x}x)^*(1 + \bar{x})]. \tag{2.1}$$

La figure 2.2 montre des exemples de traces alternées pour $\Sigma = \{a, b, c\}$ et $ab = ba$.

Nous allons maintenant énoncer le résultat principal de ce chapitre.

Théorème 2.22 *Dans $\mathbb{Z}\langle\langle \mathbb{M}(\tilde{\Sigma}, \tilde{D}) \rangle\rangle$, on a l'égalité suivante :*

$$(\sum_{t \in \phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D}))} t).(\sum_{d \in \mathfrak{D}} (-1)^{|d|} d) = 1. \tag{2.2}$$

Démonstration. Pour montrer ce résultat, on utilise un argument bijectif similaire à celui utilisé dans [91] (remarque 5.2). Pour une trace $t \in \mathbb{M}(\tilde{\Sigma}, \tilde{D})$, définissons

$$\mathrm{Top}(t) = \{a \in \mathrm{alph}(t) \mid \exists u,\ t = ua\}.$$

C'est l'ensemble des pièces entièrement visibles quand le tas est vu de dessus. Étant donnés une trace t de $\mathbb{M}(\tilde{\Sigma}, \tilde{D})$ et $h \in \mathfrak{D}$, définissons $M(t, h) \subseteq \Sigma$ par $M(t, h) = \imath(\mathrm{Top}(h) \cup \{a \in \mathrm{Top}(t) \mid \forall b \in \mathrm{Top}(h),\ a\tilde{I}b\})$. Observons que $M(t, h) = \emptyset$ si et seulement si $(t, h) = (\varepsilon, \varepsilon)$. Définissons maintenant une application

$$\psi : \phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D})) \times \mathfrak{D} \to \phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D})) \times \mathfrak{D}. \tag{2.3}$$

On a $\psi(\varepsilon, \varepsilon) = (\varepsilon, \varepsilon)$. Pour $(t, h) \in \phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D})) \times \mathfrak{D}$ et $(t, h) \neq (\varepsilon, \varepsilon)$, on distingue quatre cas. Soit a la lettre minimale (on suppose que $\tilde{\Sigma}$ est équipé d'un ordre total).

1. Si on a $t = t_1 a_1$ avec $a_1 \in \{a, \bar{a}\}$ et $a, \bar{a} \notin \mathrm{alph}(h)$, alors $\psi(t, h) = (t_1, a_1 h)$.

2. Si on a $h = a_1h_1$ avec $a_1 \in \{a, \bar{a}\}$ et $a, \bar{a} \notin \mathrm{Top}(t)$, alors $\psi(t, h) = (ta_1, h_1)$.
3. Si on a $t = t_1a_1$ avec $a_1 \in \{a, \bar{a}\}$ et $h = \bar{a}_1h_1$, alors $\psi(t, h) = (t_1, a_1h)$.
4. Si on a $t = t_1a_1$ avec $a_1 \in \{a, \bar{a}\}$ et $h = a_1h_1$, alors $\psi(t, h) = (ta_1, h_1)$.

Les quatre cas à considérer pour la fonction ψ sont représentés sur la figure 2.3 avec l'ordre $a < \bar{a} < b < \bar{b} < c < \bar{c}$.

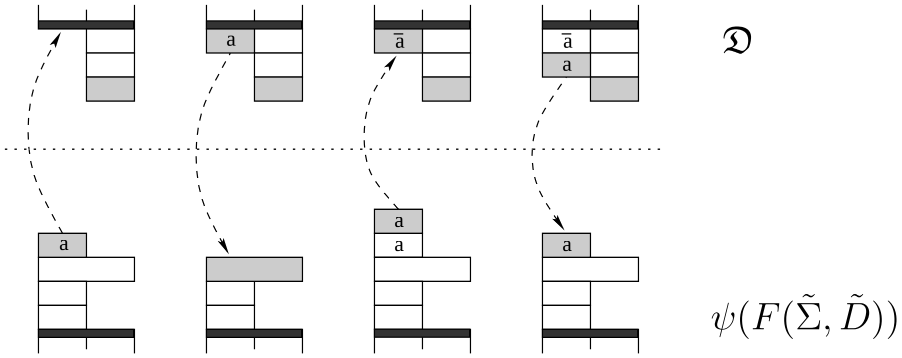


FIG. 2.3 – L'involution Ψ et les principaux cas à considérer.

On remarque que les configurations de type aa sont autorisées dans $\phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D}))$ et pas dans $\mathfrak{D}$ et qu'à l'inverse, les configurations de type $a\bar{a}$ sont autorisées dans $\mathfrak{D}$ et pas dans $\phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D}))$. De plus, la lettre minimale de $M(t, h)$ est la même que celle de $M\psi(t, h)$. Par conséquent, on a $\psi^2 = Id$. De plus, pas construction, le seul point fixe de ψ est $(\varepsilon, \varepsilon)$. En outre, si $(t, h) \neq (\varepsilon, \varepsilon)$ et si $\psi(t, h) = (t', h')$, on a $th = t'h'$ et $|h'| = |h| \pm 1$. Ainsi,

$$(-1)^{|h|}th = -(-1)|h'|t'h'.$$

Soit $S = \phi(\mathbb{G}(\tilde{\Sigma}, \tilde{D})) \times \mathfrak{D}$. En décomposant la somme sur les orbites de ψ, on obtient

$$\Big(\sum_{t\in\phi(\mathbb{G}(\tilde{\Sigma},\tilde{D}))} t\Big).\Big(\sum_{d\in\mathfrak{D}}(-1)^{|d|}d\Big) = 1 + \sum_{(t,h)\in S\setminus\{(\varepsilon,\varepsilon)\}} (-1)^{|h|}th = 1. \quad (2.4)$$

□

Corollaire 2.23 *La série génératrice des éléments du groupe $\mathbb{G}(\tilde{\Sigma}, \mathcal{D})$ est donnée par*

$$L_{\mathbb{G}}(x) = \frac{1}{\mu_{\tilde{\mathbb{M}}}(\frac{x}{1+x})}, \quad (2.5)$$

où $\mu_{\tilde{\mathbb{M}}}$ est le polynôme de Möbius associé au monoïde $\mathbb{M}(\tilde{\Sigma}, \mathcal{D})$.

Démonstration. Remplaçons dans la formule obtenue dans le théorème précédent toutes les variables par une unique variable x. Alors l'inverse de la fonction de Möbius devient $\sum_{d\in\mathfrak{D}}(-1)^{|d|}x^{|d|}d$. On a donc, d'après la définition de $\mathfrak{D}$,

$$\begin{aligned}
\sum_{d\in\mathfrak{D}}(-1)^{|d|}x^{|d|}d &= \sum_{c\in\mathfrak{C}}((-x\sum_{m\in\mathbb{N}} x^{2m}(1-x))^{|c|} \\
&= \sum_{c\in\mathfrak{C}}(-\frac{x-x^2}{1-x^2})^{|c|} \\
&= \sum_{c\in\mathfrak{C}}(-1)^{|c|}(\frac{x}{1+x})^{|c|} = \mu_{\tilde{\mathbb{M}}}(\frac{x}{1+x}).
\end{aligned}$$

□

Exemple 2.24 Le polynôme de Möbius associé au monoïde de l'exemple 1 est : $\mu_{\mathbb{M}}(y) = 1 - 4x + 3x^2 = (1-x)(1-3x)$.

Le polynôme de Möbius associé au monoïde dont le graphe de dépendance est celui de l'exemple 2 est : $\mu_{\tilde{\mathbb{M}}}(y) = 1 - 8x + 12x^2 = (1-2x)(1-6x)$.

La fonction génératrice des longueurs des éléments du groupe de l'exemple 2 est donc :

$$L_{\mathbb{G}} = \frac{1}{(1-2\frac{x}{1+x})(1-6\frac{x}{1+x})} = \frac{(1+x)^2}{(1-x)(1-5x)}$$

Lien entre les racines des séries génératrices des longueurs de $\mathbb{G}$ et de $\mathbb{M}$. On a $L_{\tilde{\mathbb{M}}}(x) = L_{\mathbb{M}}(2x)$. En effet, à chaque lettre a dans $\mathbb{M}$, on peut associer $(a+\bar{a})$. D'après la section précédente, on a donc :

$$L_{\mathbb{G}}(x) = L_{\mathbb{M}}(\frac{2x}{1+x}).$$

On sait que $\frac{1}{L_{\mathbb{M}}(x)}$ est μ, le polynôme de Möbius associé à $\mathbb{M}$. Décomposons-le en éléments simples :

$$\mu = \alpha \prod_{i=1}^{n}(1 - \rho_i^{-1}x)$$

Soit $R = \frac{1}{L_{\mathbb{G}}(x)}$. On a alors :

$$\begin{aligned} R &= \alpha \prod_{i=1}^{n}(1 - \rho_i^{-1}\frac{2x}{1+x}) \\ &= \alpha \frac{\prod_{i=1}^{n}(1-(2\rho_i^{-1}-1)x)}{(1+x)^n} \end{aligned}$$

Par conséquent, on a la proposition suivante :

Proposition 2.25 *Soit $\mathbb{M} = \mathbb{M}(\Sigma, D)$ un monoïde de traces, et $\mathbb{G} = \mathbb{G}(\tilde{\Sigma}, \tilde{D})$ le groupe de traces construit à partir de $\mathbb{M}$. On note $L_{\mathbb{M}}$ et $L_{\mathbb{G}}$ les séries génératrices des longueurs des traces dans $\mathbb{M}$ et dans $\mathbb{G}$. Si les pôles de $L_{\mathbb{M}}$ sont $\rho_1 < \cdots < \rho_k$ avec les multiplicités respectives $\alpha_1, \ldots, \alpha_k$, alors les pôles de $L_{\mathbb{G}}$ sont $\rho_1/(2-\rho_1) < \cdots < \rho_k/(2-\rho_k)$ avec les multiplicités $\alpha_1, \ldots, \alpha_k$.*

En particulier, si ρ est le pôle dominant de $L_{\mathbb{M}}$, alors $\rho/(2-\rho)$ est le pôle dominant de $L_{\mathbb{G}}$, avec la même multiplicité.

Un tableau en fin de chapitre donne les formules des fonctions génératrices des hauteurs pour les monoïdes et les groupes de traces définis sur un petit alphabet.

Interprétation en variables non commutatives.

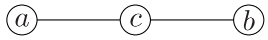
Le langage des représentants en forme normale lexicographique d'un groupe ou d'un monoïde de traces est rationnel. De plus, on peut définir un ensemble de mots interdits de ces langages : ce sont des mots qui ne peuvent apparaître comme facteurs dans les représentants (c'est donc un idéal à gauche et à droite). Il existe un plus petit ensemble qui génère l'ensemble des mots interdits (c'est la *base de l'idéal*). Dans tout ce paragraphe, on note toujours $\mathcal{L}_S$ le langage des formes normales lexicographique correspondant à la structure S (groupe ou monoïde de traces), et $Forb_S$ la base du langage des mots interdits de $\mathcal{L}_S$.

On va maintenant définir la notion de chaîne recouvrante pour $Forb_S$.

Définition 2.26 *Une* chaîne recouvrante *pour $Forb_S$ est une suite finie $u_1, u_2 \ldots, u_n \in \mathcal{L}_S$ telle que $u_1 \in \Sigma$ et pour tout i de 1 à $n-1$, $u_i u_{i+1} \in \Sigma^* Forb_S \backslash \Sigma^* Forb_S \Sigma^+$.*

Les chaînes recouvrantes de longueur 1 sont les lettres, qui ne sont pas sans $Forb_S$, et les chaînes recouvrantes de longueur 2 sont les mots de $Forb_S$ de longueur supérieure ou égale à 2.

Exemple 2.27 Pour le monoïde de graphe de dépendance,



avec l'ordre $a < b < c$, la base des mots interdits est $\{ba\}$, les chaînes recouvrantes sont : a,b,c,ba. Pour le groupe $\mathbb{G}$ correspondant à ce monoïde, avec l'ordre $a < \bar{a} < b < \bar{b} < c < \bar{c}$, les mots interdits ont pour base $\{ba, \bar{b}a, b\bar{a}, \bar{b}\bar{a}, a\bar{a}, \bar{a}a, b\bar{b}, \bar{b}b, c\bar{c}, \bar{c}c\}$. Les chaînes recouvrantes sont en nombre infini, et de la forme $(b\bar{b})^n(\bar{a}a)^m\bar{a}$, par exemple.

On note $\mathcal{F}orb_S$ l'ensemble des chaînes recouvrantes. Le théorème suivant est une conséquence directe de [36, theorem 4.4.2], valable pour tout langage que l'on peut définir par une base de mots interdits.

Théorème 2.28 *Soient les séries sur le monoïde libre Σ^**

$$\begin{aligned} \mu_{\mathcal{F}_S} &= \sum_{(w_1,\ldots,w_n)\in\mathcal{F}orb_S} (-1)^n (w_1 \ldots w_n) \text{ et} \\ \zeta_{\mathcal{F}_S} &= \sum_{w\in\mathcal{L}_S} w. \end{aligned}$$

Ces séries formelles $\mu_{\mathcal{F}_S}$ et $\zeta_{\mathcal{F}_S}$ de $\mathbb{Z}\langle\langle\tilde{\Sigma}\rangle\rangle$ sont inverses l'une de l'autre.

Dans le cas du monoïde, on retrouve la fonction de Möbius en variables non commutatives. On obtient un résultat similaire pour les groupes.

Exemple 2.29 Reprenons l'exemple précédent :

Pour le monoïde, on obtient,

$$\zeta_{\mathcal{F}_{\mathbb{M}}} = \frac{1}{1 - a - b - c + ba}$$

Pour le groupe, le résultat est plus compliqué, mais assez facile à obtenir. En effet, il suffit dans ce cas particulier de remplacer chaque lettre $a \in \Sigma$ par $(a(\bar{a}a)^*(1-\bar{a}) + \bar{a}(a\bar{a})^*(1-a))$.

Dans certains cas, on peut trouver l'ensemble des chaînes recouvrantes d'un groupe de traces à l'aide de celles du monoïde de traces associé.

Cas 1 : la relation $I\cap <$ est transitive sur Σ. Dans ce cas, la base de l'idéal des mots interdits est uniquement formé de mots de deux lettres et donc, dans le monoïde les chaînes recouvrantes, sont en nombre fini. Elles correspondent aux cliques du graphe d'indépendance, les lettres classées par ordre décroissant. Dans chaque chaîne recouvrante, une lettre n'apparaît qu'une seule fois. Pour obtenir la série génératrice du groupe de traces, il suffit de remplacer chaque lettre a par $l(a) = (a(\bar{a}a)^*(1-\bar{a}) + \bar{a}(a\bar{a})^*(1-a))$. On vérifie bien que les expressions rationnelles utilisées sont non ambiguës.

Cas 2 : la relation $I\cap <$ n'est pas transitive sur Σ. Il y a alors des mots interdits de longueur 2, mais aussi des familles de la forme $cI_{b,c}^*b$ où $I_{b,c}$ est l'ensemble des lettres qui commutent avec b, qui sont plus petites que b, et qui dépendent de c. Les chaînes recouvrantes deviennent alors beaucoup plus complexes, et n'ont pas été étudiées de manière aussi approfondie que pour les ordres transitifs.

2.2.3 Sur la série génératrice des hauteurs

Il y a deux manières de calculer la série génératrice des longueurs : on passe soit par l'interprétation matricielle d'un automate $((u, A(1,y),\ v(1,y)))$, soit par une méthode directe qui donne une interprétation combinatoire à la série. C'est la fonction de Möbius.

La première méthode permet aussi de calculer la série génératrice de la hauteur $((u, A(x,1),$ $v(x,1)))$, mais aucune interprétation combinatoire, qui donnerait un calcul direct de H n'a jusqu'à présent été trouvée.

Dans ce paragraphe, nous allons donner une interprétation combinatoire de la série génératrice des hauteurs. Mais cette interprétation ne donne pas directement la formule pour H. En fait, la série H est utilisée pour calculer une autre série, dont on peut aussi donner une formule explicite.

On considère les cliques $\mathfrak{C}$ associées au monoïde (ou au groupe, le raisonnement étant le même). Pour calculer H, on interprète une trace comme un tas de pièces, dont les tranches définissent un empilement de cliques CF-admissibles. Les empilements que l'on veut dénombrer sont donc les empilements CF-admissibles, vus comme des empilements de cliques. On considère tout d'abord un empilement quelconque de cliques. On note $D_{\mathbb{M}}$ la série génératrice des empilements quelconques de cliques du monoïde de traces $\mathbb{M}$ (et $D_{\mathbb{G}}$ celle correspondant au groupe de traces $\mathbb{G}$), en gardant les cliques comme variables commutatives. On a : $D_{\mathbb{M}} = 1/(1-\mathfrak{C})$. On essaie maintenant de trouver une bijection entre les éléments d'un empilement quelconque de cliques et ceux d'un empilement CF-admissible.

On définit pour chaque clique c l'ensemble $\mathfrak{C}_c = \{u \in \mathfrak{C} | (c,u) \notin CF\}$ des cliques qui ne peuvent être empilées sur la clique c dans le monoïde. On transforme un empilement quelconque en un empilement CF-admissible comme suit : on garde la première clique, puis la première clique rencontrée qui est CF-admissible avec la première, et ainsi de suite, comme le montre la figure 2.4.

Plus formellement, la transformation entre les empilements quelconques de cliques et les empilements CF-admissibles est la suivante :

$$f : \begin{array}{ccc} \mathfrak{C}^* & \to & \mathfrak{C}^* \\ (c_1 \dots c_k) & \mapsto & (c_{i_1} \dots c_{i_\ell}) \end{array}$$

où $i_1 = 1$ et $c_{i_j} = \min_{s > i_{j-1}} \{c_s | (c_{i_{j-1}}, c_s) \in CF\}$. Si c_{i_j} n'est pas défini, on s'arrête à $c_{i_{j-1}}$.

La fonction f est surjective sur l'ensemble des formes normales CF, et on a

$$f^{-1}(c_1 \dots c_j) = c_1 \mathfrak{C}_{c_1}^* \dots c_j \mathfrak{C} c_j{}^*.$$

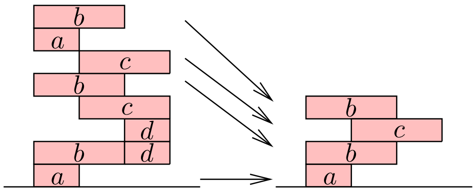


FIG. 2.4 – Transformation d'un empilement quelconque de cliques en un empilement CF-admissible.

Il en découle la proposition suivante :

Proposition 2.30 *Soit* $\mathbb{M} = \mathbb{M}(\Sigma, D)$ *un monoïde de traces,* $\mathfrak{C} = \{c_1, \dots c_n\}$ *l'ensemble des cliques de son graphe d'indépendance. On a alors l'égalité suivante dans* $\mathbb{N}\langle\langle \mathfrak{C}^* \rangle\rangle$*.*

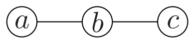
$$D_{\mathbb{M}}(c_1, c_2, \dots, c_n) = H_{\mathbb{M}}\left(\frac{c_1}{1 - \mathfrak{C}_{c_1}}, \frac{c_2}{1 - \mathfrak{C}_{c_2}}, \dots, \frac{c_n}{1 - \mathfrak{C}_{c_n}}\right)$$

Pour trouver $H_{\mathbb{M}}$ il faut donc résoudre le système

$$\left\{\begin{array}{ccc} c'_1 & = & \frac{c_1}{1-\mathfrak{C}_{c_1}} \\ c'_2 & = & \frac{c_2}{1-\mathfrak{C}_{c_2}} \\ \dots & & \dots \\ c'_n & = & \frac{c_1}{1-\mathfrak{C}_{c_n}} \end{array}\right. .$$

On obtiendra ensuite $H_{\mathbb{M}}$ en injectant les expressions trouvées pour les c_i dans $D_{\mathbb{M}}$.

Exemple 2.31 On calcule la série génératrice des hauteurs associée au monoïde de trace dont le graphe de dépendance est :



On a le système d'équations :

$$\left\{\begin{array}{ccc} a' & = & \frac{a}{1-c-(ac)} \\ b' & = & b \\ c' & = & \frac{c}{1-a-(ac)} \\ (ac)' & = & (ac) \end{array}\right.$$

En résolvant ce système, on trouve :

$$\left\{\begin{array}{ccc} a & = & \frac{a'(1-(ac)')(1-c')}{1-a'c'} \\ b & = & b' \\ c & = & \frac{c'(1-(ac)')(1-a')}{1-a'c'} \\ (ac) & = & (ac)' \end{array}\right.$$

Comme

$$D = \frac{1}{1 - a - b - c - (ac)},$$

on obtient :

$$H(x) = \frac{1}{1 - 2\frac{x(1-x)^2}{(1-x^2)} - 2x} = \frac{1+x}{1-3x}$$

2.3 Propriétés asymptotiques des séries génératrices

Notations : Soit $S \in \mathbb{N}[[x]]$ une série rationnelle. On la note $S = \sum (S|n)x^n$, sa singularité dominante est notée ρ_S, de multiplicité k_S.

Si S est une fraction rationnelle, on connaît le comportement asymptotique des coefficients $(S|n)$ ([44]) :

$$(S|n) \sim \frac{\rho_S}{(k_S-1)!}[S(y)(\rho_S - y)^{k_S}]_{|y=\rho_S} n^{k_L-1}\rho_S^{-n}$$

2.3.1 Étude des singularités des séries $L_{\mathbb{G}}$ et $H_{\mathbb{G}}$

Nous avons calculé les expressions des séries génératrices des longueurs et des hauteurs. À partir de celles-ci, on peut trouver le comportement asymptotique des coefficients des séries. On peut alors approximer le nombre de traces de longueur n, ou de hauteur h. Les techniques utilisées consistent à trouver le pôle dominant et la multiplicité de la série génératrice.

Donnons tout d'abord un premier lemme ayant rapport au graphe des cliques, pour le monoïde, puis pour le groupe :

Lemme 2.32 ([64]) *Si le graphe de dépendance (Σ, D) du monoïde de traces $\mathbb{M} = \mathbb{M}(\Sigma, D)$ est fortement connexe, alors le graphe des cliques correspondant est aussi fortement connexe.*

Lemme 2.33 *Si le graphe de dépendance de $(\tilde{\Sigma}, \tilde{D})$ du groupe de traces $\mathbb{G} = \mathbb{G}(\tilde{\Sigma}, \tilde{D})$ est fortement connexe , alors le graphe des cliques correspondant est aussi fortement connexe.*

Notation : $\bar{a}$ est l'inverse de a donc on prend la convention $\bar{\bar{a}} = a$ et on étend la notation aux mots : la trace représentée par $\bar{u}$ est l'inverse de celle représentée par u. En particulier, $a \in \mathrm{alph}(u) \Leftrightarrow \bar{a} \in \mathrm{alph}(\bar{u})$.
Démonstration. Il s'agit de montrer que pour toutes cliques u et v il existe un chemin dans Γ de u à v. Faisons-le par récurrence sur la taille des cliques : $|u| + |v|$. Si $|u| + |v| = 2$, alors $|u| = |v| = 1$ et comme il existe un chemin de u à v dans le graphe $(\tilde{\Sigma}, \tilde{D})$, il existe aussi un chemin de $\{u\}$ à $\{v\}$ dans le graphe des cliques. Si $|u| + |v| > 2$, si $|u| > 1$, il existe $a \in \mathrm{alph}(u)$ tel que (u, a) est admissible, puis, si nécessaire ($\bar{a} \in \mathrm{alph}(v)$), un chemin de a à une lettre qui n'appartient pas à $\mathrm{alph}(\bar{v})$, notée b.

Il existe un chemin de b à v : on considère le sous-graphe de $\tilde{D}$ dont les nœuds sont, pour chaque lettre c de Σ, soit c, soit $\bar{c}$, sachant que si $c \in \mathrm{alph}(vb)$, alors on choisit c. On garde toutes les arêtes qui joignent les nœuds sélectionnés. Ce graphe est par construction isomorphe à D, et donc fortement connexe. D'après le lemme précédent, le graphe des cliques restreint à ces nœuds est fortement connexe : il existe un chemin de b à v. □

Le lemme suivant permet de caractériser les structures commutatives. Nous le démontrons dans le cas du groupe (pour le monoïde, il déjà été démontré dans [64]) :

Lemme 2.34 *On a $\rho_{L_{\mathbb{G}}} = 1$ ou $\rho_{H_{\mathbb{G}}} = 1$ si et seulement si $\mathbb{G}$ est le groupe commutatif libre.*

Démonstration. D'après la formule de la croissance exponentielle, on a $\limsup_n (L_{\mathbb{G}}|n)^{1/n} = 1/\rho_{L_{\mathbb{G}}}$. On a donc : $\rho_{L_{\mathbb{G}}} \leq 1$ et de même $\rho_{H_{\mathbb{G}}} \leq 1$. S'il existe $a, b \in A$ tel que $a\tilde{D}b$, $a \neq b$, alors toutes les traces $t_1 \dots t_n$ avec $t_i \in \{a, b\}$ sont de longueur et de hauteur n. Donc $(L_{\mathbb{G}}|n) \geq 2^n$ et $(H_{\mathbb{G}}|n) \geq 2^n$. Donc on a $\rho_{L_{\mathbb{G}}} \leq 1/2$ et $\rho_{H_{\mathbb{G}}} \leq 1/2$.

Dans le cas du monoïde commutatif libre, on a le résultat grâce à la proposition 2.25, et au cas de l'alphabet à une lettre où l'on voit facilement que $\rho_{L_{\mathbb{G}}} = \rho_{H_{\mathbb{G}}} = 1$. En effet,

$$L_{\mathbb{G}} = H_{\mathbb{G}} = \frac{1+x}{1-x}$$

□

Proposition 2.35 *Soit (Σ, D) un graphe de dépendance fortement connexe et $\mathbb{G} = \mathbb{G}(\tilde{\Sigma}, \tilde{D})$ le groupe de traces correspondant. Alors $L_{\mathbb{G}}$ et $H_{\mathbb{G}}$ ont une unique singularité dominante qui est réelle, positive et de multiplicité 1.*

Démonstration. Pour le monoïde, le résultat ([64]) n'était pas immédiat, mais dans le cas du groupe, la preuve pour $L_{\mathbb{G}}$ en est une conséquence immédiate : $L_{\mathbb{G}}$ est de la forme $L_{\mathbb{M}}(\frac{2x}{1+x})$, où $L_{\mathbb{M}}$ est la série génératrice du monoïde associé, qui a un graphe de dépendance fortement connexe si celui du groupe l'est. Donc $L_{\mathbb{M}}$ a une unique singularité dominante, de multiplicité 1, et la relation entre les singularités de $L_{\mathbb{G}}$ et $L_{\mathbb{M}}$, donnée au paragraphe précédent montre que $L_{\mathbb{G}}$ a aussi une unique singularité dominante, de multiplicité 1 : $\rho_{L_{\mathbb{M}}}/(2-\rho_{L_{\mathbb{M}}})$.

La preuve pour $H_{\mathbb{G}}$ ainsi qu'une autre preuve pour L est donnée en fin de chapitre, avec d'autres preuves. □

La version de cette proposition pour les monoïdes est prouvée dans [64].

Proposition 2.36 *Soit (Σ, D) un graphe de dépendance non connexe et $\mathbb{G} = \mathbb{G}(\tilde{\Sigma}, \tilde{D})$ le groupe de traces correspondant. Soit $(\Sigma_s, D_s)_{s\in S}$ sa partition en composantes fortement connexes. Notons L_s et H_s les fonctions génératrices des longueurs et des hauteurs correspondantes.*

1. *La série $L_{\mathbb{G}}$ a une unique singularité dominante : $\rho_{L_{\mathbb{G}}} = \min_{s\in S}\rho_{L_s}$, dont la multiplicité est $\#\{s,\ \rho_{L_s} = \rho_{L_{\mathbb{G}}}\}$*
2. *La série $H_{\mathbb{G}}$ a une unique singularité dominante : $\rho_{H_{\mathbb{G}}} = \prod_{s\in S}\rho_{H_s}$, dont la multiplicité est $|\Sigma|$ si $\mathbb{G}$ est le groupe libre commutatif et $1 + \#\{s \in S,\ |\Sigma_s| = 1\}$ sinon.*

Démonstration. On a

$$L_{\mathbb{G}} = \prod_{s\in S} L_s = \prod_{s\in S} \frac{1}{\mu_s(\frac{x}{1+x})},$$

donc la fraction $L_{\mathbb{G}}$ s'annule uniquement en -1. De plus on sait que les pôles dominants des fractions L_s sont positifs. Donc il n'y a pas de simplification possible entre le numérateur et le dénominateur mettant en jeu les singularités dominantes des L_s. Ainsi, la singularité dominante de L et sa multiplicité sont bien celles énoncées dans la proposition.

Considérons maintenant la série génératrice des hauteurs, et prouvons le résultat par induction sur $|S|$. Si $|S| = 1$, alors le résultat est déjà démontré, à la proposition précédente. Si $|S| = m > 1$, $S = \{1, \ldots, m\}$, et supposons que le résultat est démontré pour $|S| = m - 1$ alors posons H_1 la série génératrice des hauteurs associée au graphe (Σ_1, D_1) et H_2 la série génératrice des hauteurs associée au graphe $(\Sigma\backslash\Sigma_1, D\backslash D_1)$. On a : $H_{\mathbb{G}} = \sum_{i,j}(H_1|i)(H_2|j)x^{\max(i,j)}$. Mais alors,

$$(H_{\mathbb{G}}|n) = (H_1|n)\sum_{i=0}^{n}(H_2|i) + (H_2|n)\sum_{i=0}^{n}(H_1|i) - (H_1|n)(H_2|n). \tag{2.6}$$

H_1 a une unique singularité dominante, de multiplicité 1, réelle positive, donc, $(H_1|n) = a_n\rho_{H_1}^{-n}$, avec $\lim_n a_n = a \in \mathbb{R}_+^*$. H_2 a une unique singularité dominante, réelle, positive d'après l'hypothèse de récurrence, et notons k sa multiplicité : $(H_2|n) = b_n n^{k-1}\rho_{H_2}^{-n}$, avec $\lim_n b_n = b \in \mathbb{R}_+^*$.

Si $\rho_{H_1} < 1$ et $\rho_{H_2} < 1$:

$$\begin{aligned}
(H_1|n)\sum_{i=0}^{n}(H_2|i) &= a_n\rho_{H_1}^{-n}(\sum_{i=0}^{n} b_i\rho_{H_2}^{-i}i^{k-1}) \\
&= a_n b\rho_{H_1}^{-n}\rho_{H_2}^{-n}(\sum_{i=0}^{n}(b_{n-i}/b)\rho_{H_2}^{i}(n-i)^{k-1}) \\
&\sim abn^{k-1}(1-\rho_{H_2})^{-1}(\rho_{H_1}\rho_{H_2})^{-n}
\end{aligned}$$

et

$$\begin{aligned}
(H_2|n)\sum_{i=0}^{n}(H_1|i) &= b_n n^{k-1}\rho_{H_2}^{-n}(\sum_{i=0}^{n} a_i\rho_{H_1}^{-i}) \\
&= b_n a n^{k-1}\rho_{H_2}^{-n}\rho_{H_1}^{-n}(\sum_{i=0}^{n}(a_{n-i}/a)\rho_{H_1}^{i}) \\
&\sim abn^{k-1}(1-\rho_{H_1})^{-1}(\rho_{H_2}\rho_{H_1})^{-n}
\end{aligned}$$

En revenant à l'équation 2.6, on obtient :

$$(H_{\mathbb{G}}|n) \sim abn^{k-1}((1-\rho_{H_1})^{-1} + (1-\rho_{H_2})^{-1} - 1)(\rho_{H_1}\rho_{H_2})^{-n}$$

On a donc bien $\rho_{H_{\mathbb{G}}} = \rho_{H_1}\rho_{H_2}$ et la multiplicité de $H_{\mathbb{G}}$ reste bien k : on a le même nombre de composantes à un élément que pour H_2.

Si maintenant $\rho_{H_1} = \rho_{H_2} = 1$, les composantes connexes du graphe D sont nécessairement des graphes à un seul nœud, et le groupe est nécessairement le groupe libre commutatif engendré par $k+1$ lettres. On a donc $(H_1|0) = (H_2|0) = 1$ et pour tout n strictement positif, on a $(H_1|n) = 2$ et $(H_2|n) \sim bn^{k-1}$. On a :

$$\begin{aligned}
(H_{\mathbb{G}}|n) &= (H_1|n)\sum_{i=0}^{n}(H_2|i) + (H_2|n)\sum_{i=0}^{n-1}(H_1|i) \\
&= 2b\sum_{i=0}^{n} b_n/bi^{k-1} + b_n n^{k-1}(2n-1) \\
&\sim cn^k.
\end{aligned}$$

$H_{\mathbb{G}}$ a pour singularité dominante 1, de multiplicité $k+1$.

Enfin, si $\rho_{H_1} = 1$ et $\rho_{H_2} < 1$ (pour le dernier cas, on peut se ramener à celui-ci) :

$$\begin{aligned}
(H_1|n)\sum_{i=0}^{n}(H_2|i) &= 2(\sum_{i=0}^{n} b_i\rho_{H_2}^{-i}i^{k-1}) \\
&= 2b\rho_{H_2}^{-n}(\sum_{i=0}^{n}(b_{n-i}/b)\rho_{H_2}^{i}(n-i)^{k-1}) \\
&\sim 2bn^k(1-\rho_{H_2})^{-1}(\rho_{H_2})^{-n}
\end{aligned}$$

et

$$\begin{aligned}
(H_2|n)\sum_{i=0}^{n}(H_1|i) &= b_n n^{k-1}\rho_{H_2}^{-n}(2n+1) \\
&\sim 2bn^{k-1}(\rho_{H_2}\rho_{H_1})^{-n}
\end{aligned}$$

En injectant de nouveau dans l'équation 2.6, on obtient $\rho_{H_{\mathbb{G}}} = \rho_{H_2}$ et la multiplicité de cette singularité est $k+1$, car il y a une composante à un seul élément de plus par rapport à H_2.

Ainsi, la proposition est prouvée pour m composantes. Elle est donc prouvée dans le cas général. □

Ces deux propositions permettent de connaître le comportement asymptotique des séries $L_{\mathbb{G}}$ et $H_{\mathbb{G}}$:

$$\begin{aligned}
(L_{\mathbb{G}}|n) &\sim \frac{\rho_{L_{\mathbb{G}}}}{(k_{L_{\mathbb{G}}}-1)!}[L_{\mathbb{G}}(y)(\rho_{L_{\mathbb{G}}}-y)^{k_{L_{\mathbb{G}}}}]_{|y=\rho_{L_{\mathbb{G}}}} n^{k_{L_{\mathbb{G}}}-1}\rho_{L_{\mathbb{G}}}^{-n} \\
(H_{\mathbb{G}}|n) &\sim \frac{\rho_{H_{\mathbb{G}}}}{(k_{H_{\mathbb{G}}}-1)!}[H_{\mathbb{G}}(y)(\rho_{H_{\mathbb{G}}}-y)^{k_{H_{\mathbb{G}}}}]_{|y=\rho_{H_{\mathbb{G}}}} n^{k_{H_{\mathbb{G}}}-1}\rho_{H_{\mathbb{G}}}^{-n}
\end{aligned}$$

2.3.2 Probabilité uniforme sur les traces

On veut maintenant savoir quelle est en moyenne la hauteur d'un mot comparée à sa longueur, et d'abord si cette limite existe. Pour cela, on se fixe une mesure de probabilité sur les traces d'une longueur donnée : on note $\mathbb{M}_n$ (resp. $\mathbb{G}_n$) l'ensemble des traces de longueur n du monoïde $\mathbb{M}$ (resp. du groupe $\mathbb{G}$), et on définit une probabilité uniforme P_n sur $\mathbb{M}_n$ (resp. $\mathbb{G}_n$) : $P_n\{t\} = 1/|\mathbb{M}_n|$ (resp. $P_n\{t\} = 1/|\mathbb{G}_n|$) pour tout $t \in \mathbb{M}_n$ (resp. $\mathbb{G}_n$). La hauteur moyenne des mots de $\mathbb{M}_n$ (resp. $\mathbb{G}_n$) est alors :

$$\mathbb{E}_n[h] = \sum_{t\in\mathbb{M}_n} P_n\{t\}h(t).$$

Sous réserve d'existence de la limite, on appelle $\lim_n \mathbb{E}_n[h]/n$ la hauteur moyenne asymptotique, et on la note $\lambda_{\mathbb{M}}$, (resp. $\lambda_{\mathbb{G}}$). On a

$$\lambda_{\mathbb{G}} = \lim_n \frac{E_n[h]}{n} = \lim_n \frac{\sum_{t\in\mathbb{G}_n} h(t)}{n|\mathbb{G}_n|} = \lim_n \frac{\sum_m m f_{m,n}}{\sum_m n f_{m,n}}$$

où $f_{m,n}$ le nombre de traces de longueur n et de hauteur m dans $\mathbb{G}$.

Dans le cas du monoïde, on sait déjà ([36]) que cette limite existe, et qu'elle est algébrique.

De même, on définit la longueur moyenne asymptotique, en faisant jouer à la longueur et la hauteur des rôles symétriques : on définit la probabilité uniforme $\tilde{P}_n$ sur les traces de hauteur n. On note ${}_n\mathbb{G}$ le nombre de traces du groupe $\mathbb{G}$ de hauteur n, et $\tilde{P}_n\{t\} = 1/|{}_n\mathbb{G}|$. La longueur moyenne d'un mot pour une hauteur donnée est alors :

$$\mathbb{E}_n[\ell] = \sum_{t\in{}_n\mathbb{G}} \tilde{P}_n\{t\}|t|.$$

et on note $\gamma_{\mathbb{G}}$ la limite, sous réserve d'existence :

$$\gamma_{\mathbb{G}} = \lim_m \frac{E_m[\ell]}{m} = \lim_m \frac{\sum_{t\in{}_m\mathbb{G}} h(t)}{m|{}_m\mathbb{G}|} = \lim_m \frac{\sum_n n f_{m,n}}{\sum_n m f_{m,n}}$$

On sait aussi ([64]) que cette limite existe et est algébrique pour le monoïde.

Proposition 2.37 *Les limites $\lambda_{\mathbb{G}}$ et $\gamma_{\mathbb{G}}$ existent et ces nombres sont algébriques.*

En outre, la preuve de cette proposition, donnée en fin de chapitre, nous permet d'avoir une formule explicite de ces limites. Soient $G = (\partial F/\partial x)(1,y)$ et $\tilde{G} = (\partial F/\partial y)(x,1)$. On a alors

$$\lambda_{\mathbb{G}} = \frac{[G(y)(\rho_L - y)^{k_L+1}]_{|y=\rho_L}}{k_L\rho_L.[L(y)(\rho_L - y)^{k_L}]_{|y=\rho_L}} \qquad \text{et} \qquad \gamma_{\mathbb{G}} = \frac{[\tilde{G}(x)(\rho_H - x)^{k_H+1}]_{|x=\rho_H}}{k_H\rho_H.[H(x)(\rho_H - x)^{k_H}]_{|x=\rho_H}}.$$

2.3.3 Probabilité uniforme sur les mots

Jusqu'ici, on a réussi à transposer ce que l'on savait sur les monoïdes de traces aux groupes de traces. On va maintenant étudier la hauteur asymptotique des traces, mais au lieu de prendre une probabilité uniforme sur les traces, on considère une probabilité uniforme sur les mots de l'alphabet $\tilde{\Sigma}$. Pour le monoïde, l'existence de la hauteur moyenne a déjà été prouvée dans [64], et on peut calculer la hauteur d'un mot grâce à un automate à multiplicité sur le semi-anneau (max,plus).

Pour les groupes, on peut aussi montrer l'existence de la hauteur asymptotique, grâce au théorème de Kingman :

Théorème 2.38 (Kingman) *Soit $(X, \mathcal{A}, P, T)$ un espace de probabilités muni d'un shift T stationnaire et ergodique. Soit $(A_i)_{i\in\mathbb{N}^*}$ une suite de variables aléatoires réelles. Si*

1. *$(A_i)_{i\in\mathbb{N}^*}$ est sous-additive ($A_{m+n} \leq A_m + A_n$) p.s., et*
2. *$\exists c \in \mathbb{R},\ \mathbb{E}[A_i] \geq ci$.*

Alors $\exists\gamma \in \mathbb{R}$ tel que $\frac{1}{n}A_n \overset{p.s.\ \mathbb{L}_1}{\underset{n}{\longrightarrow}} \gamma$.

Pour tout n, on met une probabilité uniforme $\mathbb{Q}_n$ sur les mots de longueur n : soit $w \in \tilde{\Sigma}^n$, alors $\mathbb{Q}_n\{w\} = 1/|\tilde{\Sigma}|^n$. Pour tous $u, v \in \mathbb{G}$, on a clairement $h(uv) \leq h(u) + h(v)$. Soit une suite $(B_i)_{i\in\mathbb{N}^*}$ de variables i.i.d. de $\tilde{\Sigma}$. On définit A_i comme la hauteur du tas formé des i premières pièces de la suite B : $A_i = h(B_1 \dots B_i)$. La suite $(A_i)_{i\in\mathbb{N}^*}$ est sous-additive, positive. Donc on peut appliquer le théorème de Kingman : il existe un réel, que l'on note λ^*, tel que la hauteur d'un tas de pièce est asymptotiquement équivalente à $\lambda^* n$.

De plus, on peut comparer les hauteurs asymptotiques obtenues pour les monoïdes et pour les groupes. Notons $\lambda^*_{\mathbb{M}}$ (resp. $\lambda^*_{\mathbb{G}}$) la hauteur moyenne asymptotique du monoïde de traces $\mathbb{M}$ (resp. du groupe de traces $\mathbb{G}$). On a :

Proposition 2.39 *Soit Σ un alphabet fini, et D une relation réflexive et symétrique sur Σ. On pose $\mathbb{M} = \mathbb{M}(\Sigma, D)$,$\tilde{\mathbb{M}} = \mathbb{M}(\tilde{\Sigma}, \tilde{D})$ et $\mathbb{G} = \mathbb{G}(\tilde{\Sigma}, \tilde{D})$. Alors on a :*

$$\lambda^*_{\mathbb{G}} \leq \lambda^*_{\tilde{\mathbb{M}}} = \lambda^*_{\mathbb{M}}$$

Démonstration. Soit un mot w de $\tilde{\Sigma}$. On a toujours $h_{\tilde{\mathbb{M}}}(w) \geq h_{\mathbb{G}}(w)$, puisque des simplifications sont permises dans le groupe et pas dans le monoïde. Or

$$\lambda_S = \lim_n \frac{\sum_{w\in\tilde{\Sigma}^n} h_S(w)}{n|\tilde{\Sigma}|^n}\ ,\ S = \mathbb{G} \text{ ou } \tilde{\mathbb{M}}.$$

L'égalité de droite est facile à voir en oubliant le coloriage des pièces de $\tilde{\mathbb{M}}$. □

Comme pour les monoïdes, on aimerait pouvoir modéliser l'évolution d'un tas de pièces colorées, et en particulier sa hauteur, grâce à des applications topicales. Mais nous n'avons pas trouvé de telle modélisation, et même nous ne savons même pas si une telle modélisation existe. En effet, pour connaître l'évolution d'un tas de pièces pour un monoïde de traces, il suffit de représenter son profil (la forme supérieure du tas sans indication des pièces qui le constituent). On peut alors déterminer le profil exact du tas où on a fait tomber une pièce supplémentaire. Ainsi, pour des exemples simples, on peut trouver une représentation par une chaîne de Markov qui soit assez simple pour mener à bien les calculs de la hauteur moyenne d'une trace. Mais dans le cas du groupe, les pièces peuvent s'annuler, et donc le tas peut disparaître en fonction des pièces du tas. On ne peut donc pas vraiment simplifier la représentation du tas de pièces pour obtenir sa hauteur. Ainsi, même pour des exemples simples (hors groupe libre, et groupe libre commutatif), on ne peut rien calculer explicitement sur la représentation par une chaîne de Markov.

Exemple 2.40 Considérons le graphe de dépendance :

$$a \text{ — } b \text{ — } c$$

On obtient pour le monoïde $\lambda^*_{\mathbb{M}} = \frac{10+\sqrt{5}}{15}$ grâce à un argument utilisant les chaînes de Markov.

On peut néanmoins calculer les valeurs de λ^* pour les deux cas extrêmes que sont le groupe libre et le groupe commutatif libre. Avec $k = |\Sigma|$, on a le tableau suivant.

λ^*	Monoïde	Groupe
libre	1	$1 - 1/k$
commutatif libre	$1/k$	0

Pour le groupe libre, on peut modéliser la hauteur par une chaîne de Markov : soit h_n la hauteur de l'empilement des n premières pièces. Alors (h_n) est une chaîne de Markov, et l'on a :

$$\mathbb{P}(h_{n+1} = h_n + 1) = 1/2k \qquad \mathbb{P}(h_{n+1} = h_n - 1) = (2k-1)/2k.$$

Donc la hauteur asymptotique moyenne est $1 - 1/k$. Pour le groupe libre commutatif, on peut aussi modéliser l'évolution du tas par une marche aléatoire sur $\mathbb{N}^k$. Un état est l'ensemble des hauteurs de chaque colonne. La chaîne est récurrente nulle. Donc $\lambda^* = 0$.

En fait, on peut obtenir la hauteur de la trace d'un groupe en deux étapes : tout d'abord, on réduit le mot, c'est à dire que l'on enlève toutes les lettres qui s'annulent. Puis on calcule la hauteur, comme si l'on était dans un monoïde. Il existe, toujours d'après le théorème de Kingman, une limite asymptotique presque sure de la longueur de la trace représentée par un mot. Notons $\alpha = \lim_n \frac{|\psi(w_1 \ldots w_n)|}{n}$ où $(w_n)_{n\in\mathbb{N}^*} \in \tilde{\Sigma}^{\mathbb{N}^*}$ est une suite de v.a. i.i.d. On a

$$\begin{aligned} \lambda^*_{\mathbb{G}} &= \lim_n \frac{h(\psi(w_1 \ldots w_n))}{n} \\ &= \lim_n \frac{h(\psi(w_1 \ldots w_n))}{|\psi(w_1 \ldots w_n)|} \frac{|\psi(w_1 \ldots w_n)|}{n}. \end{aligned}$$

On note $\gamma = \lim_n \frac{h(\psi(w_1 \ldots w_n))}{|\psi(w_1 \cdots_n)|}$. On a $\alpha\gamma = \lambda^*_{\mathbb{G}}$. Intuitivement on peut se demander si $\gamma = \lambda^*_{\mathbb{M}}$.

Les résultats des simulations que nous avons effectuées montrent que ce n'est pas le cas. Le tableau des résultats est donné dans le paragraphe 2.7.3. Dans ce tableau, on vérifie que l'on a bien $\lambda^*_{\mathbb{M}} \geq \lambda^*_{\mathbb{G}}$, et on constate que $\gamma \geq \lambda^*_{\mathbb{M}}$ et en général $\gamma \neq \lambda^*_{\mathbb{M}}$. De plus, si l'on calcule la différence $\lambda^*_{\mathbb{M}} - \lambda^*_{\mathbb{G}}$, on remarque qu'elle est proche de $1/|\Sigma|$. L'égalité $\lambda^*_{\mathbb{M}} - \lambda^*_{\mathbb{G}} = 1/|\Sigma|$ est vérifiée pour le groupe libre et le groupe commutatif libre. Pour les autres graphes de dépendance, on ne peut pas faire de calculs. Mais on remarque qu'à deux centièmes près, l'égalité est vérifiée.

2.4 Monoïde involutif de traces.

On se place toujours dans l'alphabet $\Sigma \cup \bar{\Sigma}$. Mais la structure associée aux traces n'est plus un groupe : aux relations de commutation attribuées précédemment, on ajoute seulement $a\bar{a} = \epsilon$, et pas $\bar{a}a = \epsilon$. On peut considérer que les lettres barrées modélisent une annulation de la tâche correspondant à la même lettre non barrée. Mais cette annulation n'a pas toujours lieu, même quand une annulation est lancée : si elle est lancée trop tard, et qu'elle arrive après une tâche avec laquelle elle ne commute pas, la pièce la modélisant reste dans le tas. On a la définition plus formelle suivante.

Définition 2.41 *Le monoïde $\mathbb{A}(\tilde{\Sigma}, D)$ est le monoïde défini par les générateurs $\tilde{\Sigma}$ et les relations : $ab = ba$, $\bar{a}b = b\bar{a}$, $a\bar{b} = \bar{b}a$ et $\bar{a}\bar{b} = \bar{b}\bar{a}$, $(a,b) \notin D$ d'une part et $a\bar{a} = \varepsilon$ $\forall a \in \Sigma$ d'autre part.*

Comme pour les deux structures précédentes, on définit le graphe des cliques, et les cliques CF-admissibles : (u, v) est CF-admissible si

$$\forall b \in \mathrm{alph}(v)\ (\exists a \in \mathrm{alph}(u) \text{ tel que } aDb \text{ et } b \in \bar{\Sigma} \Rightarrow \bar{b} \notin \mathrm{alph}(u)).$$

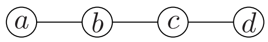
Il existe aussi une formule agréable pour la série génératrice des longueurs de la structure asymétrique $\mathbb{A}$. En effet, il existe une bijection entre les éléments de cette structure et celle du monoïde de trace noté $\mathbb{M}_{\mathbb{A}}$ défini sur le même alphabet $\tilde{\Sigma}$ que $\mathbb{A}$, avec la relation de dépendance $\tilde{D}$:

1. $aDb \Rightarrow a\tilde{D}b$, $a\tilde{D}\bar{b}$, $\bar{a}\tilde{D}b$, $\bar{a}\tilde{D}\bar{b}$ $\forall a \neq b$,
2. $\bar{a}\tilde{I}a$, $\forall a \in \tilde{\Sigma}$.

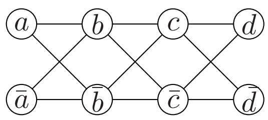
En effet si l'on prend l'ordre sur $\tilde{\Sigma}$: $\bar{a} < a < \bar{b} < b < \cdots$, les représentants en forme normale lexicographique pour ces deux structures sont exactement les mêmes, puisqu'on autorise dans $\mathbb{A}$ les empilements du type $\bar{a}a$, mais pas ceux du type $a\bar{a}$.

On obtient aussi les mêmes résultats que pour les monoïdes et les groupes quant aux hauteurs asymptotiques des traces de cette structure.

Exemple 2.42 Si l'on reprend le premier exemple :



Le graphe associé au monoïde de même série génératrice est :



On a donc :

$$L_{\mathbb{A}} = \frac{1}{1 - 8x + 16x^2 - 12x^3 + 3x^4}$$

2.5 Une application aux systèmes à événements discrets

Considérons un modèle tâches-ressources, comme proposé dans [49] : une fabrique exécute un nombre fini de tâches Σ à l'aide d'un ensemble de ressources. Chaque tâche requiert un sous-ensemble donné de ressources pour être exécutée. Soit $D \in \Sigma \times \Sigma$ la relation symétrique et réflexive constituée des paires de tâches qui nécessitent des ressources non disjointes. Deux tâches peuvent être exécutées simultanément si et seulement si elles nécessitent des ressources disjointes. Il est donc naturel de modéliser l'exécution des tâches par le monoïde de traces $\mathbb{M}(\Sigma, D)$. Dans ce contexte, les deux principales mesures de performances associées à une trace t sont sa longueur $|t|$ (le temps d'exécution séquentiel) et sa hauteur $h(t)$ (le temps d'exécution parallèle).

Maintenant, on modélise l'ordre (aléatoire) des commandes par une suite $(\xi_n)_{n\in\mathbb{N}}$ de variables aléatoires indépendantes uniformément distribuées sur Σ. Le temps de cycle de la fabrique est alors $\lambda^*_{\mathbb{M}(\Sigma,D)} = \lim_n n^{-1}\mathbb{E}[h \circ \pi(\xi_0\xi_1 \ldots \xi_{n-1})]$ (voir paragraphe précédent).

Ici, on veut en plus quantifier la résistance aux pannes et aux annulations d'ordres. Pour ce faire, considérons le modèle suivant. La production à l'étape n est donnée par une trace t_n dans $\mathbb{M}(\Sigma, D)$. Néanmoins, on se place dans $\mathbb{G}(\tilde{\Sigma}, \tilde{D})$, et l'annulation de la tâche $a \in \Sigma$ est représentée par la lettre $\bar{a}$. On rappelle que $Max(t_n)$ est l'ensemble des lettres a telles que $|t_n\bar{a}| = |t_n| - 1$. La production à l'étape $n+1$ est une variable aléatoire t_{n+1} donnée par :

$$\begin{cases} \forall a \in Max(t_n), & \mathbb{P}(t_{n+1} = t_n\bar{a}) = \alpha/K,\ \mathbb{P}(t_{n+1} = t_n a) = 1/K \\ \forall a \in \Sigma\backslash Max(t_n), & \mathbb{P}(t_{n+1} = t_n a) = 2/K \end{cases},$$

où α est un paramètre fixé et $K = (\alpha-1)\#Max(t_n)+2\#\Sigma$ est la constante de normalisation. Remarquons que pour $\alpha = 1$ on a $\lim_n n^{-1}\mathbb{E}[h \circ \pi(t_n)] = \lambda^*_{\mathbb{G}(\tilde{\Sigma},\tilde{D})}$. Sous réserve d'existence de la limite, on définit le seuil $\alpha_* \in \mathbb{R}_+$ tel que presque sûrement,

$$\begin{cases} \lim_n n^{-1}\mathbb{E}[h \circ \pi(t_n)] > 0 & \text{si } \alpha < \alpha_* \\ \lim_n n^{-1}\mathbb{E}[h \circ \pi(t_n)] = 0 & \text{si } \alpha > \alpha_* \end{cases}.$$

Le paramètre α représente l'intensité des ordres d'annulation, et α_* l'intensité au-dessus de laquelle la fabrique ne produit plus rien.

Adapté à notre contexte, le résultat de Lyons [68] montre l'existence de α_* et que $\alpha_* = \rho_{L_{\mathbb{G}(\tilde{\Sigma},\tilde{D})}}$. D'où une interprétation inattendue : le degré de résistance aux annulations dans un système tâches-ressources peut être paramétré par $\rho_{L_{\mathbb{G}(\tilde{\Sigma},\tilde{D})}}$.

*
* *

L'étude faite ici montre que les groupes de traces ont les mêmes propriétés que les monoïdes de traces : la hauteur moyenne des traces converge asymptotiquement lorsqu'on met une probabilité uniforme sur les mots ou sur les traces. De plus, on sait calculer de manière simple la série génératrice des traces pour le groupe. Par contre, les calculs déjà difficiles pour les monoïdes le sont encore plus pour les groupes, en particulier pour la hauteur asymptotique des tas de pièces.

2.6 Preuves

On donne la preuve de la proposition 2.35 pour $\lambda_{\mathbb{G}}$, la preuve pour $\gamma_{\mathbb{G}}$ étant identique. Une démonstration directe a déjà été donnée concernant la singularité dominante de L, mais nous allons en donner une autre, beaucoup plus longue, mais plus précise : elle montre égalité entre la singularité dominante de L et l'inverse du rayon spectral de A. Toutefois, pour être complet, il aurait quand même fallu faire le même raisonnement pour H.

On rappelle que $L(y) = F(1, y) = \sum_{t\in\mathbb{M}} y^{|t|}$ est la série génératrice des longueurs des traces. Soit

$$G = \frac{\partial F}{\partial x}(1, y) = \sum_{t\in\mathbb{M}} h(t) y^{|t|}.$$

Sous réserve d'existence, on a donc

$$\lambda_{\mathbb{M}} = \lim_n (G|n)/(n(L|n)). \tag{2.7}$$

2.6.1 Sur les singularités de L et G et le rayon spectral de A

Les séries L et G sont à coefficients positifs (puisqu'on compte des objets), donc ils ont une singularité dominante réelle positive (théorème de Pringsheim). Notons les respectivement ρ_L et ρ_G. On a les inégalités $n(L|n)C^{-1} \leq (G|n) \leq n(L|n)$ puisque $|t|/C \leq h(t) \leq |t|$ (C est la taille maximale d'une clique). Donc $\rho_L = \rho_G$. Soit k_L la multiplicité de cette singularité dans L.

Hypothèses :

1. ρ_L est l'unique singularité dominante dans L.
2. La multiplicité de ρ_L dans G est $k_L + 1$ et cette multiplicité est strictement plus grand que les multiplicités des autres singularités dominantes de G de module ρ_L.

Alors $\lambda_{\mathbb{G}}$ existe et on déduit aisément de l'équation(2.7) que :

$$\lambda_{\mathbb{G}} = \frac{[G(y)(\rho_L - y)^{k_L+1}]_{|y=\rho_L}}{k_L\rho_L.[L(y)(\rho_L - y)^{k_L}]_{|y=\rho_L}}.$$

En particulier $\lambda_{\mathbb{G}}$ est algébrique. En outre, on a une expression explicite pour cette limite. Reste maintenant à prouver que les suppositions faites sont vraies. On sait que $F(x, y) = 1 + u(I - A(x, y))^{-1}v(x, y)$, mais on va plutôt travailler sur une partition équitable des cliques $\pi = \{\mathfrak{C}_1, \ldots, \mathfrak{C}_m\}$, où pour chaque élément $\mathfrak{C}_i$ de la partition π, toutes les cliques ont la même longueur ℓ_i. On précisera quelle est cette partition par la suite. Alors,

$$F(x, y) = 1 + u_\pi(I - A_\pi(x, y))^{-1}v_\pi(x, y) = 1 + \frac{u_\pi Adj(I - A_\pi(x, y))v_\pi(x, y)}{\det(I - A_\pi(x, y))}.$$

On note $F(x, y) = P(x, y)/Q(x, y)$ avec $Q(x, y) = \det(I - A_\pi(x, y))$. Par conséquent, avec cette écriture L et G deviennent :

$$L(y) = P(1, y)/Q(1, y) \qquad \text{et} \qquad G(y) = \frac{1}{Q(1, y)}\left(\frac{\partial P}{\partial x}(1, y) - L\frac{\partial Q}{\partial x}(1, y)\right). \tag{2.8}$$

Les singularités de L et G sont donc des singularités de $1/Q(1, y)$, donc ρ_L est plus grand que le module de la singularité dominante de $1/Q(1, y)$. Montrons que les modules des singularités dominantes de L, G et $1/Q(1, y)$ sont égaux.

Une transformation d'écriture

On va maintenant transformer la représentation $(u_\pi, A_\pi(1,y), v_\pi(1,y))$ en une représentation $(\tilde{u}_\pi, y\tilde{A}_\pi, y\tilde{v}_\pi)$ de telle sorte que les vecteurs et la matrice $\tilde{u}_\pi$, $\tilde{A}_\pi$ $\tilde{v}_\pi$ aient tous leurs coefficients dans $\mathbb{N}$. L'idée est d'éclater les lignes représentant des cliques à k éléments en k lignes. On a alors la représentation suivante :

$E = \{(\mathfrak{C}_i, 1) \dots (\mathfrak{C}_i, \ell_i),\ i \in \mathbb{N}_m\}$, $\tilde{u}_\pi \in \mathbb{N}^{1\times E}$, $\tilde{A}_\pi \in \mathbb{N}^{E\times E}$ et $\tilde{v}_\pi \in \mathbb{N}^{E\times 1}$.

$$(\tilde{u})_{\pi i} = \begin{cases} |\mathfrak{C}_p| & \text{si } i = (\mathfrak{C}_p, 1),\ p \in \mathbb{N}_m \\ 0 & \text{sinon.} \end{cases}$$

$$(\tilde{A})_{\pi i,j} = \begin{cases} 1 & \text{si } i = (\mathfrak{C}_p, k),\ j = (\mathfrak{C}_p, k+1) \text{ et } p \in \mathbb{N}_m \\ n_{p,p} & \text{si } i = (\mathfrak{C}_p, \ell_p),\ j = (\mathfrak{C}_p, 1) \text{ et } p \in \mathbb{N}_m \\ n_{p,q} & \text{si } i = (\mathfrak{C}_p, \ell_p),\ j = (\mathfrak{C}_q, 1) \text{ et } A_{p,q} \neq 0 \\ 0 & \text{sinon.} \end{cases}$$

$$(\tilde{v})_{\pi i} = \begin{cases} 1 & \text{si } i = (\mathfrak{C}_p, \ell_p),\ p \in \mathbb{N}_m \\ 0 & \text{sinon.} \end{cases}$$

On rappelle que ℓ_i est la taille commune des cliques de $\mathfrak{C}_i$ et que $n_{p,q}$ est le nombre d'empilements CF-admissibles (u, v) avec $u \in \mathfrak{C}_p$ fixé et $v \in \mathfrak{C}_q$ (par définition, ce nombre ne dépend pas de u).

Il y a une bijection entre les chemins réussis de l'automate $(u_\pi, A_\pi(1,y), v_\pi(1,y))$ et ceux de $(\tilde{u}_\pi, y\tilde{A}_\pi, y\tilde{v}_\pi)$. On a donc :

$$L = 1 + u_\pi(I - A_\pi(1,y))^{-1} v_\pi(1,y) = 1 + \tilde{u}_\pi(I - y\tilde{A}_\pi)^{-1} y\tilde{v}_\pi^T.$$

On a aussi l'égalité suivante (une preuve est donnée dans [64]) :

$$Q(1,y) = \det(I - A_\pi(1,y)) = \det(I - y\tilde{A}_\pi).$$

Étude des valeurs propres de $\tilde{A}_\pi$.

Pour montrer que les singularités dominantes de L et G sont égales à celles de $1/Q(1,y)$, il suffit de montrer que la valeur propre maximale de $\tilde{A}_\pi$ est l'inverse de la singularité dominante de L. Pour ce faire, décomposons le graphe $(\tilde{\Sigma}, \tilde{D})$ en ses composantes fortement connexes (Σ_i, D_i), $i \in \mathfrak{U}$. On donne une première condition sur la partition π :

Condition 1 : $\forall i \in \{1, \dots, m\}$, $\mathfrak{C}_i$ est composé de cliques qui ont leurs éléments dans exactement les mêmes composantes fortement connexes de $(\tilde{\Sigma}, \tilde{D})$.

Cette condition peut toujours être vérifiée : il suffit de prendre la partition triviale (une clique par élément de la partition). On va maintenant ordonner les éléments de la partition de façon à obtenir une matrice A triangulaire supérieure par blocs. On regroupe dans le même bloc les éléments de π selon les composantes de $(\tilde{\Sigma}, \tilde{D})$ avec lesquelles elles ont une intersection non vide : si le support de $\mathfrak{C}_i$ est une union de cliques non vides de $(\tilde{\Sigma}_j)_{j\in J}$, alors on la groupe avec les $\mathfrak{C}_k$ qui dont le support est une union de cliques non vides de $(\tilde{\Sigma}_j)_{j\in J}$. On obtient alors une nouvelle partition de $\pi, \mathcal{U} = \{U_1, \dots, U_r\}$ et on note d_i le nombre de composantes fortement connexes qui composent les cliques des éléments de U_i. De plus, on ordonne les lignes et les colonnes de A_π selon ces ensembles : si $d_i \leq d_j$, alors le bloc U_i est placé au-dessous du bloc U_j. La matrice A_π est alors triangulaire supérieure par blocs, avec $A_{\pi U_1}, \dots, A_{\pi U_k}$ en diagonale, où les $A_{\pi U_i}$ sont les restrictions de A_π à U_i. De plus, ces blocs sont irréductibles. En effet, un empilement (u,v) n'est CF-admissible que si le support de v est inclus dans celui de u.

On a donc :

$$\begin{aligned} \det(I - A_\pi(1,y)) &= \prod_{i\in\{1,\ldots,r\}} \det(I - A_{\pi U_i}(1,y)) \\ \det(I - y\tilde{A}_\pi) &= \prod_{i\in\{1,\ldots,r\}} \det(I - y\tilde{A}_{\pi U_i}). \end{aligned}$$

Soit e_i un vecteur ligne nul sauf en position i, où le coefficient est égal à 1. Pour $i,j \in \mathbb{N}_m$, définissons $L_{i,j} = |\mathfrak{C}_i| e_i(I - A_\pi)(1,y))^{-1} y^{|j|} e_j^T$.

Le coefficient $(L_{i,j}|n)$ s'interprète comme le nombre de chemins de $\mathfrak{C}_i$ à $\mathfrak{C}_j$ de poids y^n dans l'automate de représentation $(u_\pi, A_\pi(1,y), v_\pi(1,y))$. On a de plus $L = 1 + \sum_{i,j} L_{i,j}$. Comme précédemment, on a :

$$L_{ij} = |\mathfrak{C}_i| e_i(I - A_\pi(1,y))^{-1} y^{|j|} e_j^T = e_{(i,1)}(I - y\tilde{A}_\pi)^{-1} y e_{(j,|j|)}^T$$

Considérons une ligne u. On note $A_{\pi[u]}$ la matrice obtenue à partir de A_π en remplaçant la ligne et la colonne u par une ligne et une colonne de zéros. On a alors :

$$\begin{aligned} L_{uu} &= |\mathfrak{C}_u| e_u(I - A_\pi(1,y))^{-1} y^{|u|} e_u^t = |\mathfrak{C}_u| y^{|u|} (I - A_\pi(1,y))^{-1}_{uu} \\ &= |\mathfrak{C}_u| y^{|u|} \frac{Adj(I - A_\pi(1,y))_{uu}}{\det(I - A_\pi(1,y))} = |\mathfrak{C}_u| y^{|u|} \frac{\det(I - A_\pi(1,y)_{[u]})}{\det(I - A_\pi(1,y)}. \end{aligned}$$

Notons maintenant $\tilde{A}_{\pi[u]}$ la matrice obtenue à partir de $\tilde{A}_\pi$ en remplaçant la ligne $(u,1)$ et la colonne $(u,|u|)$ par une ligne et une colonne de zéros. On obtient :

$$y^{|u|} \det(I - A_\pi(1,y)_{[u]}) = y \det(I - y\tilde{A}_{\pi[u]}).$$

Si u appartient à $U \in \mathcal{U}$, alors on note $\tilde{A}_{\pi U,[u]}$ la restriction de $\tilde{A}_{\pi[u]}$ à U. Les expressions ci-dessus ensemble donnent la formule suivante :

$$L_{uu} = |\mathfrak{C}_u| \frac{y \det(I - \tilde{A}_{\pi[u]})}{\det(I - y\tilde{A}_\pi)} = |\mathfrak{C}_u| \frac{y \det(I - \tilde{A}_{\pi U,[u]})}{\det(I - y\tilde{A}_{\pi U})}.$$

Or $\tilde{A}_{\pi U,[u]} \leq \tilde{A}_{\pi U}$, et ces deux matrices sont différentes. De plus $\tilde{A}_{\pi U}$ est irréductible. Donc on peut appliquer le théorème de Perron-Frobenius pour les matrices irréductibles : le rayon spectral de $\tilde{A}_{\pi U,[u]}$ est strictement plus petit que celui de $\tilde{A}_{\pi U}$. Pour toute matrice M, les racines du polynôme $\det(I - yM)$ sont les inverses des valeurs propres non nulles de M. Donc il n'y a pas de simplification possible entre le numérateur et le dénominateur de l'équation précédente concernant une singularité dominante : les singularités dominantes de L_{uu} sont exactement les singularités dominantes de $1/\det(I - y\tilde{A}_{\pi U})$.

Or $(L|n) \geq (L_{uu}|n)$ pour tout n, donc la singularité dominante de L est plus petite que celle de L_{uu}. Nous en déduisons donc que la singularité dominante de L est de module plus petit que $1/\det(I - y\tilde{A}_{\pi U})$, et ce pour tout U. Donc elle est plus petite que celle de $1/Q(1,y)$. Comme $(G|n) \geq (L|n)$, on a le même résultat pour G.

Ceci montre donc que les singularités dominantes de L, G et $1/Q(1,y)$ sont égales en module : c'est aussi l'inverse du rayon spectral de $\tilde{A}_\pi$.

Si $(\tilde{\Sigma}, \tilde{D})$ est fortement connexe, la matrice $\tilde{A}_\pi$ est irréductible et primitive (pour tout $a \in \tilde{\Sigma}$, $\tilde{A}_{\pi(a,1)(a,1)} > 0$). Donc d'après le théorème de Perron Frobenius, la matrice $\tilde{A}_\pi$ a une unique valeur propre de module maximal, de multiplicité 1 et réelle positive. On en conclut donc que ρ_L est l'unique singularité dominante de L et G, puisque $nC^{-1}(L|n) < (G|n) < n(L|n)$. La multiplicité de ρ_L dans G est alors 2. L'hypothèse 1 est donc vérifiée dans ce cas. Dans le cas de H, on a prouvé que la singularité dominante de H est de multiplicité 1.

Revenons au cas général.

2.6.2 Cas général

Nous savons maintenant que L a une unique singularité dominante. En effet, la proposition 2.35 vient d'être démontrée, et on peut donc appliquer la proposition 2.36. Étudions les singularités dominantes de G, et donc les valeurs propres de module maximal de $\tilde{A}_\pi$. La matrice $\tilde{A}_\pi$ est triangulaire supérieure par blocs, donc il suffit d'étudier ces blocs. Fixons une partie V de la partition $\mathcal{U}$, et considérons le bloc diagonal $\tilde{A}_V$. Distinguons deux cas :

Cas 1 Supposons qu'une des composantes $(\tilde{\Sigma}_s, \tilde{D}_s)$ de $(\tilde{\Sigma}, \tilde{D})$ qui apparaît dans V est différente du graphe complet. Alors il existe deux ensembles de cliques $\mathfrak{C}_i, \mathfrak{C}_j \in V$ tels que $\ell_i = \ell_j + 1$. Donc $\tilde{A}_{\pi V}$ est apériodique (à $\mathfrak{C}_i$ correspond un cycle de longueur ℓ_i dans la matrice $\tilde{A}_\pi$ et à $\mathfrak{C}_j$ un cycle de longueur ℓ_j.) Comme $\tilde{A}_{\pi V}$ est irréductible, elle est primitive. D'après le théorème de Perron Frobenius, la matrice $\tilde{A}_{\pi V}$ a une unique valeur propre de module maximal, qui de plus est réelle, positive et de multiplicité 1.

Cas 2 Supposons maintenant que toutes les composantes $(\tilde{\Sigma}_i, \tilde{D}_i)$ de $(\tilde{\Sigma}, \tilde{D})$ qui apparaissent dans V sont des graphes complets. On peut trouver une partition π qui regroupe toutes les cliques de ces graphes, et plus précisément, on donne une deuxième condition sur π :

Construction d'une partition particulière : Considérons toutes les composantes fortement connexes de $(\tilde{\Sigma}, \tilde{D})$ qui sont des graphes complets. Soit W l'ensemble des parties non vides de l'ensemble de ces composantes. On regroupe tous les éléments de W qui sont isomorphes : soit $(n_1, \ldots, n_k) \in \mathbb{N}^k$ une suite d'entiers tels que $n_1 \leq \cdots \leq n_k$. On met dans le même élément de π toutes les parties (caractérisées par leur alphabet) $(\tilde{\Sigma}_{i_1}, \ldots, \tilde{\Sigma}_{i_k})$ qui vérifient $|\tilde{\Sigma}_{i_1}| = n_1, \ldots, |\tilde{\Sigma}_{i_k}| = n_k$.

Cette partition est compatible avec la condition 1, puisqu'on ne regroupe dans le même élément de π que des cliques qui ont le même nombre de lettres, et on choisit toujours une et une seule lettre par composante fortement connexe.

Avec une telle décomposition, les blocs diagonaux de A_π correspondant à des sous-graphes complets maximaux de $(\tilde{\Sigma}, \tilde{D})$ sont de dimension 1, et correspondent donc à un unique élément $\mathfrak{C}_i$ de π. Soit $n_1, \ldots, n_k$ la suite des tailles des alphabets de taille au moins 2 dont est issue chaque lettre d'une clique quelconque de $\mathfrak{C}_i$. Le coefficient de $A_{\pi\{\mathfrak{C}_i\}}$ est $y^k \prod_{j=1}^k (n_j - 1)$: chaque clique peut avoir pour successeur CF-admissible une clique qui ne contient pas de lettre inverse à celles de la clique. Notons $K = \prod_{j=1}^k (n_j - 1)$. Alors, on a :

$$\det(I - y^k A_{\pi\{\mathfrak{C}_i\}}(1,1)) = (1 - Ky^k).$$

Pour revenir à l'étude de $\tilde{A}_{\pi\{\mathfrak{C}_i\}}$, les valeurs propres sont

$$K^{1/k} \exp\left(\frac{2i\pi j}{k}\right),$$

avec multiplicité 1 pour tous, et $\rho(\tilde{A}_{\pi\{\mathfrak{C}_i\}}) = K^{1/k}$.

Le rayon spectral de $\tilde{A}_\pi$ est $\rho(\tilde{A}_\pi) = \max_{V\in\mathcal{U}} \rho(\tilde{A}_{\pi V})$. Définissons les deux ensembles suivants : $\mathcal{V} = \{V \subset \mathcal{U} : \forall v \in V, I_v = \emptyset\}$ et $\mathcal{S} = \{V \in \mathcal{U} : \rho(\tilde{A}_{\pi V}) = \rho(\tilde{A}_\pi)\}$. Selon les deux cas étudiés, nous pouvons de nouveau distinguer deux cas :

$\mathcal{V}\cap\mathcal{S} = \emptyset$: le premier cas s'applique : $\tilde{A}_\pi$ n'a qu'une valeur propre de module maximal. $\rho(\tilde{A}_\pi)^{-1} = \rho_L = \rho_G$ est l'unique singularité dominante de G.

Supposons maintenant qu'il existe $U \in \mathcal{V}\cap\mathcal{S}$. Le deuxième cas implique que $U = \{\mathfrak{C}_i\}$ et $\rho(\tilde{A}_\pi) = \rho(\tilde{A}_{\pi U}) = (\prod_{i\in\mathbb{N}_k}(n_k - 1))^{1/k}$, où les n_i sont les tailles des alphabets des composantes qui interviennent pour une clique quelconque de $\mathfrak{C}_i$. Pour être maximal, tous les n_i doivent être égaux, et cette valeur commune doit être maximale : soit

$$\mathcal{V}\cap\mathcal{S} = \mathcal{P}(V), \text{ avec } V = \{v \in \mathcal{V} : |\tilde{\Sigma}_v| = \max_{x\in\mathcal{V}} |\tilde{\Sigma}_x|\}$$

L'ensemble des valeurs propres maximales de $\tilde{A}_\pi$ est donc donné précisément donné par :

$$\bigcup_{L=1}^{|V|} \left\{ (\max_{x\in\mathcal{V}} |\tilde{\Sigma}_x| - 1) \exp(2i\pi\ell/L),\ \ell = 0, \ldots, L-1 \right\}. \tag{2.9}$$

Soit $K = \max_{x\in\mathcal{V}}(|\Sigma_x| - 1)$. D'après la construction, toutes les valeurs propres maximales apparaissent avec une multiplicité égale au nombre de fois où elles apparaissent dans l'équation (2.9). On remarque alors que K apparaît avec une multiplicité $|V|$. Ensuite, la valeur propre maximale qui apparaît avec la seconde plus grande multiplicité est $-K$, qui apparaît avec multiplicité $\lfloor |V|/2 \rfloor$. Il s'ensuit que la singularité dominante non positive réelle de plus grande multiplicité de $1/Q(1, y)$ a pour multiplicité $\lfloor |V|/2 \rfloor$. Or d'après l'équation (2.8), on en déduit que la multiplicité maximale d'une singularité dominante non réelle positive de G est au plus $\lfloor |V|/2 \rfloor$.

De plus, on peut écrire $L = \prod_{v\in V} L_v \prod_{v\notin V} L_v$. La fonction de Möbius de $(\tilde{\Sigma}_v, I_v), v \in V$ est $(1 - Ky)$, et donc :

$$L = \frac{1}{(1 - Ky)^{|V|}} \prod_{v\notin V} L_v.$$

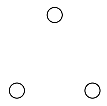
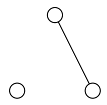
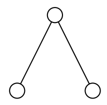
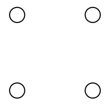
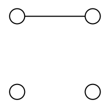
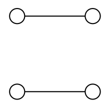
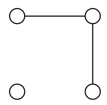
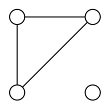
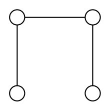
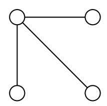
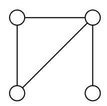
La multiplicité de la singularité $1/K$ dans L est alors au moins $|V|$. Puisque $n(L|n)/C \leq (G|n) \leq n(L|n)$, on déduit que la multiplicité de la singularité dominante de G est $k_L+1 \geq |V|+1$. La multiplicité maximale d'une singularité dominante non réelle positive de $\mathbb{G}$ est $\lfloor |V|/2 \rfloor$

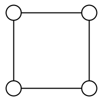
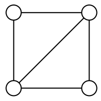
Les hypothèses faites en début de preuve sont donc vérifiées, et ceci achève la preuve.

2.7 Tableaux

2.7.1 Fonctions de Möbius

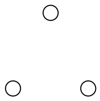
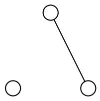
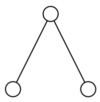
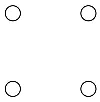
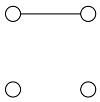
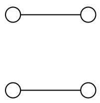
(Σ, D)	$\mu_{\mathbb{M}}$	$\mu_{\mathbb{G}}$
K_n	$(1 - nx)$	$\frac{1-(2n-1)x}{1+x}$
○ ○	$(1 - x)^2$	$\frac{(1-x)^2}{(1+x)^2}$

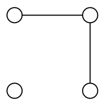
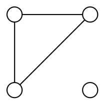
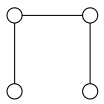
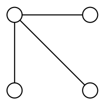
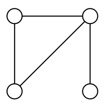
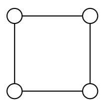
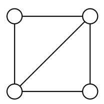
(Σ, D)	$\mu_{\mathbb{M}}$	$\mu_{\mathbb{G}}$
	$(1-x)^3$	$\frac{(1-x)^3}{(1+x)^3}$
	$(1-x)(1-2x)$	$\frac{(1-x)(1-3x)}{(1+x)^2}$
	$1-3x+x^2$	$\frac{1-4x-x^2}{(1+x)^2}$
	$(1-x)^4$	$\frac{(1-x)^4}{(1+x)^4}$
	$(1-x)^2(1-2x)$	$\frac{(1-x)^2(1-3x)}{(1+x)^3}$
	$(1-2x)^2$	$\frac{(1-3x)^2}{(1+x)^2}$
	$(1-x)(1-3x-x^2)$	$\frac{(1-x)(1-4x-x^2}{(1+x)^3}$
	$(1-x)(1-3x)$	$\frac{(1-x)(1-5x)}{(1+x)^2}$
	$(1-x)(1-3x)$	$\frac{(1-x)(1-5x)}{(1+x)^2}$
	$1-4x+3x^2-x^3$	$\frac{1-5x-x^2-3x^3}{(1+x)^3}$
	$1-4x+2x^2$	$\frac{1-6x+x^2}{(1+x)^2}$

(Σ, D)	$\mu_{\mathbb{M}}$	$\mu_{\mathbb{G}}$
	$1-4x+2x^2$	$\frac{1-6x+x^2}{(1+x)^2}$
	$1-4x+x^2$	$\frac{1-6x-3x^2}{(1+x)^2}$

2.7.2 Taux de croissance de $L_{\mathbb{M}}$ et $L_{\mathbb{G}}$, hauteur asymptotique

Valeurs exactes

(Σ, D)	$\rho_{L_{\mathbb{M}}}^{-1}$	$\lambda_{\mathbb{M}}$	$\rho_{L_{\mathbb{G}}}^{-1}$	$\lambda_{\mathbb{G}}$
K_n	n	1	2n-1	1
	1	$3/4$	1	$3/4$
	1	$11/18$	1	$11/18$
	2	1	3	1
	$(\sqrt{5}+3)/2$	$(7+\sqrt{5})/10$	$\sqrt{5}+2$	$(25-3\sqrt{5})/20$
	1	$25/48$	1	$25/48$
	2	1	3	1
	2	$3/4$	3	$3/4$

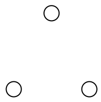
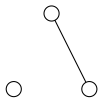
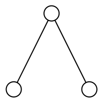
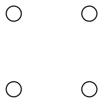
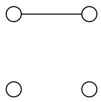
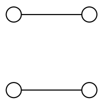
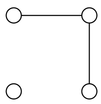
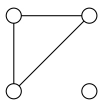
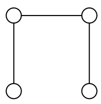
(Σ, D)	$\rho_{L_{\mathbb{M}}}^{-1}$	$\lambda_{\mathbb{M}}$	$\rho_{L_{\mathbb{G}}}^{-1}$	$\lambda_{\mathbb{G}}$
	$(\sqrt{5}+3)/2$	$(7+\sqrt{5})/10$	$\sqrt{5}+2$	$(25-3\sqrt{5})/20$
	3	1	5	1
	3	$19/22$	5	$379/444$
	3,155...	0,873...	5,291...	0,860...
	$2+\sqrt{2}$	$(6+\sqrt{2})/8$	$3+2\sqrt{2}$	$(41+\sqrt{2})/46$
	$2+\sqrt{2}$	$(11+\sqrt{2})/14$	$3+2\sqrt{2}$	$7/8$
	$2+\sqrt{3}$	$(4+\sqrt{3})/6$	$2\sqrt{3}+3$	$(9-3\sqrt{3})/4$

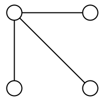
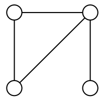
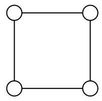
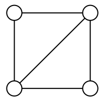
Remarque 2.43 Pour le graphe dont les valeurs sont données sous forme décimale, les valeurs données dans le tableau sont approchées. En effet, les formules exactes seraient trop longues pour entrer dans le tableau. Les voici tout de même :

$$\begin{aligned}
\rho_{L_{\mathbb{M}}} &= \frac{1}{72}\left(72-12X-(9-\sqrt{93})X^2\right)\\
\lambda_{\mathbb{M}} &= \frac{1}{333}\left(193+(4+\frac{94}{93}\sqrt{93})X+(\frac{65}{6}-\frac{203}{186}\sqrt{93})X^2\right)\\
X &= (108+12\sqrt{93})^{1/3}.
\end{aligned}$$

$$\begin{aligned}
\rho_{L_{\mathbb{G}}} &= -\frac{1}{9}+(\frac{1}{9}-\frac{1}{484}\sqrt{93})Y+\frac{47}{4356}Y^2\\
\lambda_{\mathbb{G}} &= \frac{6}{13}-(\frac{7}{104}-\frac{43}{9672}\sqrt{93})Y+(\frac{29}{25168}+\frac{1919}{2340624}\sqrt{93})Y^2\\
Y &= (188+36\sqrt{93})^{1/3}.
\end{aligned}$$

Valeurs approchées

(Σ, D)	$\rho_{L_{\mathbb{M}}}^{-1}$	$\lambda_{\mathbb{M}}$	$\rho_{L_{\mathbb{G}}}^{-1}$	$\lambda_{\mathbb{G}}$
	1	0,75	1	0,75
	1	0,611	1	0,611
	2	1	3	1
	2,618	0,924	4,23	0,915
	1	0,521	1	0,521
	2	1	3	1
	2	0,75	3	0,75
	2,618	0,924	4,23	0,915
	3	1	5	1
	3	0,864	5	0,854

(Σ, D)	$\rho_{L_{\mathbb{M}}}^{-1}$	$\lambda_{\mathbb{M}}$	$\rho_{L_{\mathbb{G}}}^{-1}$	$\lambda_{\mathbb{G}}$
	3,155	0,873	5,291	0,860
	3,414	0,927	5,828	0,922
	3.414	0,887	5,828	0,875
	3,732	0,955	6,464	0,951

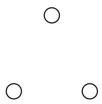
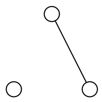
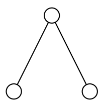
La table ci-dessus représente les singularités dominantes des séries génératrices des monoïdes de traces et des groupes de traces pour les exemples les plus simples. On a donné le graphe D pour chaque cas, le graphe $\tilde{D}$ se déduisant de ce dernier. On a aussi calculé les hauteurs asymptotiques des mots pour la probabilité uniforme sur les traces. On remarque que l'on a toujours, du moins pour ces exemples $\lambda_{\mathbb{G}} \leq \lambda_{\mathbb{M}}$ et l'implication

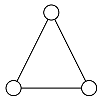
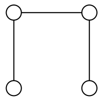
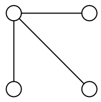
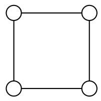
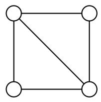
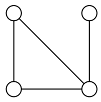
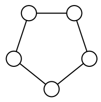
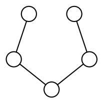
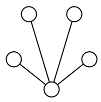
$$\lambda_{\mathbb{M}_1} \leq \lambda_{\mathbb{M}_2} \Rightarrow \lambda_{\mathbb{G}_1} \leq \lambda_{\mathbb{G}_2}.$$

On peut donc se demander si cette inégalité et cette implication sont toujours vraies.

2.7.3 Hauteur asymptotique avec probabilité uniforme sur les mots

Les valeurs étoilées sont les valeurs qu'on a pu calculer de façon exacte. Les autres ont été obtenues par simulation, à l'aide d'un programme écrit en Caml.

(Σ, D)	$\lambda_{\mathbb{M}}^*$	α	γ	$\lambda_{\mathbb{G}}^*$	$\lambda_{\mathbb{M}}^* - \lambda_{\mathbb{G}}^*$
	$1/3^*$	0,019	0,552	0^*	$1/3^*$
	$2/3^*$	0,341	0,98	$1/3^*$	$1/3^*$
	0,816	0,577	0,854	0,492	0,324

(Σ, D)	$\lambda^*_{\mathbb{M}}$	α	γ	$\lambda^*_{\mathbb{G}}$	$\lambda^*_{\mathbb{M}} - \lambda^*_{\mathbb{G}}$
	1^*	$2/3^*$	1^*	$2/3^*$	$1/3^*$
	0,691	0,604	0,745	0,450	0,241
	0,681	0,597	0,742	0,443	0,238
	0,789	0,678	0,806	0,546	0,243
	0,894	0,714	0,907	0,648	0,246
	0,802	0,660	0,846	0,559	0,243
	0,671	0,688	0,699	0,481	0,190
	0,596	0,619	0,652	0,404	0,192
	0,584	0,603	0,655	0,395	0,189

Deuxième partie

Performances des réseaux de Petri

Chapitre 3

Débit dans les réseaux de Petri

Dans ce chapitre, nous rappelons la définition et les premières propriétés des réseaux de Petri présentés de manière informelle dans le chapitre d'introduction. Nous étendons cette notion aux réseaux de Petri temporisés. Ces notions nous amènent naturellement à étudier le semi-anneau (max,plus) utilisé afin d'étudier le comportement temporisé des réseaux de Petri. Nous nous attachons en particulier au cas des graphes d'événements et des réseaux à choix libres 1-bornés, dont le comportement a été étudié ([11, 50]).

3.1 Réseaux de Petri

Les réseaux de Petri ont été introduits en 1962 par Carl A. Petri dans le but d'analyser les propriétés qualitatives de certains systèmes concurrents comportant des synchronisations. Ces réseaux permettent d'étudier de grandes classes de systèmes à événements discrets et ont donc été par la suite très abondamment étudiés ([83, 74, 34]) et utilisés pour diverses applications ([53]) : validation de systèmes, ateliers flexibles de production dans [35, 42], réseaux de communication dans [13, 17].

3.1.1 Définition des réseaux de Petri

Définition 3.1 (Réseau) *Un réseau est un triplet $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F})$ où :*
- *$\mathcal{P}$ et $\mathcal{T}$ sont des ensembles finis et disjoints ($\mathcal{P} \cap \mathcal{T} = \emptyset$) ;*
- *$\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$.*

Les éléments de $\mathcal{P}$ sont appelés des places et les éléments de $\mathcal{T}$ sont appelés des transitions.

On remarque qu'un réseau $\mathcal{N}$ est en fait un graphe biparti orienté. Soit $x \in \mathcal{P} \cup \mathcal{T}$. On note ${}^\bullet x$ l'ensemble des prédécesseurs de x et $x^\bullet$ l'ensemble de ses successeurs : ${}^\bullet x = \{y \in \mathcal{T} \cup \mathcal{P} \mid (y, x) \in \mathcal{F}\}$ et $x^\bullet = \{y \in \mathcal{T} \cup \mathcal{P} \mid (x, y) \in \mathcal{F}\}$. Une place ne peut avoir que des transitions pour prédécesseurs et successeurs. De même, une transition ne peut avoir que des places pour prédécesseurs et successeurs.

Soit $x \in \mathcal{P} \cup \mathcal{T}$. On appelle *cluster* de x, noté $[x]$, l'ensemble minimal de nœuds tels que $x \in [x]$ et si $p \in \mathcal{P} \cap [x]$, $p^\bullet \subset [x]$ et si $t \in \mathcal{T} \cap [x]$, ${}^\bullet t \subset [x]$. L'ensemble des clusters forme une partition des nœuds de $\mathcal{N}$.

Définition 3.2 (Réseau de Petri) *Un réseau de Petri est un couple $(\mathcal{N}, M)$ où $\mathcal{N}$ est un réseau et M une application $M : \mathcal{P} \to \mathbb{N}$ appelée marquage (initial) de $\mathcal{N}$.*

On dit qu'une place $p \in \mathcal{P}$ est marquée si $M(p) > 0$. Dans un réseau de Petri $(\mathcal{N}, M)$, $\mathcal{N}$ représente la structure du réseau et M son aspect dynamique : le marquage évolue selon la règle de tirs suivante :

- Une transition $t \in \mathcal{T}$ est dite *habilitée* si tous ses prédécesseurs sont marqués : $\forall p \in {}^\bullet t$, $M(p) > 0$;
- une transition habilitée peut être franchie (ou encore peut tirer) ;
- lorsqu'une transition est franchie, le marquage évolue et devient M', avec $\forall p \in \mathcal{P}$, $M'(p) = M(p) - \mathbf{1}_{^\bullet t}(p) + \mathbf{1}_{t^\bullet}(p)$.

Graphiquement, on représente les places par des cercles, les transitions par des rectangles, les éléments de $\mathcal{F}$ par des flèches reliant les places et les transitions. Le marquage est quant à lui représenté par les jetons dans les places (dans la place p, il y a $M(p)$ jetons). Une transition t est habilitée si chaque place en entrée de t contient au moins un jeton, et le tir d'une transition consiste à enlever un jeton dans chaque place d'entrée de la transition et à en ajouter un dans chaque place en sortie de la transition, comme l'illustre la figure 3.1. Le *degré d'habilitation* d'une transition t est $\min\{M(p) \mid p \in t^\bullet\}$.

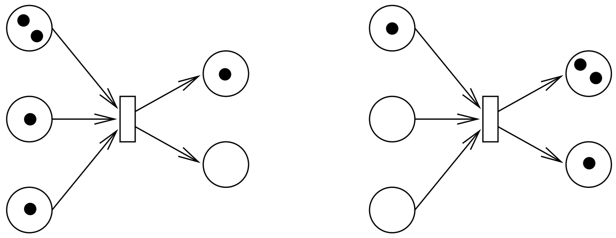


FIG. 3.1 – Tir d'une transition : à gauche, la transition est habilitée et à droite, le marquage après le tir de cette transition. À droite, la transition n'est plus habilitée.

Les réseaux de Petri sont utilisés pour modéliser des systèmes réels. Les transitions représentent les actions qui peuvent être effectuées. Le tir d'une transition équivaut alors à l'exécution de l'action correspondante. Les places, elles, représentent les ressources nécessaires ou les conditions devant être remplies pour l'exécution d'une action. La présence de jetons dans celles-ci indique le nombre de ressources disponibles ou qu'une condition est satisfaite. Le tir d'une transition consomme les ressources en amont de cette transition et produit les ressources en sortie de la transition.

Si une transition t a plusieurs places en entrée, il faut alors, pour pouvoir la franchir, la présence simultanée d'au moins un jeton dans chaque place d'entrée de t. On dit alors que t est une transition de synchronisation. Si une place p a plusieurs transitions en sortie, alors chaque jeton peut servir à franchir une seule de ces transitions. On parle alors de place de choix ou de conflit. On dit aussi que les transitions en sortie de p sont conflictuelles. Les réseaux de Petri peuvent aussi modéliser la concurrence, ou simultanéité, de plusieurs actions : plusieurs transitions habilitées n'ayant pas de place d'entrée en commun peuvent être franchies indépendamment les unes des autres, et l'ordre de tir n'a pas d'importance.

Exemple 3.3 La figure 3.2 représente un premier exemple de réseau de Petri. Les places a et b sont initialement marquées. Donc les transitions 1 et 3 sont habilitées. Si la transition 3 est tirée, alors, la place a n'est plus marquée, mais la place d le devient. Ensuite, seule la transition 4 est habilitée et on revient au marquage initial en tirant cette transition.

La transition 1 a plusieurs places en entrée. C'est donc une transition de synchronisation. La place a a plusieurs transitions en sortie. C'est donc une place de choix, ou de conflit, et les transitions 1 et 3 sont conflictuelles.

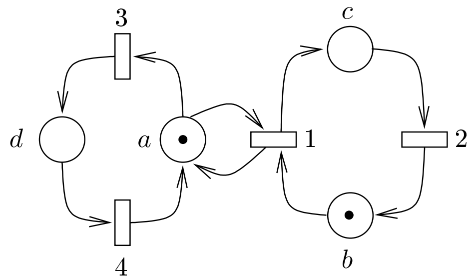


FIG. 3.2 – Un premier réseau de Petri.

3.1.2 Propriétés des réseaux de Petri

Si, à partir du marquage M on obtient le marquage M' en tirant la transition t, alors on note $M \xrightarrow{t} M'$. Soit $\sigma = t_1 \dots t_n \in \mathcal{T}^*$ une suite finie de transitions. Cette suite est dite admissible à partir de M s'il existe une suite de marquages $M_0, \dots, M_n$ telle que $M = M_0 \xrightarrow{t_1} M_1 \dots M_{n-1} \xrightarrow{t_n} M_n$. On note alors $M \xrightarrow{\sigma}$ ou $M \xrightarrow{\sigma} M'$ si $M' = M_n$ est le marquage obtenu.

Un marquage M' est *atteignable* à partir de M s'il existe une suite de tirs admissible $\sigma \in \mathcal{T}^*$ telle que $M \xrightarrow{\sigma} M'$. On note alors $M \to M'$ et $\mathcal{R}(M) = \{M' \in \mathbb{N}^{\mathcal{P}} \mid M \to M'\}$ l'ensemble des marquages atteignables à partir de M.

Définition 3.4 (Vivacité) *Un réseau de Petri $(\mathcal{N}, M)$ est dit vivant si pour toute transition $t \in \mathcal{T}$ et tout marquage atteignable $M_1 \in \mathcal{R}(M)$ il existe un marquage $M_2 \in \mathcal{R}(M_1)$ qui habilite t.*

Définition 3.5 (Blocage) *Un blocage de $(\mathcal{N}, M)$ est un marquage atteignable qui n'habilite aucune transition.*

Il est facile de voir que si un réseau de Petri a un blocage, alors le réseau n'est pas vivant. Mais si un réseau de Petri ne contient pas de blocage, il n'est pas nécessairement vivant.

Définition 3.6 (Borne) *Soit $(\mathcal{N}, M)$ un réseau de Petri et $p \in \mathcal{P}$ une place de ce réseau. On dit que p est b-bornée si pour tout marquage atteignable M', $M'(p) \leq b$. Le réseau est b-borné si toutes ses places le sont. Plus généralement, on dit qu'un réseau est borné s'il existe un b tel que ce réseau est b-borné.*

Toutes ces notions dépendent fortement du marquage initial du réseau de Petri. Mais on peut définir la vivacité structurelle qui ne dépend que de la structure du réseau : on dit qu'un réseau $\mathcal{N}$ est structurellement vivant s'il existe un marquage M tel que $(\mathcal{N}, M)$ est vivant. Si pour un réseau $\mathcal{N}$, il existe un marquage M tel que $(\mathcal{N}, M)$ est vivant et borné, alors il est dit bien formé.

Proposition 3.7 ([34],Théorème 2.25) *Un réseau bien formé est fortement connexe.*

Exemple 3.8 Reprenons l'exemple de la figure 3.2. L'ensemble des marquages atteignables est $\{(a, b), (a, c), (b, d), (c, d)\}$. On voit alors que le réseau est 1-borné. Le réseau est vivant et ne possède donc pas de marquage de blocage.

Savoir si un réseau de Petri a telle ou telle propriété définie ci-dessous est très complexe dans le cas général : savoir si un réseau de Petri est b-borné est PSPACE-complet, savoir si un réseau de Petri est vivant nécessite un espace au moins exponentiel ([41]). Ainsi, l'analyse des réseaux de Petri est très complexe dans le cas général, et on se restreint souvent à des sous-classes de réseaux plus faciles à analyser.

3.1.3 Sous-classes de réseaux de Petri

Définition 3.9 (Machine d'états) *Un réseau (de Petri) $\mathcal{N}$ $((\mathcal{N}, M))$ est une machine d'états si pour toute transition $t \in \mathcal{T}$, $|^\bullet t| = |t^\bullet| = 1$: chaque transition a un unique prédécesseur et un unique successeur.*

Les machines d'états modélisent des systèmes pour lesquels il n'y a pas de synchronisation, ni de « fork ». Le réseau de Petri de la figure 3.3 est une machine d'états qui modélise le comportement d'un distributeur automatique.

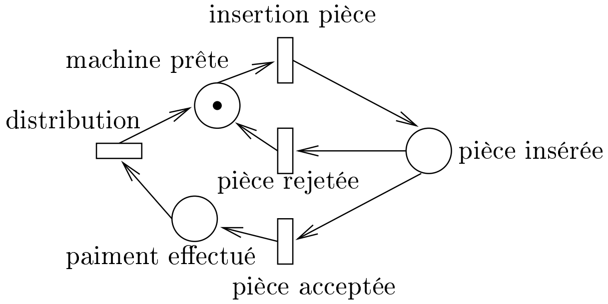


FIG. 3.3 – Machine d'états.

Proposition 3.10 ([34], théorèmes 3.3 et 3.5) *Une machine d'états fortement connexe est vivante si et seulement si son marquage initial n'est pas nul. Une machine d'états fortement connexe est bornée.*

Définition 3.11 (Graphes d'événements) *Un réseau (de Petri) $\mathcal{N}$ $((\mathcal{N}, M))$ est un graphe d'événements si pour toute place $p \in \mathcal{P}$, $|^\bullet p| = |p^\bullet| = 1$: chaque place a un unique prédécesseur et un unique successeur.*

Les graphes d'événements modélisent des systèmes pour lesquels il n'y a pas de conflit, mais il peut y avoir des synchronisations. Le réseau de Petri de la figure 3.4 est un graphe d'événements qui modélise le comportement de deux files d'attente en série.

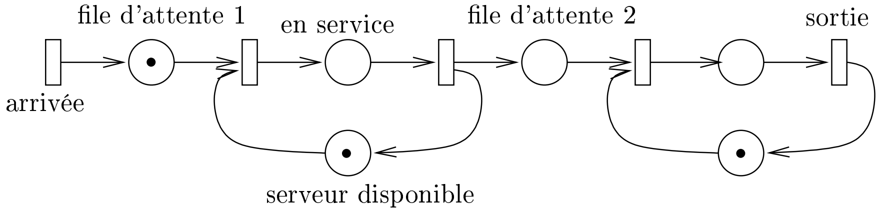


FIG. 3.4 – Graphe d'événements.

Proposition 3.12 ([34], théorèmes 3.15 et 3.18) *Un graphe d'événements fortement connexe est borné et vivant si et seulement si tous ses circuits sont marqués. Un graphe d'événements fortement connexe est borné.*

Définition 3.13 (Réseaux à choix libres) *Un réseau (de Petri) $\mathcal{N}$ $((\mathcal{N}, M))$ est un réseau à choix libres si pour toute place $p \in \mathcal{P}$, $|p^\bullet| > 1 \Rightarrow \forall t \in p^\bullet$, $|^\bullet t| = 1$: si une place a plusieurs transitions en sortie, alors chacune de ces transitions n'a que cette place en entrée.*

Dans les réseaux à choix libres, il peut y avoir des synchronisations et des conflits, mais ces deux phénomènes n'ont jamais lieu en même temps : une transition ne peut pas être à la fois de synchronisation et conflictuelle. Cette sous-classe de réseaux de Petri englobe les deux précédentes. La figure 3.5 montre un réseau à choix libres qui n'est ni une machine d'états ni un graphe d'événements.

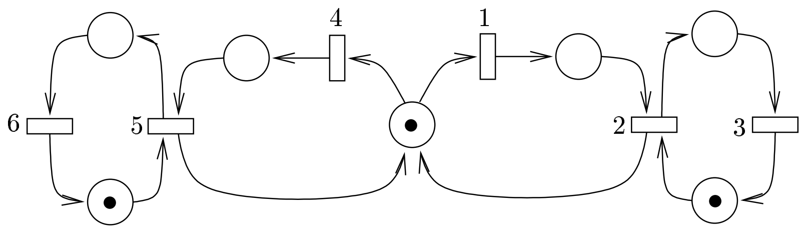


FIG. 3.5 – Réseau à choix libres.

Remarquons aussi que les clusters d'un réseau à choix libres contiennent soit une seule transition, soit une seule place. De plus, si une transition d'un cluster est habilitée, toutes les transitions de ce cluster sont habilitées.

Il n'existe pas de propriété aussi simple que pour les machines d'états et les graphes d'événements pour dire si un réseau à choix libres est vivant ou borné. Par exemple, savoir si un réseau à choix libres est vivant est NP-complet ([40]).

3.1.4 Analyse des réseaux de Petri

Nous présentons maintenant des outils permettant d'analyser les réseaux de Petri.

Graphe des marquages

Une première méthode consiste à étudier l'ensemble des marquages atteignables. Soit $(\mathcal{N}, M)$ un réseau de Petri. Le *graphe des marquages* associé à ce réseau est appelé $G_{(\mathcal{N},M)}$. L'ensemble de ses nœuds est $\mathcal{R}(M)$, l'ensemble des marquages atteignables à partir de M, et il y a un arc de M_1 à M_2 s'il existe une transition $t \in \mathcal{T}$ telle que $M_1 \xrightarrow{t} M_2$.

Le graphe des marquages d'un réseau est fini si et seulement si celui-ci est borné, et un blocage est un marquage sans arc sortant.

Exemple 3.14 Reprenons à nouveau le réseau de la figure 3.2. Son graphe des marquages est donné figure 3.6. Le graphe des marquages est fini, puisque le réseau est borné.

Cette méthode n'est pas assez précise, ni assez efficace : l'étude du caractère borné d'un réseau de Petri équivaut à la construction d'un graphe, dont la taille (s'il est fini) peut être exponentielle en la taille du réseau. De plus, l'étude de la vivacité du réseau de Petri demande alors une analyse précise de ce graphe.

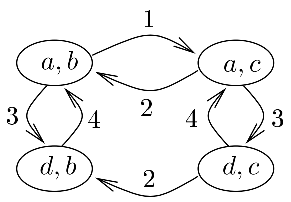


FIG. 3.6 – Graphe des marquages du réseau de Petri de la figure 3.2.

Analyse algébrique

Nous présentons maintenant quelques méthodes algébriques pour l'étude des réseaux de Petri.

Définition 3.15 (Matrice d'incidence) *Soit $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F})$ un réseau. La matrice d'incidence $N \in \mathbb{Z}^{\mathcal{P}\times\mathcal{T}}$ associée à $\mathcal{N}$ est définie par : $N_{p,t} = \mathbf{1}_{t^\bullet}(p) - \mathbf{1}_{^\bullet t}(p)$.*

Le coefficient $N_{p,t}$ représente la manière dont le marquage change dans la place p lorsque la transition t est franchie. Le marquage peut être représenté par un vecteur colonne de $\mathbb{N}^{\mathcal{P}}$. Soit $\sigma \in \mathcal{T}^*$ une suite finie de transitions. On appelle *vecteur de Parikh* de σ, noté $\vec{\sigma}$, l'image commutative de σ : $(\vec{\sigma})_{t\in\mathcal{T}} = (|\sigma|_t)_{t\in\mathcal{T}}$ ($|\sigma|_t$ est le nombre d'occurrences de t dans σ). On peut le voir comme un vecteur ligne.

Nous pouvons maintenant énoncer un premier lemme.

Lemme 3.16 (Équation du marquage, [34], lemme 2.12) *Pour toute suite finie $\sigma \in \mathcal{T}^*$ telle que $M \xrightarrow{\sigma} M'$ d'un réseau $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F})$ de matrice d'incidence N, on a*

$$M' = M + N.\vec{\sigma}.$$

Le marquage final ne dépend donc pas de l'ordre dans lequel sont franchies les transitions, pour deux suites admissibles ayant le même vecteur de Parikh.

Exemple 3.17 La matrice d'incidence du réseau de la figure 3.2 est

$$N = \begin{pmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}.$$

P-invariants et T-invariants

Définition 3.18 (P-invariant) *Un P-invariant $X \in \mathbb{Z}^{1\times\mathcal{P}}$ est une solution non nulle de $X.N = 0$. Un P-invariant est positif si pour tout $p \in \mathcal{P}$, $X_p > 0$. Un P-invariant est semi-positif si pour tout $p \in \mathcal{P}$, $X(p) \geq 0$ et $X \neq 0$.*

Voici maintenant une condition suffisante pour qu'un réseau de Petri soit borné.

Proposition 3.19 ([34], théorème 2.31) *Soit $(\mathcal{N}, M)$ un réseau de Petri. Si sa matrice d'incidence N a un P-invariant positif, alors le réseau $(\mathcal{N}, M)$ est borné.*

On remarque que cette condition ne dépend pas du marquage initial.

Définition 3.20 (T-invariant) *Un T-invariant $X \in \mathbb{Z}^{\mathcal{T}}$ est une solution non nulle de $N.X = 0$. Un T-invariant est positif si pour tout $t \in \mathcal{T}$, $X_t > 0$. Un T-invariant est semi-positif si pour tout $t \in \mathcal{T}$, $X(t) \geq 0$ et $X \neq 0$.*

Proposition 3.21 (Propriété fondamentale, [34], proposition 2.37) *Soit σ une suite admissible de transitions à partir de M. Le vecteur de Parikh de σ, $\vec{\sigma}$ est un T-invariant si et seulement si $M \xrightarrow{\sigma} M$.*

Une conséquence de cette propriété est :

Lemme 3.22 ([34], théorème 2.38) *Tout réseau bien formé a un T-invariant positif.*

Nous énonçons maintenant des lemmes valables aussi bien pour les P-invariants et les T-invariants. Le support d'un invariant I est noté $\langle I \rangle$.

Proposition 3.23 ([34], proposition 2.41) *Tout invariant semi-positif I satisfait ${}^\bullet\langle I \rangle = \langle I \rangle^\bullet$.*

Définition 3.24 (Invariants minimaux) *Un invariant semi-positif I est minimal s'il n'existe pas d'invariant semi-positif J tel que $\langle J \rangle \subset \langle I \rangle$.*

Proposition 3.25 ([34], théorème 2.43)

1. *Tout invariant semi-positif est la somme d'invariants minimaux.*
2. *Si un réseau a un invariant positif, alors tout invariant est combinaison linéaire d'invariants minimaux.*

Les T-invariants des graphes d'événements sont de la forme $(x, \ldots, x)^T$, et les P-invariants des machines d'états sont de la forme $(x, \ldots, x)$.

Nous allons étudier les propriétés des réseaux à choix libres. En effet, ce sont ces réseaux qui vont nous intéresser dans la suite.

Siphons et trappes Un *siphon* est un ensemble de places R tel que ${}^\bullet R \subset R^\bullet$. Si un siphon ne contient pas de jeton, alors au cours de l'évolution du réseau, il ne contiendra jamais de jeton. En effet, si le siphon R est vide, pour qu'un jeton apparaisse dans R, il faut qu'une transition en entrée d'une place de R, donc de ${}^\bullet R$, soit franchie. Or ${}^\bullet R \subseteq R^\bullet$. Donc cette transition est aussi en sortie d'une place de R. Puisqu'on a supposé R initialement vide, cette transition n'est pas habilitée. Donc R ne peut pas être marquée.

Une *trappe* est un ensemble de places R tel que $R^\bullet \subset {}^\bullet R$. Si une trappe est initialement marquée, alors elle le reste. En effet, quand une transition de $R^\bullet$ est franchie, cette transition appartient aussi à ${}^\bullet R$, et donc des nouveaux jetons apparaissent dans R.

Le théorème de Commoner ([34], partie 4.3) donne une condition nécessaire et suffisante pour établir la vivacité d'un réseau de Petri à choix libres.

Théorème 3.26 (Commoner, [34], partie 4.3) *Un réseau de Petri $(\mathcal{N}, M)$ à choix libres est vivant si et seulement si tout siphon non trivial contient une trappe initialement marquée.*

T-composantes et P-composantes Une *T-composante* d'un réseau $\mathcal{N}$ est un sous-réseau $\mathcal{N}'$ non vide d'ensemble de nœuds X tel que pour toute transition t de X, ${}^\bullet t \cup t^\bullet \subset X$ et $\mathcal{N}'$ est un graphe d'événements fortement connexe. Un réseau $\mathcal{N}$ est couvert par des T-composantes s'il existe un ensemble de T-composantes de $\mathcal{N}$ tel que toutes les transitions de $\mathcal{N}$ appartiennent à au moins une T-composante de cet ensemble. Les notions de T-composante et de T-invariant sont très proches. En effet, si T_1 est l'ensemble des transitions d'une T-composante, alors χ_{T_1} est un T-invariant minimal. Réciproquement, si J est un T-invariant minimal, il existe une T-composante dont l'ensemble des transitions est le support de J.

Théorème 3.27 ([34], théorème 5.8) *Un réseau à choix libres bien formé est couvert par des T-composantes.*

Une *P-composante* d'un réseau $\mathcal{N}$ est un sous-réseau $\mathcal{N}'$ non vide d'ensemble de nœuds X tel que pour toute place p de X, ${}^\bullet p \cup p^\bullet \subset X$ et $\mathcal{N}'$ est une machine d'états fortement connexe. Un réseau $\mathcal{N}$ est couvert par des P-composantes s'il existe un ensemble de P-composantes de $\mathcal{N}$ tel que toutes les places de $\mathcal{N}$ appartiennent à au moins une P-composante de cet ensemble.

Théorème 3.28 ([34], théorèmes 5.6 et 5.11) *Un réseau à choix libres bien formé est couvert par des P-composantes. Un réseau de Petri à choix libres vivant 1-borné est couvert par des P-composantes 1-bornées.*

Marquage de blocage Soit $(\mathcal{N}, M)$ un réseau de Petri et soit t une transition de $\mathcal{N}$. Un *marquage de blocage* de t est un marquage atteignable pour lequel la seule transition habilitée est t.

Théorème 3.29 ([52], théorème 3.1) *Soient $(\mathcal{N}, M)$ un réseau de Petri à choix libres vivant borné fortement connexe et t une transition non conflictuelle de $\mathcal{N}$. Il existe un unique marquage de blocage de t, M_t, qui est atteignable à partir de tout autre marquage atteignable et sans franchir la transition t.*

On peut étendre ce théorème aux transitions conflictuelles : on bloque alors toutes les transitions d'un cluster, et on obtient un unique marquage où seules ces transitions sont habilitées. Par abus de langage, pour une transition conflictuelle, on dit que son marquage de blocage est ce marquage.

3.2 Réseaux de Petri temporisés

L'analyse des réseaux de Petri permet d'étudier le comportement logique des systèmes à événements discrets. Pour une analyse quantitative (performance), l'introduction d'une dimension temporelle à ces réseaux est donc nécessaire. Deux approches différentes ont été introduites. La première est celle des réseaux de Petri temporels, qui se rapprochent des automates temporisés : le tir d'une transition a lieu dans un intervalle de temps. Ces réseaux sont plutôt utilisés pour la vérification de systèmes ([29]). La deuxième approche, qui est l'objet de notre étude, est celle des réseaux de Petri temporisés. Il existe plusieurs manières d'introduire le temps dans ces réseaux, les différentes composantes du réseau pouvant être temporisées : les arcs, les places, les transitions et les jetons. Ici, nous considérons sans perte de généralité que seules les transitions sont temporisées. Il nous faut aussi savoir quelle est la transition qui va être franchie lorsqu'il y a un choix à effectuer. Il faut alors définir une *politique de résolution de conflits*. Commençons tout d'abord par définir les réseaux de Petri temporisés.

On note $(\mathcal{N}, M, \tau)$ un réseau de Petri temporisé, avec $(\mathcal{N}, M)$ un réseau de Petri et $\tau \in \mathbb{R}_+^{\mathcal{T}}$ le vecteur des temporisations. Le franchissement d'une transition a qui devient habilitée au temps t de temporisation τ_a se fait de la manière suivante :

- à l'instant t, le franchissement commence ;
- à l'instant $t + \tau_a$, le franchissement termine. Si la transition a est toujours habilitée, la transition a est franchie à cet instant en suivant la même règle de tir que celle déjà décrite.

Dans le cas où plusieurs transitions en conflit pourraient être franchies au même instant, une seule de ces transitions sera effectivement franchie, un jeton pouvant être utilisé pour le franchissement d'une seule transition. Nous verrons comment se fait ce choix un peu plus loin.

Nous avons défini le comportement du réseau de Petri localement, pour le tir d'une transition. Il faut maintenant définir le comportement des autres transitions lorsqu'une transition a tiré : que deviennent les temps de tirs des transitions qui ont commencé à tirer ? Si une transition devient non habilitée, elle annule alors son tir, et le temps de tir commencé est oublié. Nous supposerons que les autres transitions continuent leur tir, puisqu'elles sont toujours habilitées.

Si $(\mathcal{N}, M)$ est un réseau de Petri vivant (resp. k-borné), alors pour $\tau \in \mathbb{R}_+^{\mathcal{T}}$, $(\mathcal{N}, M, \tau)$ est un réseau de Petri temporisé vivant (resp. k-borné).

On peut aussi définir les réseaux de Petri *stochastiques*, qui sont des réseaux de Petri temporisés, mais dont les temporisations sont des variables aléatoires positives : $(\mathcal{N}, M, \tau)$ est un réseau de Petri stochastique si $(\mathcal{N}, M)$ est un réseau de Petri et $\tau = (\tau_t)_{t \in \mathcal{T}}$, avec $\tau_t = (\tau_t(n))_{n \in \mathbb{N}}$ est une suite de variables aléatoires. La temporisation de la transition t lorsqu'elle est franchie pour la n-ième fois est $\tau_t(n)$.

Les réseaux de Petri temporisés ont commencé à être étudiés vers la fin des années 70. La première classe de réseaux de Petri temporisés à avoir été étudiée est la classe des graphes d'événements, grâce à l'algèbre (max,plus). Un ouvrage de référence à ce sujet est [11]. Les graphes d'événements avec des temporisations non constantes ont aussi été étudiés dans [8, 12] par exemple. Le terme de réseau de Petri stochastique (*stochastic Petri net*) a été introduit par Molloy pour désigner les réseaux de Petri avec des temporisations exponentiellement distribuées. Ces réseaux particuliers ont beaucoup été étudiés, ainsi qu'une extension, où l'on autorise les temporisations nulles ([4, 3]). On peut encore généraliser ces réseaux en introduisant des distributions phase-type [58], ce qui génère un accroissement de la complexité des problèmes. Une manière de les résoudre est de trouver une bonne décomposition du réseau. D'autre part, les réseaux de Petri avec des temporisations plus générales ont aussi été étudiés sous le même nom de *stochastic Petri nets* dans [60]. L'existence d'un comportement stable y est étudiée via des méthodes probabilistes, sous réserve de l'existence de « points de régénérescence », pour lesquels le comportement du réseau est bien défini entre deux passages en un tel point. Des méthodes de simulations y sont aussi présentées.

Dans ce mémoire, nous utiliserons le terme de réseau de Petri stochastiques pour des temporisations avec des distributions générales. Lorsque nous utiliserons des distributions exponentielles, nous le préciserons.

Définissons maintenant différentes politiques de résolution des conflits.

3.2.1 Différentes politiques de résolution des conflits

Afin de résoudre les conflits dans les réseaux de Petri, lorsqu'une place a plusieurs transitions en sortie, on définit des politiques de routages : ces politiques définissent la transition qui doit être franchie quand un jeton arrive dans une place conflictuelle. Présentons tout d'abord les quatre politiques de routage auxquelles nous allons nous intéresser.

Soit $(\mathcal{N}, M, \tau)$ un réseau à choix libres avec des temporisations.

La compétition (race policy)

La *race policy*, ou politique de compétition, est définie comme suit. Lorsqu'un jeton arrive dans une place conflictuelle p, toutes les transitions $a \in p^\bullet$ commencent leur tir. La transition qui finit la première son tir est effectivement franchie avec ce jeton (elle a gagné la course), alors que les autres transitions abandonnent leur tir. La probabilité que la transition $a \in p^\bullet$ soit franchie pour le n-ième jeton arrivant dans la place p est

$$\mathbb{P}(\tau_a(n) = \min_{a' \in p^\bullet} \{\tau_{a'}(n)\}).$$

Comme le réseau est à choix libres, si $a \in p^\bullet$ est habilitée, alors toute transition $a' \in p^\bullet$ est habilitée. On note $\mathcal{N}_{\text{race}}$ le réseau induit par $\mathcal{N}$ et obéissant à la race policy. Si k transitions en compétition terminent leur tir au même instant, alors la transition effectivement franchie est choisie aléatoirement, pour chaque transition t, la probabilité d'être choisie est $1/k$.

Si les temporisations sont constantes, la politique de compétition n'a pas grand intérêt : c'est toujours la même transition qui est choisie.

Routage Bernoulli

Un réseau de Petri routé est un 5-uplet $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M, u)$ où $(\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$ est un réseau de Petri et $u = (u_p)_{p \in \mathcal{P}}$. Pour toute place p, $u_p \in (p^\bullet)^{\mathbb{N}^*}$, est la fonction de routage de p. Elle indique, par $u_p(n)$, pour le n-ième jeton arrivant en p quelle transition il va franchir.

Le routage Bernoulli est une classe de routage telle que pour chaque place p, la fonction de routage $(u_p(n))_n$ est une suite de variables aléatoires indépendantes et identiquement distribuées (d'où le nom de routage Bernoulli). On suppose de plus que toutes les suites $(u_p)_p$ sont mutuellement indépendantes.

Routage périodique

Le routage périodique est une classe de routages déterministes telle que pour toute place p, la fonction de routage u_p est périodique. On note d_p la période de la fonction u_p.

Les routages 0-1 sont un cas particulier des routages périodiques : ce sont des routages dont toutes les fonctions u_p sont constantes (donc de période 1).

Routages extrémaux

Le routage optimal donne le débit le plus élevé, sous réserve d'existence, dans une certaine classe de routages. Le routage le pire est un routage qui donne le débit le moins élevé.

3.2.2 Débit dans les réseaux de Petri temporisés

Soit $(\mathcal{N}, M, \tau)$ un réseau de Petri temporisé (temporisations constantes ou stochastiques) et $\sigma = \sigma(1)\sigma(2) \ldots \in \mathcal{T}^{\mathbb{N}}$ une suite de tirs admissible infinie. Posons $\sigma_n = \sigma(1) \ldots \sigma(n)$. On regarde l'évolution temporisée du réseau à partir de l'instant initial et associée à la suite de tirs σ. On appelle *activité* de σ le nombre de tirs asymptotique moyen par unité de temps, notée $Act(\sigma)$. Cette activité vaut

$$Act(\sigma) = \liminf_{n \to \infty} \frac{n}{d(n)}, \tag{3.1}$$

où $d(n)$ est le premier instant de complétion de tous les tirs de σ_n.

Dans certains cas, on voudra une définition plus flexible et plus générale. Alors, on pondère l'activité de chaque transition : un poids $\alpha_t \in \mathbb{R}_+$ est associé à chaque transition t. On pose $\alpha = (\alpha_t)_{t\in\mathcal{T}}$. On suppose que tous les poids ne sont pas nuls ($\alpha \neq (0, \dots, 0)$). On définit alors le débit associé à la suite σ et au poids α, noté $D(\sigma)$, comme

$$D(\sigma) = \liminf_{n\to\infty} \frac{\sum_{t\in\mathcal{T}} \alpha_t \overrightarrow{\sigma_n}_t}{d(n)}. \tag{3.2}$$

Si tous les poids sont égaux à 1, le débit est égal à l'activité. D'autre part, si $\alpha_t = 1$ et pour toutes les autres transitions t', $\alpha_{t'} = 0$, le débit mesure le taux de tirs de la transition t.

La notion de débit définie ici permet de modifier le réseau sans changer le débit. Par exemple, on peut remplacer une transition dont le temps de tir est $n \in \mathbb{N}$ et le poids α par n transitions de temps de tir 1 et de poids α/n sans changer le comportement du réseau ni son débit.

Le débit dans les réseaux de Petri *fluides* a été étudié dans [51, 82, 86, 31]. Dans ces réseaux, le nombre de jetons n'est plus nécessairement un entier, mais les jetons s'écoulent de manière continue à travers les transitions. Pour des proportions de routages fixées, le débit se calcule par un programme linéaire et l'optimisation des proportions de routages pour maximiser le débit aussi. Dans les prochains chapitres, nous nous intéressons aux réseaux de Petri non fluides, dont le comportement est plus difficile à appréhender.

Modélisation des graphes d'événements temporisés (temporisations constantes)

Soit $(\mathcal{N}, M, \tau)$ un graphe d'événements vivant temporisé fortement connexe. Soit t une transition de $\mathcal{N}$. La date de fin du n-ième franchissement de t est la date à laquelle il est passé au moins n jetons (y compris ceux du marquage initial) dans chaque place en entrée de t plus τ_t. C'est donc

$$X_t(n) = \max_{p\in{}^\bullet t} X_{{}^\bullet p}(n - M(p)) + \tau_t, \tag{3.3}$$

en identifiant ${}^\bullet p$ à son unique élément.

Soit m le nombre maximal de jetons d'une place dans le marquage initial. On définit les matrices A_i, $i \in \mathbb{N}_m$ par

$$(A_i)_{t't} = \begin{cases} \tau_t & \text{si } \exists p \in {}^\bullet t \cap t'^\bullet \text{ et } M(p) = i, \\ -\infty & \text{sinon.} \end{cases} \tag{3.4}$$

L'équation 3.3 devient alors dans la semi-algèbre (max,plus)

$$X(n) = \bigoplus_{i\in\{0,\dots,m\}} X(n-i) \otimes A_i. \tag{3.5}$$

Comme on a supposé le réseau vivant, il n'existe pas de circuit sans jetons (proposition 3.12), et la matrice A_0^* existe (proposition 1.3). On a alors

$$X(n) = \bigoplus_{i\in\mathbb{N}_m} X(n-i) \otimes A_i \otimes A_0^*. \tag{3.6}$$

On appelle cette équation *équation aux dateurs*, car elle définit les dates de tirs des transitions sous forme d'une équation de récurrence d'ordre m. On peut transformer cette équation en une équation de récurrence d'ordre 1 (matrice compagnon). On a alors $Y(n) = Y(n-1) \otimes A$, avec $Y(n) = (X(n), \dots, X(n+m-1))$ et

$$A = \begin{pmatrix} \mathcal{E}_n & \mathcal{E}_n & \cdots & A_m A_0^* \\ E_n & \mathcal{E}_n & \cdots & A_{m-1} A_0^* \\ \mathcal{E}_n & E_n & \mathcal{E}_n & \vdots \\ \ddots & \mathcal{E}_n & E_n & A_1 A_0^* \end{pmatrix}, \tag{3.7}$$

où $\mathcal{E}_n$ et E_n sont les matrices nulle et identité de dimension n dans l'algèbre (max,plus).

La matrice (max,plus) obtenue a une unique valeur propre, $\rho(A)$, qui est aussi égale au maximum, sur chacun des circuits du graphe d'événements du rapport entre la somme des temporisations des transitions du circuit et du nombre de jetons dans ce circuit. La valeur propre est le temps moyen entre deux tirs d'une même transition. Le débit d'une transition est alors l'inverse de la valeur propre de A. Notons que cette vitesse ne dépend pas de la transition du réseau pour des graphes d'événements fortement connexes.

Le débit vaut donc
$$D = \frac{\sum_{t \in \mathcal{T}} \alpha_t}{\rho(A)}.$$

Modélisation des réseaux à choix libres 1-bornés temporisés (temporisations constantes)

Nous allons maintenant voir comment modéliser le comportement temporisé d'un réseau à choix libres 1-borné par des tas de pièces (et des automates (max,plus)). Nous donnons ici les résultats de [50].

Soient $(\mathcal{N}, M, \tau)$ un réseau à choix libres 1-borné temporisé et $\sigma = \sigma(1) \dots \sigma(n) \in \mathcal{T}^*$ une suite de tirs admissibles.

Le tas de pièces associé à $\mathcal{N}$ est défini comme suit :
- les ressources sont les places : $\mathcal{R} = \mathcal{P}$;
- chaque transition a est une pièce dont les ressources sont $R(a) = {}^\bullet a \cup a^\bullet$;
- $\forall p \in R(a)$, $l(a, p) = 0$;
- $\forall p \in a^\bullet$, $u(a, p) = \tau_a$;
- $\forall p \in {}^\bullet a \setminus a^\bullet$, $u(a, p) = 0$.

Le temps d'exécution de la suite σ est alors la hauteur du tas défini lorsqu'on empile les pièces $\sigma(1), \dots, \sigma(n)$ dans cet ordre.

Lemme 3.30 *Soit H un empilement de pièces d'une suite admissible de tirs avec le modèle de tas de pièces défini comme ci-dessus. Si les temporisations sont strictement positives, le marquage du réseau se déduit du profil du tas de pièces.*

Démonstration. Un réseau à choix libres vivant 1-borné est couvert par des P-composantes 1-bornées et toutes ces P-composantes contiennent toujours un et un seul jeton. Si l'on regarde les colonnes du tas de pièces correspondant à une de ces P-composantes, alors le jeton se trouve dans la place correspondant à la colonne la plus haute. Ceci permet de retrouver le marquage. □

Pour une suite de tirs admissible infinie, on peut définir l'activité en fonction de la hauteur du tas de pièces : si $h(n)$ est la hauteur quand les n premières pièces ont été empilées, l'activité du réseau est alors
$$\liminf_n \frac{n}{h(n)}.$$

On peut diminuer le nombre de ressources pour un autre tas de pièces qui modélise le réseau de la menière suivante. Soit $S_1, \dots, S_k$ une couverture par des P-composantes 1-bornées. On définit un modèle de tas de pièces comme suit :

- les ressources sont $R = \{S_1, \dots, S_k\}$;
- pour chaque transition a, on définit une pièce, dont les ressources sont $R(a) = \{S_i \mid a \in S_i\}$;
- $l(a, S) = 0$ pour $S \in R(a)$ et $u(a, S) = \tau_a$ pour $S \in R(a)$.

Ce dernier tas de pièce a une représentation plus compacte en termes de tas de pièces, mais la proposition 3.30 ne s'applique plus.

Exemple 3.31 Reprenons l'exemple du réseau à choix libre de la figure 3.5. On peut représenter l'évolution de ce réseau par un tas de pièces à 3 colonnes et 6 pièces. Toutes les temporisations sont égales à 1, sauf celle de la transition 3, qui est égale à 2. Les transitions 1,4,2,5 occupent la colonne 2, les transitions 2 et 3 la colonne 3, et les transitions 5 et 6 la colonne 1. Si les choix $1, 4, 1, 1, 4$ sont effectués, on obtient l'empilement représenté en figure 3.7.

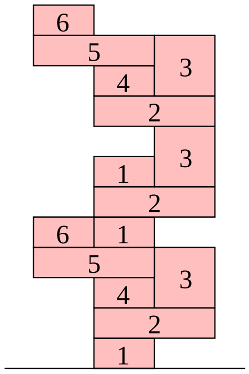


FIG. 3.7 – Évolution d'un réseau représenté par un tas de pièces.

*
* *

Dans ce chapitre, nous avons présenté les réseaux de Petri et défini la notion de débit. En outre, plusieurs manières de modéliser les réseaux de Petri par l'algèbre (max,plus) ont été abordées. Dans les chapitres suivants, nous allons étudier ce débit pour les réseaux à choix libres et chercher à optimiser ce débit. Dans le chapitre 4, nous considérons les réseaux avec des temporisations constantes, alors que dans les chapitres 5 et 6, les réseaux avec des temporisations stochastiques sont étudiés.

Chapitre 4

Débit extrémaux dans les réseaux de Petri temporisés

Dans le chapitre précédent, nous avons défini la notion de débit dans les réseaux de Petri. Nous étudions dans ce chapitre les réseaux de Petri avec des temporisations constantes. Nous nous posons deux sortes de questions : étant donné un réseau de Petri temporisé et un routage, peut-on calculer le débit ? Étant donné un réseau de Petri temporisé, pour quel routage le débit est-il maximal, minimal ? Un autre problème serait de savoir quelle est la forme des routages qui permettent de trouver les débits extrémaux.

En vue de répondre à ces questions, nous étudions plusieurs cas : intuitivement, on a tendance à penser que le débit minimal est obtenu en routant toujours les jetons dans les mêmes transitions, dans les parties les plus lentes du réseau. Nous verrons que ce n'est pas tout à fait vrai. Néanmoins, nous étudierons la classe de ces routages (routages 0-1). Ensuite, nous nous limitons aux réseaux avec des temporisations rationnelles. Nous montrons que les débits extrémaux sont obtenus par des routages périodiques. Mais les techniques développées ne permettent pas de connaître la forme du routage atteignant ces extrémaux. Enfin, nous étudions la classe des réseaux 1-bornés. En même temps que nous exhibons un réseau dont le débit minimal n'est pas atteint par un routage 0-1, nous donnons un algorithme efficace pour le calcul du débit minimal ainsi que le routage correspondant. Le routage minimal est atteint pour un routage périodique de petite période. Pour montrer ce résultat nous introduisons la notion de TT-invariant.

Les résultats obtenus dans ce chapitre sont publiés dans [22] et [20].

4.1 Routages 0-1 et réseaux sans choix

4.1.1 Les réseaux sans choix

On appelle *réseau sans choix* un réseau dont toutes les places ont une et une seule transition en sortie. Les graphes d'événements sont donc une sous-classe des réseaux sans choix, et les réseaux sans choix sont des réseaux à choix libres.

Proposition 4.1 *Un réseau sans choix fortement connexe est structurellement vivant.*

Démonstration. Soit $\mathcal{N}$ un réseau sans choix fortement connexe. Toute place a exactement une transition en sortie. Donc tout circuit de $\mathcal{N}$ est une trappe (l'ensemble des transitions en sortie des places d'un circuit est inclus dans l'ensemble des transitions en entrée de ces mêmes places).

Les siphons de $\mathcal{N}$ contiennent au moins un circuit (car l'ensemble des transitions en entrée des places d'un siphon sont incluses dans l'ensemble des transitions de sortie de ces places).

Définissons maintenant le marquage initial, M, tel que $M(p) = 1$ pour toute place p. Alors, tout siphon contient une trappe initialement marquée. Par le théorème de Commoner ([34, Section 4.3]), ce réseau de Petri est vivant. Donc $\mathcal{N}$ est structurellement vivant. □

Proposition 4.2 *Un réseau sans conflit fortement connexe, vivant et borné est un graphe d'événements.*

Démonstration. Soit $(\mathcal{N}, M)$ un réseau de Petri vivant, sans choix et borné. On veut montrer que chaque place de $\mathcal{N}$ a exactement une transition en entrée. Procédons par l'absurde, et supposons qu'il existe une place p ayant plusieurs transitions, $t_1, \ldots, t_k$, en entrée. Soit t la transition de sortie de p. Comme le réseau est borné et sans choix, on peut appliquer le théorème 3.29 du marquage de blocage à la transition t : il existe un unique marquage M_t atteignable tel que t est la seule transition habilitée. À partir de M_t, franchissons la transition t, et bloquons à nouveau cette transition. Le réseau évolue et atteint de nouveau le marquage M_t. La suite des transitions qui ont été franchies est unique à commutation près de certaines transitions car le réseau est sans choix ([52], lemme 4.4).

Clairement, une et une seule des transitions en entrée de p a été franchie : par unicité du marquage de blocage, le nombre de jetons en p doit être le même. Mais alors, si la transition t_1, par exemple, n'a pas été franchie, elle ne le sera jamais, même si l'on débloque t. Ainsi, le réseau n'est pas vivant, ce qui est une contradiction. Par conséquent, chaque place a exactement une transition en entrée, ce qui veut dire que c'est un graphe d'événements. □

La réciproque est bien évidement vraie : un graphe d'événements est par définition sans choix, et s'il est fortement connexe, il est toujours borné.

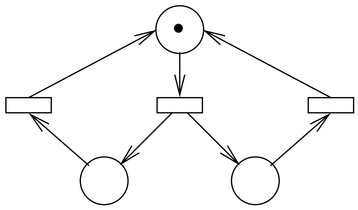


FIG. 4.1 – Exemple d'un réseau sans choix fortement connexe et non borné.

On peut donc regrouper ces deux propositions en un théorème :

Théorème 4.3 *Un réseau sans conflit fortement connexe est structurellement vivant. De plus, il est borné si et seulement si c'est un graphe d'événements.*

4.1.2 Routages 0-1

Partie vivante des réseaux avec routages 0-1

Soit $(\mathcal{N}, M, u)$ un réseau de Petri à choix libres vivant borné, où u est un routage 0-1. On obtient le réseau $\widetilde{\mathcal{N}}$ à partir de $\mathcal{N}$ en supprimant les transitions (et les arcs entrants et sortants de ces transitions) qui ne sont jamais choisies par ce routage. Le réseau $\widetilde{\mathcal{N}}$ obtenu peut ne plus être fortement connexe (des places n'ont plus de transitions en entrée), et donc ne plus être vivant.

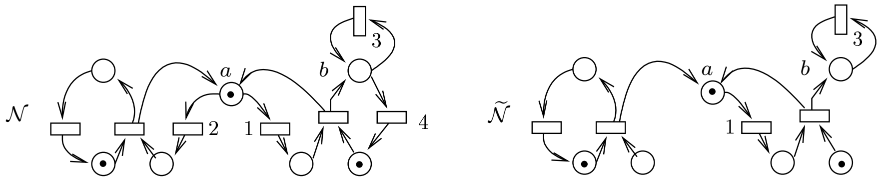


FIG. 4.2 – Le réseau $\widetilde{\mathcal{N}}$ est construit en enlevant toutes les transitions non choisies par u. Ici, u envoie tous les jetons de la place a vers la transition 1, et tous les jetons arrivant en b sont routées vers la transition 3. Les transitions 2 et 4 sont donc éliminées pour construire $\widetilde{\mathcal{N}}$.

Le réseau de Petri $(\widetilde{\mathcal{N}}, M)$ est par construction sans choix et borné, mais peut ne plus être vivant. Nous nous intéressons donc maintenant à la caractérisation de sa partie vivante. Cette partie vivante dépend a priori du marquage initial et du routage.

Le réseau $\widetilde{\mathcal{N}}$ peut être décomposé en ses composantes fortement connexes maximales. On distingue alors deux types de composantes fortement connexes, les composantes terminales, et les autres.

Lemme 4.4 *Les composantes fortement connexes maximales terminales de $\widetilde{\mathcal{N}}$ sont des T-composantes de $\mathcal{N}$. Les composantes non terminales sont des graphes d'événements, mais pas des T-composantes de $\widetilde{\mathcal{N}}$. Ces dernières ne peuvent pas appartenir à la partie vivante de $\widetilde{\mathcal{N}}$.*

Démonstration. Toutes les composantes fortement connexes maximales non triviales sont des réseaux sans choix fortement connexes. Comme le réseau initial est vivant borné (et comme c'est une propriété structurelle des réseaux à choix libres), toutes les composantes sont bornées. En appliquant le théorème 4.3, on déduit que ces composantes sont des graphes d'événements.

Soit C une composante fortement connexe non terminale. La composante C est non terminale et le réseau sans choix, donc il existe une transition $t \in C$ ayant une place de sortie $p \notin C$ (ce n'est donc pas une T-composante). Si C était incluse dans la partie vivante de $\widetilde{\mathcal{N}}$, la transition t pourrait être franchie infiniment souvent, et la place p recevoir un nombre infini de jetons, ce qui implique que le réseau initial serait non borné. Ainsi, toutes les composantes non terminales sont des graphes d'événements non vivants.

Soit C une composante fortement connexe terminale. On modifie le routage (ou allocation) comme suit : dans C, rien n'est modifié, et en dehors de C, l'allocation pointe vers C (c'est-à-dire qu'il existe un chemin de tout nœud vers C dans $\widetilde{\mathcal{N}}$). Dans le nouveau réseau obtenu, il n'y a qu'une composante fortement connexe terminale, qui est C. C'est la seule partie vivante possible, d'après ce qui précède. Comme le réseau initial est vivant, cette partie est vivante. C'est aussi une T-composante, car pour toute transition de C, toute place en entrée appartient à C (sinon, cette composante ne serait pas vivante car une place ne serait pas vivante), et toute place en sortie est aussi dans C (C est terminale). □

Nous avons besoin du lemme suivant pour pouvoir ensuite caractériser les parties vivantes du réseau.

Lemme 4.5 *Soit $(\mathcal{G}, M)$ un graphe d'événements vivant et 1-borné. Alors,*
- *si un jeton est enlevé, alors le réseau n'est plus vivant ;*
- *si un jeton est ajouté, alors le graphe d'événements n'est plus 1-borné.*

Démonstration. Soit p une place initialement marquée : $M(p) = 1$, et M' le marquage obtenu à partir de M en enlevant le jeton en p. Si $(\mathcal{G}, M)$ était vivant, il existerait une suite de tirs σ telle que $M' \xrightarrow{\sigma} M''$ et $M''(p) = 1$. Mais, cette séquence serait aussi admissible à partir de M, et dans le marquage obtenu, il y aurait deux jetons en p, ce qui est en contradiction avec le caractère 1-borné du graphe d'événements. Donc $(\mathcal{N}, M')$ n'est pas vivant.

Si un jeton est ajouté dans $(\mathcal{N}, M)$, et que le réseau obtenu $(\mathcal{N}, M')$ reste 1-borné, alors, on pourrait à nouveau enlever ce jeton, tout en laissant le réseau vivant (c'est le réseau initial). On déduit donc, en appliquant le premier point, que le réseau $(\mathcal{N}, M')$ n'est pas 1-borné. $\square$

Théorème 4.6 *Soit $(\mathcal{N}, M)$ un réseau à choix libres vivant borné, avec un routage 0-1. Soit $\widetilde{\mathcal{N}}$ obtenu à partir de $\mathcal{N}$ en enlevant les transitions qui ne sont par sélectionnées par le routage. La partie vivante de $(\widetilde{\mathcal{N}}, M)$ est une union disjointe non vide de graphe d'événements de $(\mathcal{N}, M)$, $(\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_k, M_k)$, où tous les $\mathcal{G}_i$ sont des T-composantes de $\mathcal{N}$. Si $(\mathcal{N}, M)$ est 1-borné ou si $k = 1$, alors l'ensemble des marquages atteignables $\mathcal{R}(\mathcal{G}_1, M_1), \ldots, \mathcal{R}(\mathcal{G}_k, M_k)$ ne dépendent pas du routage 0-1 qui fait que la partie vivante de $(\mathcal{N}, M)$ est $\mathcal{G}_1 \cup \ldots \cup \mathcal{G}_k$.*

Réciproquement, si $\mathcal{G}$ est une T-composante de $\mathcal{N}$, il existe un routage 0-1 tel que la partie vivante de $(\widetilde{\mathcal{N}}, M)$ est $(\mathcal{G}, M')$ pour un marquage M'.

Démonstration. La première partie de ce théorème est une conséquence directe du lemme 4.4 : la partie vivante de $(\mathcal{N}, M)$ est une union disjointe de graphe d'événements qui sont des T-composantes de $\mathcal{N}$.

Supposons que $k = 1$ et prouvons que $\mathcal{R}(\mathcal{G}_1, M_1)$ ne dépend pas du routage 0-1 qui fait de $\mathcal{G}_1$ la partie vivante de $\mathcal{N}$. Dans un graphe d'événements, l'ensemble des marquages atteignables ne dépend que du nombre de jetons dans chaque circuit. De plus, par unicité du marquage de blocage d'une transition de $\mathcal{G}_1$ dans $\mathcal{N}$, on déduit que le nombre de jetons dans un circuit de $\mathcal{G}_1$ ne dépend pas de M_1.

Supposons maintenant que $(\mathcal{N}, M)$ est 1-borné. Il s'ensuit que les graphes d'événements $(\mathcal{G}_i, M_i)$ sont 1-bornés. Regardons l'ensemble des marquages $\mathcal{R}(\mathcal{G}, M_i)$, et prouvons que c'est le même ensemble que si $\mathcal{G}_i$ était la partie vivante de $\mathcal{N}$. Prenons pour marquage M_i' le marquage de blocage d'une transition t de $\mathcal{G}_i$ dans $(\mathcal{G}_i, M_i)$, et pour M', le marquage de blocage de t dans $(\mathcal{N}, M)$. Nécessairement, il n'y a pas plus de jetons dans les circuits de $(\mathcal{G}_i, M_i')$ que dans $(\mathcal{N}, M')$. Mais, si dans un circuit de $\mathcal{G}_i$, il y a des jetons en plus, d'après le lemme 4.5, le réseau $(\mathcal{N}, M')$ ne serait pas 1-borné. Donc les marquages $\mathcal{R}(\mathcal{G}_i, M_i)$ sont les mêmes que ceux de $\mathcal{G}_i$ si c'était la seule T-composante vivante dans le réseau.

Fixons maintenant une T-composante $\mathcal{G}$ de $\mathcal{N}$. Soit C l'ensemble des clusters contenant une transition au moins de $\mathcal{G}$. On définit une allocation qui pointe vers C. Pour les clusters de C, on définit l'allocation vers la transition qui appartient à $\mathcal{G}$. On a donc définit une allocation totale, où tous les jetons sont dirigés vers $\mathcal{G}$. Avec le routage déduit de l'allocation, le réseau $\widetilde{\mathcal{N}}$ a une seule composante terminale la partie vivante de $\widetilde{\mathcal{N}}$ est précisément cette composante finale, $\mathcal{G}$. $\square$

Le lemme suivant va être utile pas la suite et se prouve grâce aux techniques développées dans ce paragraphe.

Un circuit élémentaire est un circuit pour lequel chaque nœud apparaît au plus une fois.

Lemme 4.7 *Dans un réseau à choix libres vivant borné, tout circuit élémentaire est inclus dans une T-composante du réseau.*

Démonstration. Soit c un circuit élémentaire de $\mathcal{N}$, un réseau à choix libres vivant borné. On choisit un routage 0-1 tel que toutes les transitions de c sont choisies, et pour toute place de choix p hors de c, la transition choisie est celle qui se trouve sur un plus court chemin de p à

c. Le réseau sans choix $\widetilde{\mathcal{N}}$ obtenu avec ce routage a une seule composante fortement connexe maximale terminale, par construction. De plus, cette composante contient c, ce qui prouve que c est incluse dans une T-composante. □

Débit dans les réseaux avec routages 0-1

Nous pouvons maintenant énoncer un résultat sur le calcul du débit dans le réseau avec des routages 0-1.

Théorème 4.8 *Soit $(\mathcal{N}, M)$ un réseau à choix libres vivant borné avec un routage 0-1. Supposons que la partie vivante est $(\mathcal{G}, M')$ où $\mathcal{G}$ est une seule T-composante, correspondant au T-invariant $J_\mathcal{G}$. Alors, le débit ne dépend pas du routage 0-1 utilisé pour que $\mathcal{G}$ soit la partie vivante de $\mathcal{N}$ et est égal à*

$$D_\mathcal{G} = \frac{\alpha^T J_\mathcal{G}}{\rho(\mathcal{G}, M)}. \tag{4.1}$$

Supposons que la partie vivante est $(\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_k, M_k)$ où $\mathcal{G}_i$ est une T-composante. Le débit est alors

$$D = \sum_{i=1}^{k} D_{\mathcal{G}_i} = \sum_{i=1}^{k} \frac{\alpha^T J_{\mathcal{G}_i}}{\rho(\mathcal{G}_i, M_i)}. \tag{4.2}$$

Si le réseau est 1-borné, alors le débit ne dépend que du routage qui fait de $\mathcal{G}_1, \ldots, \mathcal{G}_k$ la partie vivante de $\mathcal{N}$.

Démonstration. C'est une conséquence immédiate du débit dans les graphes d'événements et du théorème 4.6. Il suffit de remarquer que le débit dans un graphe d'événements ne dépend que du nombre de jetons dans chaque circuit. □

En particulier, nous avons le corollaire :

Corollaire 4.9 *Dans un réseau à choix libres 1-borné, le routage le pire parmi les routages 0-1 sont parmi ceux qui rendent une seule composante vivante.*

4.2 Calcul du débit pour les routages périodiques dans les réseaux à choix libres

4.2.1 Calcul du débit dans un réseau borné avec des temporisations rationnelles et un routage périodique

Considérons $(\mathcal{N}, M, \tau, u)$ un réseau à choix libres vivant borné temporisé routé avec des temporisations entières (on peut faire la même chose avec des temporisations rationnelles) et un routage périodique.

L'état du réseau de Petri au temps t peut être décrit par un triplet (M_t, R_t, R_t), où M_t est le marquage au temps t, R_t le vecteur du temps de tir restant des transitions ayant commencé leur tir, et U_t le routage courant dans les places de choix. Le nombre d'états est fini, car le réseau est borné, le routage périodique et les temporisations entières. Le nombre d'états est borné par $(k+1)^\mathcal{P} \times F^{k|\mathcal{T}|} \times L^{|\mathcal{P}|}$, où k est la borne du réseau de Petri (nombre maximal de jetons par place), F est le temps de tir maximal d'une transition et L est la période maximale du routage d'une place du réseau.

Puisque le comportement du réseau est déterministe, d'un état, on ne peut aller en un pas que vers un seul état. Le comportement du réseau est donc périodique. Ainsi, le débit existe et peut être calculé en regardant une période du réseau : c'est le rapport entre le nombre de transitions franchies et la longueur du cycle du graphe.

La complexité des calculs est très grande à la fois en temps et en espace car le nombre d'états peut être très élevé : le nombre des marquages peut être exponentiel en la taille du graphe et les temporisations ; la période des routages peut augmenter encore le nombre d'états.

4.2.2 Calcul du débit avec des temporisations réelles

Soit $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M, \tau, u)$ un réseau à choix libres vivant 1-borné temporisé routé où u est une fonction de routage périodique. On suppose dans un premier temps que le routage est équitable : pour toute place p, $u_p(\mathbb{N}) = p^\bullet$. Le réseau est donc vivant. On peut modéliser l'évolution au plus tôt des réseaux à choix libres 1-bornés par des tas de pièces. L'ordre partiel des tirs dans le réseau $(\mathcal{N}, M, u)$ non temporisé est entièrement déterminé par la fonction de routage (le réseau étant 1-borné, les jetons ne peuvent se dépasser, et un jeton dans une place va nécessairement servir au franchissement de la transition vers laquelle il est routé dès que cette transition est habilitée). Ainsi, l'ordre partiel de franchissement des transitions ne dépend pas des temporisations, et seule la hauteur des pièces empilées change en fonction des temporisations.

Plus précisément, soit $\mathcal{A}$ l'automate des tas de pièces modélisant le réseau 1-borné routé. Les états de l'automate sont $\mathcal{R}(M) \times \prod_{p \in \mathcal{P}} \mathbb{N}_{d_p}$, où pour tout $p \in \mathcal{P}$, d_p est la période de u_p. En effet, pour connaître les transitions franchissables du réseau dans un certain état, il suffit connaître les fonctions u_p.

Lemme 4.10 *Si le routage est équitable alors le comportement du réseau temporisé routé est périodique.*

Démonstration. On peut sans perte de généralité ordonner totalement les transitions et décider que si dans un état de l'automate il y a plusieurs transitions franchissables, on choisit la plus petite dans cet ordre.

Un état de l'automate $\mathcal{A}$ est visité un nombre infini de fois. Entre deux états, exactement les mêmes transitions sont franchies dans le même ordre. Soit σ cette suite de transition, et soit $\mathcal{M}$ la matrice (max,plus) associée à cette suite. À partir d'un certain temps, la suite σ est donc répétée infiniment souvent.

La matrice $\mathcal{M}$ est irréductible. En effet, sinon, la partie vivante du graphe ne serait pas fortement connexe. Soit c la cyclicité de la matrice M. D'après le théorème 1.7, à partir d'un certain rang n_0, $(M^c)^{n+1} = \rho(M^c) \otimes (M^c)^n$. Donc après avoir franchi σ^c, toutes les colonnes du tas de pièces ont augmenté de $c.\rho$. Le débit est donc

$$D = \frac{|\sigma|}{|\rho(M)|}.$$

□

Remarque 4.11 La preuve montre comment calculer le débit : il se ramène à un calcul de valeur propre d'une matrice (max,plus) irréductible.

Lemme 4.12 *Si le routage n'est pas équitable, alors on peut calculer le débit sur les composantes fortement connexes maximales de la partie vivante. Le débit global est la somme des débits sur ces composantes.*

Démonstration. Par un raisonnement similaire à celui effectué pour les routages 0-1, si le routage n'est pas équitable, on peut supprimer les transitions jamais franchies. On obtient un réseau qui n'est plus fortement connexe, et la partie vivante du réseau est une union disjointe de composantes fortement connexes maximales terminales. Pour chacune de ces composantes, on calcule le débit par le lemme précédent. Si les valeurs propres des matrices ne sont pas commensurables, le comportement n'est pas périodique. □

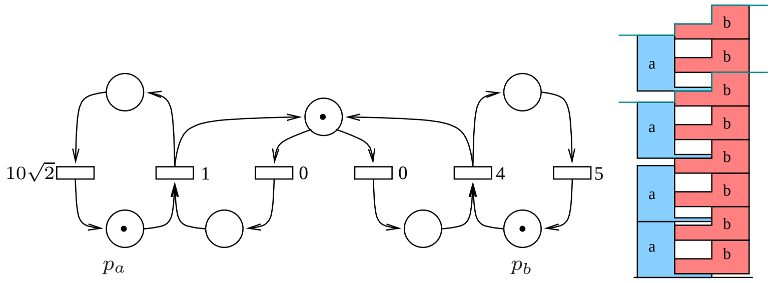


FIG. 4.3 – Comportement périodique pour un routage périodique.

Exemple 4.13 Pour le routage $(abb)^\infty$, l'évolution est montrée sur le tas de pièce de la figure 4.3 : la pièce a est le tir des transitions lorsque la transition de gauche est franchie et la pièce b pour les transitions de la partie droite. Lorsque les temporisations ne sont pas rationnelles, on obtient tout de même un comportement périodique.

4.3 Débits extrémaux avec des temporisations rationnelles

Dans cette section, nous supposons que toutes les temporisations sont rationnelles, et nous montrons que les routages extrémaux sont périodiques. Nous donnons aussi un algorithme de construction, qui est très proche de celui donné par [27].

Soit $(\mathcal{N}, M, \tau)$ un réseau de Petri à choix libres vivant borné et temporisé, avec $\tau_t \in \mathbb{Q}_+^*$, $\forall t \in \mathcal{T}$, et $\alpha = (\alpha_t)$ un vecteur de poids sur les transitions. Soit $x \in \mathbb{Q}$ tel que pour toute transition t, $\tau_t = xk_t$, avec $k_t \in \mathbb{N}^*$. On construit alors un nouveau réseau de Petri à choix libres vivant borné équivalent. Une transition t de temporisation $\tau_t = xk_t$ et de poids α_t est remplacée par k_t transitions en série de temporisation 1 et de poids $\alpha_t/(k_t x)$. Si σ est une suite infinie de tirs dans le réseau d'origine et σ' la suite infinie lui correspondant dans le réseau modifié, alors les débits coïncident. On peut donc considérer, sans perte de généralité, que pour tout $t \in \mathcal{T}$, $\tau_t = 1$.

4.3.1 Graphe des marquages au plus tôt

Soit N la matrice d'incidence de $\mathcal{N}$. Étant donné un marquage, plusieurs transitions peuvent être habilitées, et ce, plusieurs fois. Donc l'ensemble des transitions qui peuvent être franchies simultanément forme un multi-ensemble. Le graphe des marquages au plus tôt d'un réseau à choix libre est défini par l'algorithme 4.1.

La construction de l'algorithme 4.1 s'arrête car le réseau est borné. Ainsi, le graphe des marquages au plus tôt est fini. La construction ressemble à celle du graphe des marquages, mais ici, on franchit le maximum de transitions à la fois. *A priori*, le nombre de nœuds du graphe des marquages au plus tôt est donc inférieur au nombre de nœuds du graphe des marquages.

Algorithme 4.1 : Graphe des marquages au plus tôt

Données : $(\mathcal{N}, M)$ un réseau de Petri à choix libres.
Résultat : Graphe des marquages au plus tôt
début
  $Q \leftarrow \{M\}$; Arc$\leftarrow \emptyset$; $\bar{Q} \leftarrow \{M\}$;
  tant que $\tilde{Q} \neq \emptyset$ **faire**
    Choisir $M' \in \bar{Q}$;
    pour chaque *multi-ensemble maximal U de transitions franchissable simultanément depuis M'* **faire**
      $M'' \leftarrow M' + N.U$;
      si $M'' \notin Q$ **alors**
        $Q \leftarrow Q \cup \{M''\}$; $\bar{Q} \leftarrow \bar{Q} \cup \{M''\}$;
      finsi
      Arc$\leftarrow$Arc $\cup\{M' \to M''$, avec étiquette et poids $[U \mid \sigma_{t \in U} \alpha_t]\}$;
    finprch
    $\bar{Q} \leftarrow \bar{Q} \setminus \{M'\}$;
  fintq
fin

Toutes les évolutions au plus tôt du réseau de Petri peuvent être lues sur le graphe construit, d'où son nom. Dans ce graphe, le poids d'un chemin est la somme des poids des arcs du graphe. Le poids moyen d'un chemin est le poids du chemin divisé par le nombre d'arcs du chemin.

Théorème 4.14 *Soit $(\mathcal{N}, M, \tau)$ un réseau de Petri à choix libres vivant borné avec des temporisations rationnelles non nulles. Les débits extrémaux sont obtenus par des routages périodiques.*

Démonstration. Soit $(\bar{\mathcal{N}}, \bar{M})$ le réseau à choix libres vivant borné à partir $(\mathcal{N}, M)$ après duplication des transitions pour que toutes les transitions soient de temporisation 1. On construit le graphe des marquages au plus tôt de $(\bar{\mathcal{N}}, \bar{M})$. Soit c un circuit du graphe de poids moyen maximum. Le débit maximal peut être trouvé en suivant ce circuit.

De même, le débit minimal peut être trouvé en suivant un circuit de poids moyen minimal.

Ceci montre que les débits extrémaux correspondent à des routages périodiques. En effet, le routage peut être déduit des étiquettes le long des chemins suivis. □

Exemple 4.15 Considérons le réseau de Petri de la figure 4.4. Chaque transition est de temporisation 1 et de poids 1. Le graphe des marquages au plus tôt est représenté figure 4.5.

Le poids moyen minimal d'un circuit est 15/9, donné par le circuit $\{(dek), (afl), (cgl), (chi), (dej), (dgk), (agl), (bhi), (chk)\}$. Ceci donne le débit minimal et le routage pour l'atteindre. Le circuit de poids maximal est atteint pour le circuit $\{(dek), (agl), (chi)\}$. Et le débit maximal est 2.

Quand le réseau de Petri est construit, calculer les débits extrémaux peut se faire en temps cubique en le nombre de nœuds du graphe des marquages au plus tôt. Mais ce graphe peut avoir une taille exponentielle en la taille du réseau de Petri d'origine. Une première raison tient au fait que le nombre de marquages peut être exponentiel en la taille du réseau. Une seconde raison bient du nombre de transitions dupliquées pour obtenir des temporisations égales à 1. De plus, cette méthode ne donne aucune indication précise sur la structure du routage correspondant aux débits extrémaux.

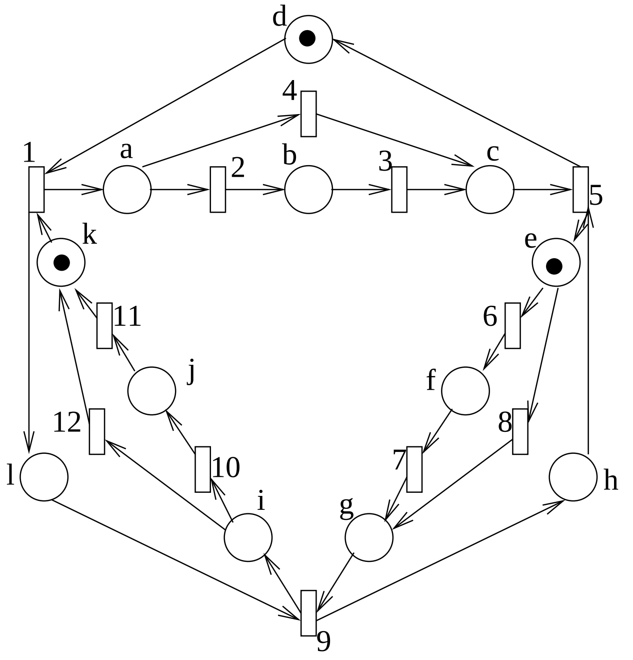


FIG. 4.4 – Exemple de réseau à choix libres.

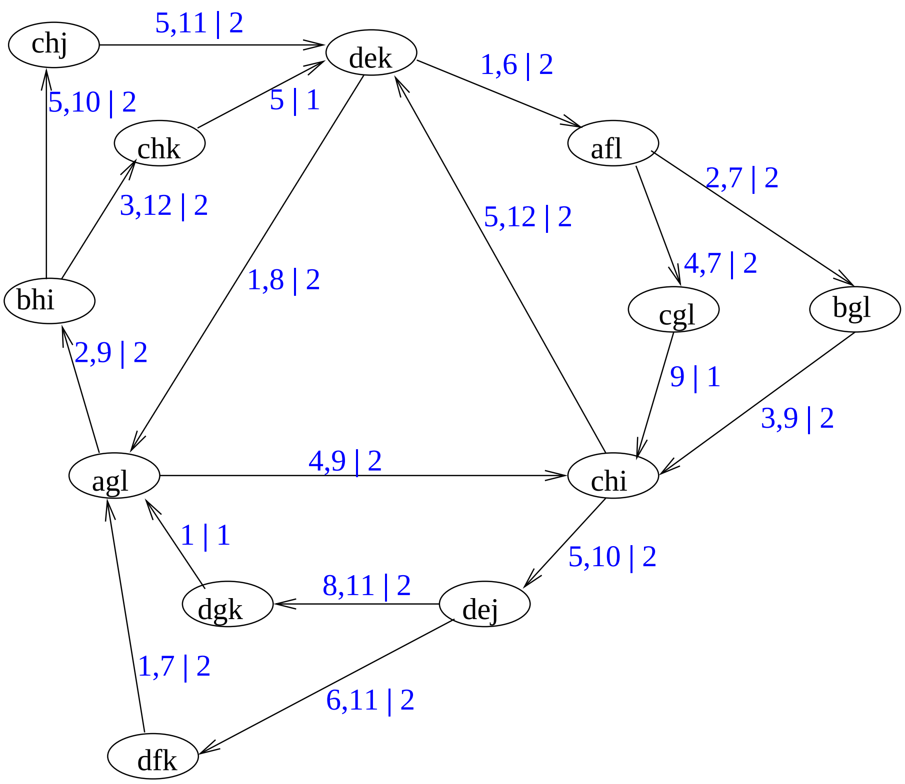


FIG. 4.5 – Graphe du marquage au plus tôt du réseau de Petri de la figure 4.4.

4.3.2 Cas des temporisations réelles non rationnelles

Pour des temporisations irrationnelles, le débit maximal n'est pas toujours périodique, même pour les réseaux 1-bornés. L'exemple suivant le montre.

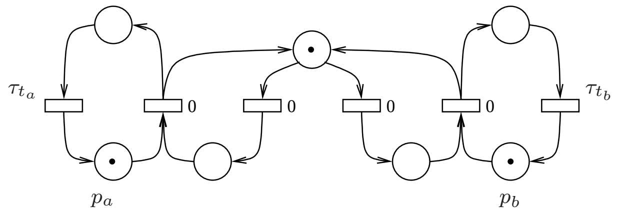


FIG. 4.6 – Réseau de Petri 1-borné dont le routage optimal n'est pas périodique.

Exemple 4.16 Considérons le réseau de la figure 4.6. Les temps de tir sont donnés à côté des transitions. On a $\tau_{t_a}/\tau_{t_b} \notin \mathbb{Q}$. Ce réseau a été étudié dans [70], dans le cadre plus général du modèle des tas de pièces. Le routage optimal est sturmien non périodique. Le routage le meilleur consiste à choisir la transition t_a ou la transition t_b, suivant qu'un jeton apparaît d'abord en p_a ou en p_b. Si t_a a tiré n_a fois et t_b n_b fois, il suffit de comparer $n_a\tau_{t_a}$ et $n_b\tau_{t_b}$. Le caractère non périodique provient de l'irrationalité du rapport entre les temporisations.

Pour des temporisations générales, nous ne savons pas si le débit minimal est atteint pour un routage périodique. Néanmoins, nous prouvons dans la partie suivante que le routage le pire pour les réseaux à choix libres 1-bornés est périodique.

4.4 Débit minimal dans les réseaux 1-bornés

4.4.1 TT-invariants

Dans le cas du calcul du débit minimal pour les réseaux 1-bornés, nous introduisons le concept de TT-invariants. Donnons d'abord quelques définitions sur les sous-suites de tirs.

Une sous-suite de tirs de $\sigma = \sigma(1)\dots\sigma(n) \in \mathcal{T}^*$ est une suite de tirs $\sigma' = \sigma'(1)\dots\sigma'(k)$ (pas nécessairement admissible) telle qu'il existe une suite strictement croissante $h : \mathbb{N}_k \to \mathbb{N}_n$ et $\sigma'(i) = \sigma(h(i))$.

Soit $h : \mathbb{N}_k \to \mathbb{N}_n$ et $h' : \mathbb{N}_{k'} \to \mathbb{N}_n$ deux fonctions strictement croissantes. La fonction h est plus petite que h' si le premier entier tel que $h(i) \neq h'(i)$ satisfait $h(i) < h'(i)$ ou si $k \leq k'$ et $h = h'_{|\mathbb{N}_k}$. Soit A un ensemble de suites finies de transitions. La première sous-suite de σ dans A est la plus petite sous-suite de σ qui appartient à A.

Définition 4.17 (suite de tirs compatible) *Soient $\pi = p_1t_1p_2t_2\dots p_\ell t_\ell$, $p_i \in \mathcal{P}$, $t_i \in \mathcal{T}$ un chemin de $\mathcal{N}$ et M un marquage marquant p_1. Soit σ une suite de tirs admissible de $(\mathcal{N}, M)$. La suite de tirs σ est dite* compatible *avec π si la première sous-suite de σ dans $[t_1][t_2]\dots[t_\ell]$ est $t_1t_2\dots t_\ell$.*

En d'autres termes, une suite est compatible avec un chemin dans un réseau de Petri si toutes les transitions de ce chemin sont franchies et dans cet ordre lorsque σ est tirée, et il ne peut y avoir de sous-suite $t_1t_2\dots t'_{\ell'}$ dans $[t_1][t_2]\dots[t_{\ell'}]$, $\ell' \leq \ell$, avec $t'_{\ell'} \neq t_{\ell'}$ plus petite dans l'ordre défini plus haut que $t_1t_2\dots t_\ell$.

Dans le cas des réseaux 1-bornés, puisqu'il ne peut y avoir plus d'un jeton par place, on peut « suivre » le jeton en p_1. Ce jeton doit suivre le chemin π.

Définition 4.18 (TT-invariant) *Soit $c = p_1 t_1 p_2 t_2 \dots p_\ell t_\ell$, $p_i \in \mathcal{P}$, $t_i \in \mathcal{T}$ un circuit de $\mathcal{N}$ et M un marquage marquant p_1. Un vecteur $J \in \mathbb{N}^{\mathcal{T}}$ est un TT-invariant généré par c et le marquage M si c'est un T-invariant et s'il est l'image commutative d'une suite de tirs compatible avec c.*

Un TT-invariant J généré par c et le marquage M est minimal si pour tout TT-invariant J' généré par c et le marquage M, $J \leq J'$.

Un TT-invariant généré par c dans $(\mathcal{N}, M)$ est minimal si pour tout TT-invariant J' généré par c et un marquage atteignable, $J \leq J'$.

Le terme TT-invariant est utilisé comme abréviation de Token-Transition invariant. Transition-invariant car un TT-invariant est un T-invariant, et Token pour jeton, puisque dans le cas 1-borné un jetons « parcourt » le circuit c et ce jeton revient alors à sa place initiale.

En dépit de ce que suggère la définition, les TT-invariants minimaux pour c ne dépendent pas du marquage de départ dans le sens où si l'image commutative de $\sigma_1.\sigma_2$ est un TT-invariant pour c, alors $\sigma_2.\sigma_1$ l'est aussi, en partant du marquage M' tel que $M \xrightarrow{\sigma_1} M'$.

Mais, contrairement aux T-invariants, les TT-invariants dépendent de l'ensemble des marquages atteignables. Nous allons voir par la suite qu'ils dépendent principalement du nombre maximal de jetons dans le circuit c.

TT-invariants d'un graphe d'événements

Nous donnons une caractérisation des TT-invariants minimaux dans les graphes d'événements.

Lemme 4.19 *Soit c un circuit élémentaire d'un graphe d'événements vivant et 1-borné contenant n jetons. Le TT-invariant minimal généré par c est $(n, \dots, n) \in \mathbb{N}^{\mathcal{T}}$.*

Démonstration. Les T-invariants d'un graphe d'événements sont de la forme $(x, \dots, x) \in \mathbb{N}^*$ ([34, Prop. 3.16]). Soit J un TT-invariant minimal associé à $c = p_1 t_1 \dots$ et soit σ une suite de tirs correspondant à ce TT-invariant. Puisque le réseau est 1-borné, les jetons ne peuvent pas se dépasser. Par conséquent, quand le jeton initialement en p_1 y revient, après le tir de σ, on sait que la transition t_1 a été franchie n fois, ce qui permet de conclure. □

Exemple 4.20 La figure 4.7 montre l'évolution d'un graphe d'événements contenant deux jetons dans le circuit c. Regardons le TT-invariant généré par c. En appliquant le lemme 4.19, le TT-invariant minimal est $(2, 2, 2, 2, 2, 2, 2, 2, 2)$. Le jeton blanc (et le jeton noir) est de retour dans sa place initiale (figure 4.7(c)).

Le T-invariant minimal est $(1, 1, 1, 1, 1, 1, 1, 1, 1)$. Observons qu'après le tir de chaque transition une fois (4.7(b)), le marquage est inchangé, mais la place du jeton noir et du jeton blanc ont été échangées. Après un tir supplémentaire de chaque transition, (figure 4.7(c)), les jetons sont de retour dans la bonne place.

TT-invariants d'un réseau à choix libres 1-borné

Nous caractérisons maintenant les TT-invariants minimaux dans les réseaux à choix libres vivants 1-bornés. C'est un cas beaucoup plus difficile, et il n'y a plus unicité.

Nous allons tout d'abord donner un algorithme pour construire les TT-invariants générés par un circuit c, et ensuite montrer que tout TT-invariant minimal généré par un circuit élémentaire c est la somme de n T-invariants minimaux, où n est le nombre maximal de jetons dans c.

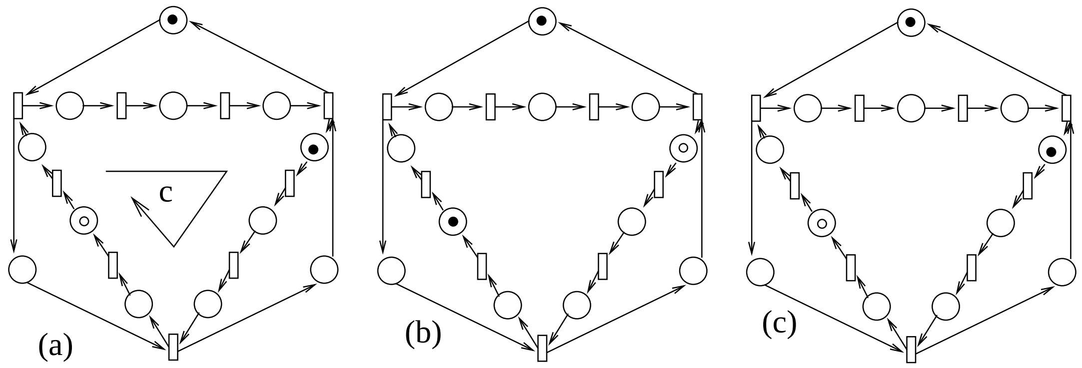


FIG. 4.7 – TT-invariant dans un graphe d'événements.

Définissons tout d'abord quelques notations, qui seront utilisées dans les preuves. Soit σ une suite de tirs admissible contenant la transition b. On note $\sigma|_b$ la première sous-suite parmi les plus longues sous-suites admissibles de σ ne contenant pas b (donc $\sigma|_b$ est bien définie). Intuitivement, $\sigma|_b$ est la sous-suite de σ tirée quand le temps de tir de b est infini. On note $\sigma \setminus \sigma|_b$ le complémentaire de $\sigma|_b$ dans σ. Si M' est le marquage obtenu après avoir franchi $\sigma|_b$, b est la seule transition franchissable parmi les transitions de $\sigma \setminus \sigma|_b$.

Lemme 4.21 *Soit c un circuit de $\mathcal{N}$, un réseau à choix libres vivant 1-borné, et b une transition de c. On note M_b l'unique marquage de blocage associé à b. Pour tout TT-invariant minimal J généré par c, il existe une suite de tirs σ compatible avec c telle que $\overrightarrow{\sigma} = J$ et telle que σ peut être franchie à partir de M_b.*

Démonstration. Soit J un TT-invariant minimal généré par c et σ une suite de tirs telle que $\overrightarrow{\sigma} = J$, admissible à partir d'un marquage atteignable M. Soit b une transition de c et M' le marquage tel que $M \xrightarrow{\sigma|_b} M'$. Soit $Cluster(\sigma \setminus \sigma|_b)$ l'ensemble des clusters dont au moins une transition apparaît dans $\sigma \setminus \sigma|_b$.

Montrons maintenant qu'il existe une suite de tirs de M' vers M_b dont aucune transition n'est dans $Cluster(\sigma \setminus \sigma|_b)$. Soit σ_1 une suite de tirs de M' vers M_b et t la première transition, si elle existe, qui appartient à $Cluster(\sigma \setminus \sigma|_b)$. Alors il existe une place p en entrée de t qui est marquée quand σ_1 est franchie. Mais cette place est aussi marquée dans le tir de $\sigma \setminus \sigma|_b$, qui est indépendante des tirs de σ_1 jusqu'à t. Mais alors, p n'est pas 1-bornée. Ceci prouve qu'une telle transition t ne peut pas exister, et M_b peut être atteint sans franchir de transitions de $Cluster(\sigma \setminus \sigma|_b)$. □

Proposition 4.22 *Soit c un circuit élémentaire de $\mathcal{N}$, un réseau à choix libres vivant 1-borné. Tout TT-invariant minimal généré par c peut être construit en appliquant l'algorithme 4.2.*

Dans cet algorithme, une suite de tirs σ telle que $M \xrightarrow{\sigma} M'$ est minimale s'il n'existe pas de sous-suite σ' de σ telle que $M \xrightarrow{\sigma'} M'$. Cette suite minimale n'est pas unique, et donc une entrée de l'algorithme peut mener à plusieurs sorties.

Démonstration. Par construction, la sortie de l'algorithme 4.2 est un TT-invariant. Soit J un TT-invariant minimal généré par c. D'après le lemme 4.21, il existe $\sigma \in \mathcal{T}^*$ telle que $\overrightarrow{\sigma} = J$ et $M_{t_1} \xrightarrow{\sigma} M_{t_1}$. Considérons maintenant la suite maximale admissible $\sigma|_{t_2}$. Franchir les transitions de $\sigma|_{t_2}$ mène au marquage M_2, et aucune transition de $\sigma \setminus \sigma|_{t_2}$ ne peut être franchie à part t_2.

Algorithme 4.2 : Construction d'un TT-invariant minimal

Données : $\mathcal{N}$, un réseau à choix libres vivant 1-borné, $c = t_1 \dots t_k$ un circuit de $\mathcal{N}$.
Résultat : Un TT-invariant minimal généré par c.
début
  $\sigma \leftarrow \epsilon$;
  pour chaque $i \in \{1, \dots, k\}$ **faire**
    $\sigma_i \leftarrow$ une suite de tirs minimale de M_{t_i} à $M_{t_{i+1}}$ dans $\mathcal{N}$;
  finprch
  $\sigma \leftarrow \sigma_1 \dots \sigma_k$;
  retourner $\overrightarrow{\sigma}$;
fin

Si M_2 n'est pas M_{t_2}, le marquage de blocage de t_2, cela veut dire qu'il existe une transition r distincte de t_2 qui est franchissable à partir de M_2. Par définition de $\sigma|_{t_2}$, cette transition reste franchissable pendant le franchissement de $\sigma \setminus \sigma|_{t_2}$. Or, dans le marquage final M_{t_1}, la seule transition franchissable est t_1. Ainsi, $r = t_1$.

Maintenant, à partir de M_2, on franchit la plus longue sous-suite admissible de $\sigma \setminus \sigma|_{t_2}$ sans franchir t_3. On atteint alors le marquage M_3, où les seules transitions franchissables sont t_3 et t_1, en faisant le même raisonnement que précédemment. En répétant encore l'argument pour les autres transitions, on atteint le marquage M_k où les seules transitions franchissables possibles sont t_1 et t_k. Finalement, on tire la transition t_k, mais alors, on ajoute un jeton supplémentaire dans la place entre t_k et t_1 dans c. Si cette place était déjà marquée, alors elle contiendrait deux jetons. Ce n'est pas possible puisque le réseau est 1-borné. Donc, dans tous les marquages M_i, $i \in \{2, \dots, k\}$, la seule transition franchissable était t_i, et $M_i = M_{t_i}$. Le vecteut J peut donc être généré par l'algorithme 4.2. □

Nous nous intéressons maintenant aux TT-invariants minimaux générés par un circuit élémentaire. Nous montrons que ceux-ci sont la somme de n TT-invariants minimaux, où n est le nombre maximal de jetons dans ce circuit. Le nombre maximal de jetons dans un circuit élémentaire est le nombre de jetons dans ce circuit dans le marquage de blocage d'une transition de ce circuit. Les deux lemmes 4.23 et 4.24 sont deux lemmes techniques en vue de prouver ce résultat (proposition 4.25).

Lemme 4.23 *Soit $c = t_1 \dots t_k t_{k+1} = t_1$ un circuit élémentaire d'un réseau à choix libres vivant 1-borné $\mathcal{N}$. Pour tout $i \in \{1, \dots, k\}$, et pour toute suite minimale de tirs σ de M_{t_i} à $M_{t_{i+1}}$, il existe un T-invariant minimal J tel que $\chi_{\{t_i, t_{i+1}\}} \leq J$ et $\overrightarrow{\sigma} \leq J$.*

Démonstration. Dans cette preuve, on pose $t = t_i$ et $t' = t_{i+1}$. Les transitions t et t' sont sur un même circuit élémentaire, donc, par le lemme 4.7, il existe une T-composante C contenant t et t'.

Soit σ une suite de tirs minimale de M_t à $M_{t'}$ et σ' une suite de tirs minimale de $M_{t'}$ à M_t en utilisant uniquement des transitions de C. Alors $\overrightarrow{\sigma.\sigma'}$ est un T-invariant minimal. En effet, c'est un T-invariant et t et t' ne sont franchies qu'une seule fois (car le réseau est 1-borné et les suites de tir sont minimales). Ainsi, les autres transitions de C sont aussi franchies une et une seule fois, puisque les T-invariants des graphes d'événements sont de la forme $(x, \dots, x)$. Notons aussi que les transitions apparaissant dans σ' n'apparaissent qu'une fois.

Maintenant, soient σ une suite de tirs minimale de M_t à $M_{t'}$, dont les transitions ne sont pas nécessairement dans C, et σ' définie comme dans le paragraphe précédent. On a donc $\overrightarrow{\sigma'} \leq \chi_C$.

L'image commutative, J, de $\sigma.\sigma'$ est un T-invariant ($M_t \stackrel{\sigma.\sigma'}{\to} M_t$). Soit p la place du ciucuit c entre t et t'. Nous avons $\sum_{a \in p^\bullet} J_a = 1$ car aucune transition de $p^\bullet$ n'apparaît dans σ par minimalité de σ et car seule t' apparaît, une fois, dans σ'. De plus, t n'apparaît qu'une seule fois dans J : il y a autant d'occurrences de transitions de $p^\bullet$ que de ${}^\bullet p$ dans un T-invariant. Donc, J est un T-invariant où t et t' apparaissent une et une seule fois.

Considérons maintenant J_1 un T-invariant minimal inclus dans J et contenant t et t'. Il existe alors σ_1 et σ'_1 deux suites de tirs telles que $M_t \stackrel{\sigma_1}{\to} M_{t'} \stackrel{\sigma_1}{\to} M_t$ et $J_1 = \overrightarrow{\sigma_1\sigma'_1}$. Puisque σ' et σ'_1 sont toutes les deux des suites minimales de tir, aucune n'est strictement incluse dans l'autre. Donc, si elles ne sont pas égales, il existe $a_1 \in \sigma'_1 \setminus \sigma'$. Alors le T-invariant $\sigma_1.\sigma'$ contient t et t' une fois chacune mais ne contient pas a_1.

Considérons de nouveau J_2 un T-invariant minimal inclus dans $\sigma_1\sigma'$ et contenant t et t'. De nouveau, il existe σ_2 et σ'_2 deux suites de tirs telles que $M_t \stackrel{\sigma_2}{\to} M_{t'} \stackrel{\sigma_2}{\to} M_t$ et $J_2 = \overrightarrow{\sigma_2\sigma'_2}$. Encore, si σ' et σ'_2 ne sont pas égales, il existe $a_2 \in \sigma'_1 \setminus \sigma'$. Dans ce cas, le T-invariant $\sigma_2\sigma'$ ne contient pas $\{a_1, a_2\}$. Cette construction peut continuer jusqu'à ce que σ' et σ'_i coïncident. Ceci arrive nécessairement au bout d'un nombre fini de pas (le nombre de transitions est fini). Finalement, on a $\sigma' = \sigma'_i$ et donc $\sigma_i\sigma'_i = \sigma_i\sigma' \subseteq \sigma\sigma'$. Alors, $J - J_i = \overrightarrow{\sigma\sigma'} - \overrightarrow{\sigma_i\sigma'_i}$ est un T-invariant inclus dans $\overrightarrow{\sigma}$. Puisque σ est une suite de tirs minimale, $J - J_i$ doit être nul. Donc $J = J_i$ est un T-invariant minimal. □

Lemme 4.24 *Soit c un circuit élémentaire d'un réseau à choix libres vivant 1-borné et n le nombre maximal de jetons dans c. Il existe un TT-invariant minimal généré par c qui est n fois le même T-invariant minimal.*

Démonstration. D'après le lemme 4.7, il existe une T-composante C contenant le circuit c. D'après le théorème 4.6, il existe aussi un routage 0-1 sous lequel la partie vivant du réseau de Petri est C. Dans le marquage de blocage d'une transition t de c, il y a n jetons dans c. Soit $\mathcal{T}_C$ l'ensemble des transitions de C. En appliquant le lemme 4.19 au graphe d'événement (C, M_t), on trouve que $n\chi_{\mathcal{T}_C}$ est un TT-invariant minimal généré par c. □

Proposition 4.25 *Soit c un circuit élémentaire d'un réseau à choix libres vivant 1-borné et n le nombre maximal de jetons dans c. Tout TT-invariant minimal généré par c est la somme de n T-invariants minimaux.*

Démonstration. D'après le lemme 4.24, il existe un TT-invariant minimal généré par c qui est la somme de n T-invariant minimaux. Soit J un tel TT-invariant, égal à nK, où K est un T-invariant minimal correspondant à une T-composante contenant le circuit c. Soit σ la suite de tirs telle que $\overrightarrow{\sigma} = J$ et telle que σ corresponde à la concaténation de $\sigma_{t_it_{i+1}}$, suite de tirs du marquage M_{t_i} à $M_{t_{i+1}}$ trouvée par l'algorithme 4.2.

En utilisant le lemme 4.23, on sait que les TT-invariants peuvent être obtenus en remplaçant les $\sigma_{t_it_{i+1}}$ par d'autres suites de tirs minimales. Nous allons donc prouver le résultat par induction sur le nombre de remplacements utilisés pour avoir un autre TT-invariant minimal. Plus précisément, nous prouvons que tous TT-invariant minimal généré par c est la somme de n T-invariants minimaux et que pour tout i, $\sigma_{t_it_{i+1}}$ appartient à un T-invariant dans la somme.

Le résultat est vrai au début pour $J = nK$, et $\overrightarrow{\sigma_{t_it_{i+1}}}$ pour tout i.

Soit t et t' deux transitions consécutives dans c, et remplaçons la suite d'origine $\sigma_{tt'}$ par une autre suite de tirs, de M_t à $M_{t'}$, notée $\sigma'_{tt'}$. Un nouveau TT-invariant généré par c est alors obtenu. Par hypothèse de récurrence, $\sigma_{tt'}$ appartient à un T-invariant minimal, H, apparaissant dans la somme des n T-invariants formant J. On a $H = \overrightarrow{\sigma_{tt'}.w}$, où $M_{t'} \xrightarrow{w} M_t$. En utilisant le lemme 4.23, $L = \overrightarrow{\sigma'_{tt'}.w}$ est aussi un T-invariant minimal. Le nouveau TT-invariant minimal $J' = J - H + L$ est aussi somme de n T-invariants positifs, et toutes les suites $\sigma_{v,v'}$ avec $v \neq t$ appartiennent à un T-invariant de la somme (ils n'ont pas changé). L'hypothèse de récurrence est donc toujours vérifiée pour ce nouveau TT-invariant, ce qui finit la preuve. □

Exemple 4.26 La figure 4.4 illustre la proposition 4.25. Considérons le circuit $c = 1, 2, 3, 5, 6, 7, 9, 10, 11$. Le marquage de blocage de la transition 1 est $\{d, g, k\}$. Pour atteindre le marquage de blocage de la transition 2, il y a deux suites de tirs possibles : $1, 9, 10, 11$ et $1, 9, 12$. Pour atteindre le marquage de blocage de la transition 3, la transition 2 est franchie, et c'est la seule possibilité. De même, pour atteindre le marquage de blocage de la transition 5, la transition 3 est franchie.

Par symétrie du réseau, il existe aussi deux possibilités pour aller de M_5 à M_6, à savoir $5, 1, 2, 3$ et $5, 1, 4$, et de M_9 à M_{10}, $9, 5, 6, 7$ et $9, 5, 8$. Toutes les autres suites de tir minimales d'un marquage de blocage d'une transition de c au marquage de blocage de la prochaine transition sur c ne contiennent qu'une transition. Il y a donc 8 TT-invariants minimaux pour c (pour trois sous-suites, il y a deux possibilités). Par exemple, $1, 9, 10, 11, 2, 3, 5, 1, 4, 6, 7, 9, 1, 8, 10, 11$ est une suite de tirs minimale compatible avec c. Son image commutative peut s'écrire comme $I + J$, où I et J sont des T-invariant minimaux de support respectif $1, 2, 3, 5, 6, 7, 9, 10, 11$ et $1, 4, 5, 8, 9, 10, 11$.

Nous pouvons maintenant énoncer un dernier lemme, caractérisant les TT-invariants minimaux pour un circuit quelconque :

Lemme 4.27 *Soit $\mathcal{C}$ un circuit élémentaire d'un réseau à choix libres vivant 1-borné et J un TT-invariant minimal pour $\mathcal{C}$. Le circuit $\mathcal{C}$ se décompose en circuits élémentaires $c_1, \ldots, c_k$. Il existe alors $J_1, \ldots, J_k$, k TT-invariants minimaux générés respectivement par $c_1, \ldots, c_k$ tels que $J = \sum_{i=1}^{k} J_k$.*

Démonstration. C'est immédiat d'après l'algorithme de construction des TT-invariants minimaux. □

4.4.2 Calcul du débit minimal pour les réseaux 1-bornés

On considère maintenant tous les routages possibles. On a vu que le routage le meilleur pour les réseaux 1-bornés n'est pas toujours périodique. Qu'en est-il du routage le pire ? Nous allons voir que l'on peut caractériser le routage qui minimise le débit dans le cas des réseaux 1-bornés à l'aide des TT-invariants et donner un algorithme pour le calculer.

Considérons $(\mathcal{N}, M, \tau)$ un réseau de Petri à choix libres vivant 1-borné temporisé. Dans cette partie, nous montrons que le débit minimal est atteint pour un routage périodique, même pour des temporisations non rationnelles. De plus, nous donnons une borne sur la période du routage et précisons quelle est la structure de ce routage. En fait, cette structure correspond à un TT-invariant minimal.

Circuit critique

Supposons à nouveau que toutes les temporisations sont égales à 1. Le routage le pire peut être trouvé en considérant sur le graphe des marquages au plus tôt un circuit de poids moyen minimal. Notons $c = (M_0, M_1, \ldots, M_{T-1})$ un tel circuit. La longueur de ce circuit est T, la période de l'évolution.

Soit t_0 une transition apparaissant dans l'étiquette de l'arc entre M_0 et M_1. On construit un chemin dans le réseau de Petri d'extrémité finale t_0 de la manière suivante : si le chemin $t_i, \dots, t_1, t_0$ est construit, alors on choisit une transition t_{i+1} dans l'étiquette U_i entre $M_{T-i-1[T]}$ et $M_{T-i[T]}$ telle que $t_{i+1} \in {}^{\bullet\bullet}t_i$.

On arrête la construction lorsque l'on a trouvé $m \in \mathbb{N}^*$ et une transition t_j telle que $t_j = t_{j-mT}$. On considère maintenant le circuit dans le réseau de Petri correspondant à la suite de transitions $t_j, t_{j-1}, \dots, t_{j-mT}$, que l'on note $\mathcal{C}_c$. Il est de taille mT par construction.

Soit U_i l'ensemble des transitions franchies entre $M_{T-i-1[T]}$ et $M_{T-i[T]}$ (ce n'est plus un multi-ensemble, puisque le réseau est 1-borné). Soit K l'image commutative de $\sigma = U_{j-1} \dots U_{j-mT}$. Par construction, K est un T-invariant. Toujours par construction, $t_{j-1} \dots t_{j-mT}$ est le premier sous-mot de σ appartenant à $[t_{j-1}] \dots [t_{j-mT}]$. Donc K est un TT-invariant généré par $\mathcal{C}_c$.

Puisque K est un T-invariant associé à une suite de tirs suivant un circuit de poids moyen minimal dans le graphe des marquages au plus tôt, le débit minimal $\underline{D}$ de $(\mathcal{N}, M, \tau)$ est :

$$\underline{D} = \frac{\alpha^T K}{mT}. \tag{4.3}$$

Puisque K est un TT-invariant généré par $\mathcal{C}_c$, il existe un TT-invariant minimal $J_{\mathcal{C}_c}$ généré par $\mathcal{C}_c$ tel que $K \geq J_{\mathcal{C}_c}$. Le circuit $\mathcal{C}_c$ peut ne pas être élémentaire. Supposons-le composé de k circuits élémentaires, $c_1, \dots, c_k$, de longueurs respectives $\ell_1, \dots, \ell_k$, avec $\sum_{i=1}^k \ell_i = mT$. Par le lemme 4.27, on a $J_c = \sum_{i=1}^k J_i$ où J_i est un TT-invariant minimal généré par c_i. Alors,

$$\underline{D} = \frac{\alpha^T K}{mT} \geq \frac{\alpha^T J_c}{mT} = \frac{\alpha^T(\sum_{i=1}^k J_i)}{\sum_{i=1}^k \ell_i} = \frac{\sum_{i=1}^k \alpha^T J_i}{\sum_{i=1}^k \ell_i} \geq \min_{i=1}^{k} \frac{\alpha^T J_i}{\ell_i}. \tag{4.4}$$

Comme $\underline{D}$ est le débit minimal, on a aussi $\underline{D} \leq \min_{i=1}^k \frac{\alpha^T J_i}{\ell_i}$. Par conséquent, on peut trouver un circuit élémentaire c du réseau de Petri, de longueur ℓ et associé à un TT-invariant minimal J tel que $\underline{D} = \frac{\alpha^T J}{\ell}$. On appelle un tel circuit un *circuit critique*.

On montre maintenant que ce résultat est vrai non seulement pour des temporisations rationnelles, mais aussi pour des temporisations réelles quelconques.

Théorème 4.28 *Considérons un réseau de Petri à choix libres vivant 1-borné temporisé avec des temporisations réelles strictement positives. Le débit minimal est obtenu pour un routage périodique. Pour chaque place, la période de ce routage est bornée par le nombre maximal de jetons dans le circuit.*

Démonstration. Supposons d'abord que les temporisations sont rationnelles et non nulles. Alors on peut considérer sans perte de généralité (comme vu précédemment) que toutes les temporisations sont égales à 1.

Soit c un circuit critique du réseau de Petri. En particulier, c est un circuit élémentaire. Par la proposition 4.25, un TT-invariant minimal généré par c est de la forme

$$J = \sum_{i=1}^{n} J_i, \qquad \text{avec } J_i \text{ T-invariant minimal,}$$

où n est le nombre maximal de jetons dans c.

Alors, chaque place est marquée au plus n fois quand une suite d'image commutative J est tirée. Comme n est au plus le nombre maximal de jetons dans le réseau de Petri, le résultat est prouvé dans le cas des temporisations rationnelles.

Pour prouver le résultat dans le cas général, nous avons besoin du lemme suivant, qui établit la continuité du débit en fonction des temporisations.

Lemme 4.29 *Soit $(\mathcal{N}, M, u, \tau)$ un réseau de Petri à choix libres vivant 1-borné. Soit α un vecteur de poids sur les transitions et considérons le débit obtenu avec ces poids. Le débit est continu et décroissant en fonction des temporisations $(\tau)_{t\in\mathcal{T}}$ sur $(\mathbb{R}_+^*)^{\mathcal{T}}$.*

Démonstration. Pour prouver ce résultat, on utilise la représentation des réseaux 1-bornés par des tas de pièces. Puisque le réseau est routé, la suite infinie résultant de ce routage est définie de manière unique à commutation près des transitions concurrentes et est indépendante des transitions. Soit σ une de ces suites de tirs. Empilons les pièces correspondantes une par une selon σ. On note H_n la hauteur du tas de pièces après empilement des n premières pièces et A_n le poids des n premières transitions. Le débit est défini par $D = \liminf_{n\to\infty} A_n/H_n$.

Changeons la temporisation d'une transition t_0 de τ_{t_0} à $\tau_{t_0} + \varepsilon$, $\varepsilon > 0$. Soit $D(\varepsilon)$ le nouveau débit. Après avoir empilé les n premières pièces, le poids des pièces empilées est toujours A_n et la hauteur du tas est $H'_n \leq H_n + K_n\varepsilon$, où K_n est le nombre d'occurrences de t_0 dans les n premières pièces. Or, $D(\varepsilon) = \liminf_{n\to\infty} A_n/H'_n$. Puisque $A_n/H_n > A_n/H'_n$, on a $D \leq D(\varepsilon)$. De plus,

$$0 \leq \frac{A_n}{H_n} - \frac{A_n}{H'_n} \leq \frac{A_n}{H_n} - \frac{A_n}{H_n + K_n\varepsilon} \leq \frac{A_nK_n\varepsilon}{H_n(H_n + K_n\varepsilon)} \leq \frac{A_nK_n\varepsilon}{H_n^2}. \tag{4.5}$$

Soient $\underline{\tau}$ le temps de tir minimal d'une transition et C le nombre de colonnes du tas de pièces. Nous avons $H_n \geq n\underline{\tau}/C$. De plus, $K_n \leq n$. Donc,

$$\frac{A_n}{H_n} - \frac{A_n}{H'_n} \leq \frac{A_nC\varepsilon}{H_n\underline{\tau}}. \tag{4.6}$$

Quand n tend vers l'infini, on obtient $D - D(\varepsilon) \leq Cste\, D\varepsilon$. En particulier, $\lim_{\varepsilon\to n} D(\varepsilon) = D$. Le même type d'argument s'applique si on change τ_{t_0} en $\tau_{t_0} - \varepsilon$ avec $\varepsilon > 0$, ce qui achève la preuve. □

Suite de la preuve du théorème 4.28

Notons $\underline{D} : (\mathbb{R}_+^*)\mathcal{T} \to \mathbb{R}_+^*$ l'infimum sur tous les routages possibles du débit vu comme une fonction des temporisations. Clairement, $\underline{D}$ est décroissante. Soit $x \in (\mathbb{R}_+^*)^{\mathcal{T}}$. On peut approcher s par une suite de vecteurs à coefficients rationnels $(x^{(k)})_{k\in\mathbb{N}}$ telle que $\forall k \in \mathbb{N}$, $x^{(k)} \geq x$. On a donc $\forall k \in \mathbb{N}$, $\underline{D}(x^{(k)}) \leq \underline{D}(x)$. Soit m le nombre maximal de jetons dans un circuit. Pour des temporisations rationnelles, on sait que le débit minimal est atteint pour un routage périodique dont la période est au plus m (c'est l'objet de la première partie de la preuve). Or, il y a seulement un nombre fini de tels routages. Il existe donc une sous-suite $(x^{(\phi(k))})$ de temporisations pour laquelle le débit minimal correspond au même routage périodique, r. Soit $D(x, r)$ le débit obtenu avec le routage r pour les temporisations x. D'après le lemme 4.29, $D(x, r) = \lim_{k\to\infty} \underline{D}((x^{(\phi(k))})$. Mais alors, $D(x, r) \leq \underline{D}(s)$, ce qui permet ce conclure : $D(x, r) = \underline{D}(s)$. □

Exemple 4.30 Reprenons l'exemple de la figure 4.4. Son graphe des marquages au plus tôt est montré en figure 4.5, et l'on trouve que le circuit critique est $(1, 2, 3, 5, 6, 7, 9, 10, 11)$. Puisque toutes les transitions ont un poids de 1, les suites de tirs de poids minimal d'un marquage de blocage d'une transition du circuit jusqu'à la prochaine sont (en commençant par le marquage de blocage de la transition 1 qui est $\{d, k, g\}$) : $(1, 9, 12)$; (2) ; (3) ; $(5, 1, 4)$; (6) ; (7) ; $(9, 5, 8)$; (10) ; (11). Le TT-invariant minimal qui permet d'atteindre le débit minimal est la somme de deux T-invariants minimaux de supports $1, 2, 3, 5, 6, 7, 9, 10, 11$ et $1, 4, 5, 8, 9, 12$. Le débit minimal est alors $15/9$, comme trouvé avec les autres méthodes. Le routage le pire peut être calculé directement à partir du TT-invariant minimal : $u_a = (2, 4)^\infty$, $u_e = (8, 6)^\infty$ et $u_i = (12, 10)^\infty$.

Algorithme de calcul du débit minimal

On considère toujours $(\mathcal{N}, M, \tau)$ un réseau à choix libres vivant 1-borné temporisé. La matrice d'incidence de $\mathcal{N}$ est notée N. On note Clusters l'ensemble des clusters de $\mathcal{N}$, et on définit $K \in \{0,1\}^{\text{Clusters}\times\mathcal{T}}$ la matrice telle que $K_{a,t} = 1$ si t appartient au cluster a, et $K_{a,t} = 0$ sinon.

Soit S un sous-ensemble de $\mathcal{T}$. L'algorithme **T-invariant-léger** calcule un T-invariant de poids minimal qui contient les transitions de S. C'est donc la solution de l'algorithme 4.3.

Algorithme 4.3 : T-invariant-léger

Données : $S \subseteq \mathcal{T}$
début
  Minimiser $\alpha^T \cdot I$;
  Avec les contraintes : $N \cdot I = 0$; $K \cdot I \leq (1, \ldots, 1)^T$; $I \geq \chi_S$.
fin

Par la suite, quand l'algorithme sera appelé, il sera appliqué pour un ensemble de deux transitions t et t' telles que $t' \in t^{\bullet\bullet}$, en particulier, avec des transitions étant sur une même T-composante. La condition $K \cdot I \leq (1, \ldots, 1)^T$ assure que la sortie est un ensemble de T-invariants minimaux de supports disjoints, dont un seul contient S (et tous les autres étant de poids nul).

L'algorithme **T-invariant-léger** renvoie le résultat en temps polynomial, puisque c'est un problème de programmation linéaire sur $\mathbb{Q}$. De plus, les contraintes sont telles que la solution est toujours un vecteur de $\{0,1\}^{\mathcal{T}}$.

Définissons maintenant les fonctions **Marquage-de-blocage**, **Temps-de-cycle** et **Temporisation** :

- **Marquage-de-blocage**(t) calcule le marquage de blocage de la transition t. Ce marquage peut être trouvé en temps $O(|\mathcal{T}|^3)$ pour les réseaux à choix libres (voir [52]) ;
- **Temps-de-cycle**(A) calcule le temps de cycle de la matrice (max,plus) A. On applique cette fonction à des matrices de taille au plus $|\mathcal{P}|$. La complexité est donc en temps $O(|\mathcal{P}|^3)$ en utilisant l'algorithme de Karp (voir [11] par exemple) ;
- **Temporisation**(c) calcule la somme des temps de tirs des transitions le long du circuit c. Ceci se fait en temps linéaire.

Dans l'algorithme 4.4, on utilise la représentation (max,plus) du comportement d'un réseau à choix libres vivant borné. Celui-ci est représenté par un tas de pièce et, pour toute transition b, A_b est la matrice (max,plus) représentant le comportement temporisé de la transition b. On rappelle que le symbole $\otimes$ représente la multiplication des matrices dans (max,plus). Cette opération se fait en temps cubique par rapport à la taille des matrices utilisées. Ici, cette taille est au plus $|\mathcal{P}|$.

Les résultats précédents permettent d'énoncer le théorème suivant :

Théorème 4.31 *L'algorithme 4.4 trouve le débit minimal d'un réseau à choix libres vivant 1-borné.*

À chaque pas de la première boucle, la complexité est de $O(|\mathcal{T}|)$, et il y a $|\mathcal{T}|$ itérations. Considérons maintenant la deuxième boucle. Il y a au plus $|\mathcal{T}|^2$ passages dans la boucle. À chaque passage, on doit trouver une suite de tirs minimale de M_b à $M_{b'}$ dans une T-composante. Ceci peut être fait en un temps $O(|\mathcal{T}|^2)$. D'après ce qui précède, chaque transition apparaît au plus une fois, donc la matrice $A_{bb'}$ peut être calculée en temps $O(|\mathcal{P}|^3|\mathcal{T}|)$.

Il y a autant de passages dans la dernière boucle que de circuits élémentaires dans le réseau. Chaque itération dans cette boucle a une complexité de $O(|\mathcal{P}|^3|\mathcal{T}|)$. La complexité de cette

---

Algorithme 4.4 : Routage-le-pire

Données : $(\mathcal{N}, M, \tau)$ un réseau à choix libre vivant 1-borné et temporisé $(\tau \in (\mathbb{R}_+^*)^{\mathcal{T}})$;
$\alpha = (\alpha_t)_{t \in \mathcal{T}} \neq (0, \dots, 0)$, un vecteur de poids.

début
  pour chaque $b \in \mathcal{T}$ **faire**
    $M_b \leftarrow$ **Marquage-de-blocage**(b);
  finprch
  pour chaque $b, b' \in \mathcal{T}$ *tels que* $\exists p,\ b \to p \to b'$ **faire**
    $J \leftarrow$**T-invariant-léger**(b, b');
    $\sigma_{bb'} \leftarrow b, t_1, \dots, t_m$ suite minimale de tirs de M_b à $M_{b'}$ avec des transitions de J;
    $A_{bb'} \leftarrow A_b \otimes A_{t_1} \otimes \cdots \otimes A_{t_m}$;
    $\alpha_{bb'} \leftarrow \alpha_b + \alpha_{t_1} + \cdots + \alpha_{t_m}$;
  finprch
  Débit $\leftarrow +\infty$;
  Tmin $\leftarrow \emptyset$;
  pour chaque *circuit élémentaire* $c = t_1 \dots t_k$ *de* $\mathcal{N}$ **faire**
    $A \leftarrow A_{t_1 t_2} \otimes A_{t_2 t_3} \otimes \cdots \otimes A_{t_k t_1}$;
    $\alpha \leftarrow \alpha_{t_1 t_2} + \cdots + \alpha_{t_k t_1}$;
    si ***Temps-de-cycle***(A) = ***Temporisation***(c) **alors**
      si $\alpha /$***Temps-de-cycle***(A) $<$*Débit* **alors**
        $J_{\min} \leftarrow \overrightarrow{\sigma_{t_1 t_2} . \dots . \sigma_{t_k t_1}}$;
        Débit$\leftarrow \alpha/$**Temporisation**(c);
      finsi
    finsi
  finprch
fin

---

dernière boucle est donc $O(\mathcal{C}|\mathcal{P}|^3|\mathcal{T}|)$, où $\mathcal{C}$ est le nombre de circuits élémentaires dans $\mathcal{N}$. Comme le nombre de circuits élémentaires peut être exponentiel en le nombre de transitions ($O(2^{|\mathcal{T}|})$), la complexité est exponentielle dans le pire des cas. Mais, pour la complexité spatiale, la taille est polynomiale en la taille du réseau.

Comparons maintenant la complexité de cet algorithme avec celle de l'algorithme décrit avec le graphe des marquages au plus tôt. La taille du graphe des marquages au plus tôt est exponentielle en la taille du réseau de Petri initial. Plus précisément, la taille est en $O(2^{|\mathcal{P}|})$. Donc la complexité en temps est $O((2^{|\mathcal{P}|})^3) = O(8^{|\mathcal{P}|})$, et la complexité en espace au moins $O(2^{|\mathcal{P}|})$. Observons que ces complexités sont évaluées sans prendre en compte la première transformation du réseau pour rendre toutes les temporisations égales à 1. Cette transformation empire la complexité, temporelle et spatiale. De plus, cet algorithme n'est valable que pour des temporisations rationnelles.

Enfin, remarquons que notre construction donne aussi des précisions sur la période de la politique de routage la pire : puisque le TT-invariant critique associé au circuit critique (avec n jetons) est la somme de n T-invariants minimaux, la période du routage est au plus $n \leq |\mathcal{P}|$ dans chaque place. Avec l'autre algorithme, on ne pouvait que déduire que la période était plus petite que $2^{|\mathcal{P}|}$.

Débit le pire dans les réseaux bornés

Si l'on considère maintenant un réseau à choix libres vivant, k-borné, avec $k \geq 2$, la construction précédente n'est plus valable. Si les temporisations sont rationnelles, le débit le pire est toujours atteint pour un routage périodique, mais la période n'est pas bornée par le nombre de jetons dans un circuit, comme le montre l'exemple suivant.

Exemple 4.32 La figure 4.8(a) représente un réseau à choix libres 2-borné, pour lequel la période du routage le pire dans la place p_1 est plus grande que le nombre de jetons dans ce circuit. De plus, le circuit critique pour ce routage n'est pas élémentaire. Toutes les temporisations et les poids sont égaux à 1. Le routage qui donne le débit le pire est $(t1t4t4t4)^\infty$. L'évolution périodique correspondante est donnée en figure 4.8(b).

Le débit pour ce routage est 0/5, alors que si seule la T-composante de gauche (resp. de droite) était vivante, le débit serait de 2. de plus, la période du routage le pire est 4 alors qu'il n'y a que deux jetons dans le circuit critique. Dans cet exemple, changer les temporisations ou augmenter le nombre de transitions peut aussi augmenter la période du routage le pire arbitrairement. Il semble alors difficile de borner la période du routage le pire, même pour les réseaux 2-bornés.

*
* *

Nous avons montré comment calculer le débit dans des réseaux à choix libres pour des routages périodiques, pour des temporisations rationnelles ou des réseaux 1-bornés. Dans ces cas, on a également donné des algorithmes de calcul des politiques extrémales de routage. Malheureusement, pour des réseaux plus généraux, nous exhibons des exemples prouvant qu'on ne peut pas appliquer ces méthodes. En effet, les réseaux 1-bornés ont des propriétés bien particulières qui permettent d'en faire une analyse plus poussée : ces réseaux sont modélisables par des tas de pièces, ce qui permet d'utiliser la théorie (max,plus) pour l'étudier. De plus, les TT-invariants introduits dans ce chapitre servent à l'analyse de ces réseaux uniquement, puisqu'ils utilisent le fait que deux jetons ne peuvent jamais se dépasser. La table 4.1 résume les différents résultats obtenus dans ce chapitre.

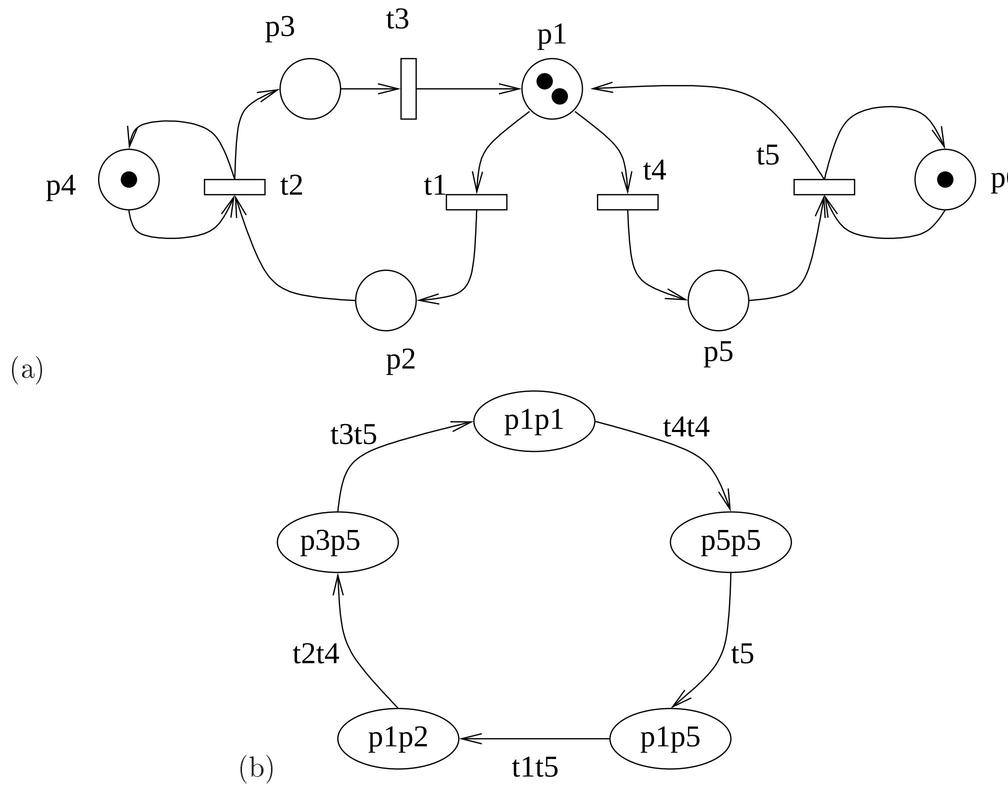


FIG. 4.8 – Réseau à choix libres et l'évolution la pire.

Calcul du débit	$\tau \in \mathbb{Q}_+^{\mathcal{T}}$	$\tau \in \mathbb{R}_+^{\mathcal{T}}$	$\tau \in \mathbb{R}_+^{\mathcal{T}}$; $(\mathcal{N}, M)$ 1-borné
Routage 0-1	Polynomial	Polynomial	Polynomial
Routage périodique	Exponentiel	? ?	Exponentiel
Routages extrémaux	$\tau \in \mathbb{Q}_+^{\mathcal{T}}$	$\tau \in \mathbb{R}_+^{\mathcal{T}}$	$\tau \in \mathbb{R}_+^{\mathcal{T}}$; $(\mathcal{N}, M)$ 1-borné
Pire routage	Périodique	? ?	Périodique de petite période
Meilleur routage	Périodique	Non périodique	Non périodique

TAB. 4.1 – Récapitulatif des résultats.

Chapitre 5

Débit dans les réseaux à choix libres stochastiques

Dans le chapitre précédent, nous avons étudié le débit et l'optimisation de ce débit pour les réseaux avec des temporisations constantes. Dans ce chapitre, nous étudions les réseaux stochastiques, et comparons différentes politiques de résolution de conflits : la politique de compétition, les routages Bernoulli et les routages périodiques. Pour des hypothèses assez générales sur les temporisations, nous montrons l'existence du débit sous ces politiques de résolution de conflits. De plus, lorsque les proportions de routages sont les mêmes, le rapport entre le débit de deux transitions ne dépend pas de la politique choisie. Pour le calcul effectif du débit, nous nous restreignons aux temporisations exponentiellement distribuées. Enfin, nous comparons plusieurs politiques optimales. Les résultats de ce chapitre sont présentés dans [21].

5.1 Existence et proportionnalité du débit dans les réseaux stochastiques

Les réseaux de Petri considérés dans ce chapitre sont toujours à choix libres, vivants, bornés et temporisés avec des temporisations i.i.d. L'existence du débit a déjà été montrée dans [52] pour le routage Bernoulli. Nous allons voir que l'on peut facilement appliquer les théorèmes d'existence pour le routage Bernoulli au cas de la politique de compétition. Pour les routages périodiques, les preuves doivent être adaptées.

Dans cette partie, on suppose toujours les routages équitables. Pour un routage non équitable, le réseau n'est plus vivant, mais on peut appliquer ce qui suit aux composantes fortement connexes de la partie vivante du réseau, pour lesquelles le routage induit est alors équitable.

On fait les hypothèses suivantes sur les temporisations et les routages :

- pour toute transition t, $(\tau_t(n))_{n\in\mathbb{N}}$ est une suite de variables aléatoires i.i.d. ;
- les suites $(\tau_t(n))_{t\in\mathcal{T}}$ sont indépendantes les unes des autres ;
- pour le cas des réseaux routés, les routages sont indépendants des temporisations des transitions et indépendants entre eux.

5.1.1 Existence du débit pour le routage Bernoulli

Dans ce paragraphe, nous rappelons les résultats de [52]. Le théorème suivant assure l'existence des débits pour les réseaux avec des routages Bernoulli.

Théorème 5.1 *Soit $(\mathcal{N}, M, \tau, u)$ un réseau routé avec u un routage Bernoulli équitable. Pour toute transition a, il existe une constante $\gamma_a \in \mathbb{R}_+$ telle que*

$$\lim_{n\to\infty} \frac{X_a(n)}{n} = \lim_{t\to\infty} \frac{t}{\mathcal{X}_a(t)} = \gamma_a \text{ presque sûrement et dans } L_1,$$

où $X_a(n)$ est l'instant de complétion du n-ième tir de a et $\mathcal{X}_a(t)$ est le nombre de tirs de a complétés à la date t.

La proposition suivante montre que le rapport entre le débit des transitions ne dépend que de la fonction de routage et non des temporisations.

Proposition 5.2 *Soit $\lambda_a = \gamma_a^{-1}$, $a \in \mathcal{T}$, le débit de la transition a. Définissons la matrice $R = (R_{ij})_{i,j\in\mathcal{T}}$ comme*

$$R_{i,j} = \begin{cases} \frac{1}{|^\bullet j|} \sum_{p: i\to p\to j} \mathbb{P}(u_p(1) = j) & \text{si } \exists p \in \mathcal{P}, \text{ tel que } i \to p \to j, \\ 0 & \text{sinon.} \end{cases}$$

La matrice R est irréductible, son rayon spectral est 1 et elle admet un unique vecteur $x = (x_a, a \in \mathcal{T})$, $x_a \in \mathbb{R}_+$, $\sum_a x_a = 1$, tel que $xR = x$. Alors il existe $c \in \mathbb{R}_+ \cup \{\infty\}$ tel que $\lambda_a = c.x_a$.

On note $\lambda(u)$ le vecteur des débits pour le routage u.

5.1.2 Existence du débit pour la politique de compétition

On peut se ramener au cas des routages Bernoulli, mais en changeant la distribution des temporisations. Soit $(\mathcal{N}, M, \tau')$ le réseau à choix libres obtenu à partir de $(\mathcal{N}, M, \tau)$ en changeant les temporisations. Pour toute transition a, τ'_a est distribuée de la manière suivante :

$$\mathbb{P}(\tau'_a \leq t) = \mathbb{P}(\tau_a \leq t \mid \forall a' \in (^\bullet a)^\bullet,\ \tau_{a'} \geq \tau_a).$$

Soit u le routage Bernoulli de $\mathcal{N}$ induit par $(\mathcal{N}, M, \tau)_{\text{race}}$. Alors, $(\mathcal{N}, M, \tau)_{\text{race}}$ et $(\mathcal{N}, M, \tau', u)$ ont le même comportement en loi. En effet, le tir d'une transition d'un réseau obéissant à la politique de compétition peut se voir comme le choix de la transition à franchir (c'est celle qui gagne la compétition), puis le tir de la transition choisie. Le temps de tir de la transition choisie a est alors distribué comme τ'_a et la probabilité de choisir la transition a est $\mathbb{P}(\tau_a(1) = \min_{\forall a' \in (^\bullet a)^\bullet}\{\tau_{a'}(1)\}$.

Du théorème 5.1 et de la proposition 5.2, on déduit de l'équivalence des deux réseaux les propositions suivantes.

Proposition 5.3 *Soit $(\mathcal{N}, M, \tau)_{race}$ un réseau à choix libres temporisé obéissant à la politique de compétition. Pour toute transition b, il existe une constante γ_b telle que*

$$\lim_{n\to\infty} \frac{X_b(n)}{n} = \lim_{t\to\infty} \frac{t}{\mathcal{X}_b(t)} = \gamma_b \text{ presque sûrement et dans } L_1.$$

Le débit de la transition b est $\lambda_b = \gamma_b^{-1}$. On note λ_{race} le vecteur des débits pour la politique de compétition.

Proposition 5.4 *Soient $(\mathcal{N}, M, \tau)$ un réseau à choix libres stochastique, u le routage induit par la compétition des transitions de $\mathcal{N}$. Alors, il existe $\alpha \in \mathbb{R}_+^*$ tel que $\lambda_{race} = \alpha\lambda(u)$.*

5.1.3 Routage déterministe périodique

Soient $(\mathcal{N}, M_0)$ un réseau à choix libres vivant borné et u une fonction de routage des places de choix, telle que u_p soit périodique de période d_p pour toute place de choix p.

Existence du débit

Soit $\mathcal{E}_{\text{pér}}$ l'ensemble des états du réseau, c'est-à-dire,

$$\mathcal{E}_{\text{pér}} = \{(M, i_1, \ldots, i_k) \mid M \in \mathcal{R}(M_0), i_j \in \{0, \ldots, d_{p_j} - 1\}\}, \tag{5.1}$$

où $p_1, \ldots, p_k$ sont les places de choix du réseau. L'ensemble des états est fini.

Si l'on bloque une transition b, le marquage atteint est toujours M_b. Supposons que l'état atteint soit (M_b, r). Si l'on franchit à nouveau la transition b une fois et que l'on laisse évoluer le réseau, l'état atteint est (M_b, r'). Mais quelle que soit la manière dont le réseau évolue (c'est-à-dire l'ordre de tir des transitions), c'est toujours le même état qui est atteint (théorème 3.29). En effet, le routage étant défini de manière déterministe et le réseau étant à choix libre, les transitions habilitées le restent jusqu'à ce qu'elles soient franchies. De plus, la suite de tirs pour atteindre l'état (M_b, r') est unique à commutation près des transitions concurrentes ([52], lemme 4.4).

En procédant par blocage/déblocages successifs de b, il existe un état (M_b, r) atteint par un marquage de blocage de b qui est visité infiniment souvent et la suite des transitions franchies entre deux de ces états est unique à commutation près des transitions franchies. On suppose que chaque transition t tire k_t fois entre deux passages successifs dans (M_b, r).

On suppose que (M_b, r) est l'état de départ du réseau de Petri, et on note $(\tilde{\mathcal{N}}, \tilde{M}_b, \tilde{\tau}, \tilde{u})$ l'expansion ouverte de $(\mathcal{N}, M_b, \tau, u)$ et b.

Soit $K = \max\{k \;:\; M_b \xrightarrow{b^k}\}$. L'expansion ouverte de $(\mathcal{N}, M_b, \tau, u)$ et b est :

– $\tilde{\mathcal{P}} = \mathcal{P} \cup \{p_b, p_I\}$;

– $\tilde{\mathcal{T}} = (\mathcal{T} - \{b\}) \cup \{I, b_i, b_o\}$;

– $\tilde{\mathcal{F}} = \left\{ \begin{array}{l} (\mathcal{F} - \{(p,b) \in \mathcal{F}, (b,p) \in \mathcal{F}\}) \\ \cup\{(p, b_o) \;:\; (p,b) \in \mathcal{F}, (b,p) \notin \mathcal{F}\} \\ \cup\{(b_i, p) \;:\; (b,p) \in \mathcal{F}, (p,b) \notin \mathcal{F}\} \\ \cup\{(b_i,p),(p,b_i) \;:\; (b,p) \in \mathcal{F}, (p,b) \in \mathcal{F}\} \\ \cup\{(I,p_I),(p_I,b_i),(b_o,p_b),(p_b,b_i)\} \end{array} \right.$;

– $\tilde{M}_p = \left\{ \begin{array}{ll} M_p & \text{si } p \in \mathcal{P} - {}^\bullet b \\ M_p - K + K\chi_{b^\bullet} & \text{si } p \in {}^\bullet b \\ K & \text{si } p = p_b \\ 0 & \text{si } p = p_I \end{array} \right.$;

– $\tilde{\tau}_a(n) = \left\{ \begin{array}{ll} \tau_a(n) & \text{si } a \in \mathcal{T} - \{b\} \\ \tau_b(n) & \text{si } a = b_i \\ 0 & \text{si } a = b_o \end{array} \right.$,

– $\tilde{u}_p(n) = u_p(n)$.

La figure 5.1 illustre cette construction. Le réseau ainsi construit n'est plus vivant, ni borné. Pour le faire évoluer, il faut définir les instants de tirs de la transition I. Quand la transition d'entrée I est saturée (elle tire un nombre infini de fois à l'instant 0), $\mathcal{N}$ et $\tilde{\mathcal{N}}$ ont le même comportement. On prouve maintenant l'existence du débit de la transition b dans $\tilde{\mathcal{N}}$. On définit ζ_1 comme le vecteur des variables aléatoires $(\tau_1(1), \ldots, \tau_1(k_1), \ldots, \tau_{|\mathcal{T}|}(1), \ldots, \tau_{|\mathcal{T}|}(k_{|\mathcal{T}|}))$ et plus généralement

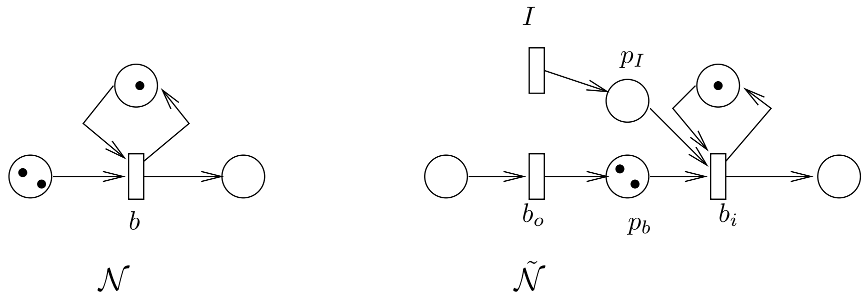


FIG. 5.1 – Expansion ouverte.

$$\zeta_i(\tau_1((i-1)k_1+1),\ldots,\tau_1(ik_1),\cdots,\tau_{|\mathcal{T}|}(((i-1)k_{|\mathcal{T}|}+1),\ldots,\tau_{|\mathcal{T}|}(ik_{|\mathcal{T}|})).$$

Du fait des hypothèses d'indépendance, la suite (ζ_i) est i.i.d. À partir de là, on reprend la démonstration du théorème 5.1 de [52] qui utilise le cadre monotone séparable de [14], ce qui permet d'énoncer le théorème suivant.

Théorème 5.5 *Soit $(\mathcal{N},M,\tau,u)$ un réseau routé avec u un routage périodique équitable. Pour toute transition b, il existe une constante $\gamma_b \in \mathbb{R}_+$ telle que*

$$\lim_{n\to\infty}\frac{X_b(n)}{n}=\lim_{t\to\infty}\frac{t}{\mathcal{X}_b(t)}=\gamma_b \text{ presque sûrement et dans } L_1,$$

où $X_b(n)$ est l'instant de complétion du n-ième tir de b et $\mathcal{X}_b(t)$ est le nombre de tirs de b complétés à la date t.

Rapport entre les débits des transitions

On montre dans cette partie que les routages périodiques donnent, pour les mêmes ratios de routages dans les places de choix, un débit proportionnel à ceux obtenus avec les deux autres politiques de routage, la politique de compétition et les routages Bernoulli.

Proposition 5.6 *Soit $(\mathcal{N},M,\tau,u)$ un réseau routé avec u un routage périodique équitable. Soit $\lambda_a=\gamma_a^{-1}$, $a\in\mathcal{T}$, le débit de la transition a. Définissons la matrice $R=(R_{ij})_{i,j\in\mathcal{T}}$ comme :*

$$R_{i,j}=\begin{cases}\frac{1}{|^\bullet j|}\sum_{p:i\to p\to j}|r_p|_j/d_p & \text{si } \exists p\in\mathcal{P},\ \text{ tel que } i\to p\to j,\\ 0 & \text{sinon,}\end{cases}$$

où $|r_p|_j$ est le nombre de transitions j choisies durant une période de u_p.

La matrice R est irréductible, son rayon spectral est 1 et elle admet un unique vecteur $x=(x_a)_{a\in\mathcal{T}}$, $x_a\in\mathbb{R}_+$, $\sum_a x_a=1$, tel que $xR=x$. Alors $\lambda_a=c.x_a$, $c\in\mathbb{R}_+\cup\{\infty\}$.

Démonstration. On suppose d'abord que λ_a est fini pour toute transition a. Pour une transition a, $\mathcal{X}_a(t)$ est le nombre de tirs finis à la date t. Soit $\mathcal{Y}_{pa}(t)$, avec $a\in\mathcal{T}$, $p\in{}^\bullet a$, le nombre de jetons affectés dans la place p à la transition a jusqu'à la date t. On a

$$\mathcal{X}_a(t)\le\mathcal{Y}_{pa}(t)\le\mathcal{X}_a(t)+\bar{M}, \tag{5.2}$$

où $\bar{M}$ est la borne de $\mathcal{N}$. On a aussi

$$\mathcal{Y}_{pa}(t) = \sum_{i=1}^{K(t)} \mathbf{1}_{\{u_p(i)=a\}}, \qquad K(t) = M_0(p) + \sum_{b \in {}^\bullet p} \mathcal{X}_b(t). \tag{5.3}$$

L'équation 5.2 donne

$$\lambda_a = \lim_t \frac{\mathcal{X}_a(t)}{t} = \lim_t \frac{\mathcal{Y}_{pa}(t)}{t}, \tag{5.4}$$

et l'équation 5.3, en utilisant la périodicité de u_p,

$$\lim_t \frac{\mathcal{Y}_{pa}(t)}{t} = \lim_t \frac{\sum_{i=1}^{K(t)} \mathbf{1}_{\{u_p(i)=a\}}}{K(t)} . \frac{K(t)}{t} = \frac{|r_p|_a}{d_p} . \frac{M_0(p) + \sum_{b \in {}^\bullet p} \mathcal{X}_b(t)}{t}. \tag{5.5}$$

Par l'existence du débit, on a

$$\lambda_a = \frac{|r_p|_a}{d_p} \sum_{b \in {}^\bullet p} \lambda_b. \tag{5.6}$$

Comme l'équation est valable pour tout $p \in {}^\bullet a$, on déduit

$$\lambda_a = \frac{1}{|{}^\bullet a|} \sum_{p \in {}^\bullet a} \frac{|r_p|_a}{d_p} \sum_{b \in {}^\bullet p} \lambda_b = \sum_{b \in {}^\bullet p} \lambda_b R_{ba}, \tag{5.7}$$

d'où $\lambda = \lambda R$.

Par construction, R est positive. Comme $\forall (p,j) \in (\mathcal{P} \times \mathcal{T}) \cap \mathcal{F}$, $|r_p|_a/d_p > 0$, donc pour tout $i, j \in \mathcal{T}$ tel qu'il existe $p \in \mathcal{P}$ telle que $i \to p \to j$, $R_{ij} > 0$. De plus, $\mathcal{N}_{\text{pér}}$ est fortement connexe. Par conséquent, R est irréductible. Alors, par le théorème de Perron-Frobenius, λ est l'unique (à un facteur multiplicatif près) vecteur propre avec des coefficients strictement positifs, et le rayon spectral de R est la valeur propre associée à ce vecteur propre et est donc 1.

S'il existe une transition a telle que $\lambda_a = \infty$ (*i.e.* $\gamma_a = 0$), alors $\lambda = (\infty, \ldots, \infty)$, car R est irréductible. De plus, R dépend seulement de la fréquence de routages des jetons dans une place de routage, et pas des temporisations des transitions. Donc si les temporisations sont remplacées par des temporisations exponentiellement distribuées de paramètre 1, et les paramètres de routages sont choisis de telle sorte que la matrice R n'est pas modifiée, alors les temps de tirs de chaque transition est presque sûrement strictement positifs et $\lambda \in \mathbb{R}_+$. En appliquant la première partie de la preuve, le rayon spectral de R est 1. □

5.1.4 Comparaison des débits entre compétition et routage Bernoulli

Soit $(\mathcal{N}, M, \tau)$ un réseau à choix libres vivant borné temporisé.

Soit p une place de choix et $p^\bullet = \{a_1, \ldots, a_k\}$ les transitions de sortie de p. Nous comparons d'abord les temps de tirs de a_1 dans $(\mathcal{N}, M, \tau)_{\text{race}}$ et dans $(\mathcal{N}, M, \tau, u)$, où u est le routage induit par la politique de compétition (la probabilité de franchir a_i est la même dans les deux cas). On note $P_B(t) = \mathbb{P}(\tau_{a_1} < t)$ la probabilité que le temps de tir de a_1 soit inférieur à t dans $(\mathcal{N}, M, \tau, u)$ et $P_r(t) = \mathbb{P}(\tau_{a_1} < t | \tau_{a_1} < \tau_{a_i},\ i \in \{2, \ldots, k\})$ la probabilité que le temps de tir de a_1 soit inférieur à t dans $(\mathcal{N}, M, \tau)_{\text{race}}$.

Lemme 5.7 *On a* $\forall t \in \mathbb{R}_+,\ P_B(t) \leq P_r(t)$.

Démonstration. Soit $P_B(t) = \mathbb{P}(\tau_{a_1} < t)$ et $P_r(t) = \mathbb{P}(\min(\tau_{a_2}, \dots, \tau_{a_k}) < t)$. On a $P_B(t) = \int_0^t dP_1(u)$ et

$$\begin{aligned} P_r(t) &= \frac{\mathbb{P}(\tau_{a_1} < t \ \wedge \ \tau_{a_1} < \tau_{a_i},\ i \in \{2, \dots, k\})}{\mathbb{P}(\tau_{a_1} < \tau_{a_i},\ i \in \{2, \dots, k\}} \\ &= P_r(t) = \frac{\int_0^\infty \int_0^{t\wedge u} dP_1(v) dP_2(u)}{\int_0^\infty \int_0^u dP_1(v) dP_2(u)}. \end{aligned}$$

Mais

$$\begin{aligned} P_B(t). \int_0^\infty \int_0^u dP_1(v) dP_2(u) &= \int_0^\infty \int_0^u dP_1(v) dP_2(u) \times \int_0^t dP_1(v) \\ &= \int_0^\infty [\int_0^u dP_1(v)][\int_0^t dP_1(v)] dP_2(u) \\ &\leq \int_0^\infty \int_0^{t\wedge u} dP_1(v) dP_2(u) \\ &= P_r(t). \int_0^\infty \int_0^u dP_1(v) dP_2(u), \end{aligned}$$

car

$$\int_0^{t\wedge u} dP_1(v) = \min(\int_0^t dP_1(v), \int_0^u dP_1(v))$$

et

$$\int_0^x dP_1(v) \leq 1 \ \forall x.$$

□

Soit τ_{r,a_1} la variable aléatoire du temps de tir de la transition a_1 pour la politique de compétition et τ_{B,a_1} dans $(\mathcal{N}, M, \tau, u)$ (c'est la même distribution que τ_{a_1}). Le lemme précédent est équivalent à la proposition suivante, par définition des ordres stochastiques.

Proposition 5.8 *Pour toute transition $a \in \mathcal{T}$, $\tau_{r,a}$ est stochastiquement inférieure à $\tau_{B,a}$, ce qui est noté $\tau_{r,a} \leq_{st} \tau_{B,a}$.*

La notion d'ordre stochastique sur les variables aléatoires est développée dans [73] et simplifie la preuve du prochain théorème. On utilisera en effet le lemme suivant.

Lemme 5.9 (Theorem 3.3.11,[73]) *Soient X_i et Y_i, $i \in \{1, \dots, n\}$ des variables aléatoires telles que $\forall i \in \{1, \dots, n\}$, $X_i \leq_{st} Y_i$ et $f : \mathbb{R}^n \to \mathbb{R}$ une fonction croissante. Alors,*

$$f(X_1, \dots, X_n) \leq_{st} f(Y_1, \dots, Y_n).$$

Théorème 5.10 *Soit $(\mathcal{N}, M, \tau)_{race}$ un réseau à choix libres vivant borné stochastique obéissant à la politique de compétition et $(\mathcal{N}, M, \tau, u)$ le réseau routé correspondant. Soient c_{race} et c_{Ber} les débits respectifs correspondant à ces réseaux. Alors, $c_{race} \geq c_{Ber}$.*

Démonstration. Pour toute transition $a \in \mathcal{T}$, $X_a(n)$ est la date de la fin du n-ième tir de la transition a. Soit $\nu_{ap} = \min\{k : \sum_{i=1}^k \mathbf{1}_{\{u_p(i)=a\}} = n\}$ le nombre minimal de jetons arrivés en p tels que n jetons ont été routés vers a. Alors $X_a(n)$ satisfait l'équation ([10])

$$X_a(n) = \left\{ \max_{p \in {}^\bullet a} \left[\min_{\substack{(n_i, i \in {}^\bullet p) : M_p + \\ \sum_{i \in {}^\bullet p} n_i = \nu_{pa}(n)}} \max_{i \in {}^\bullet p} X_i(n_i) \right] \right\} + \tau_a(n),$$

où $\tau_a(n)$ est le temps du n-ième tir de la transition a (le réseau considéré est $(\mathcal{N}, M, \tau)_{\text{race}}$ ou $(\mathcal{N}, M, \tau, u)$, et donc $\tau_a(n)$ est distribuée comme $\tau_{r,a}(1)$ ou comme $\tau_{B,a}(1)$). Ces équations sont satisfaites par $(\mathcal{N}, M, \tau)_{\text{race}}$ et par $(\mathcal{N}, M, \tau, u)$. Quand on les déplie, ces équations peuvent être exprimées en fonction des $\tau_b(i)$, $b \in \mathcal{T}$ seulement. La fonction X_a est croissante, car la fonction

$$\Phi : ((x_{i,n_i})_i, y) \to \left\{ \max_{p \in {}^\bullet a} \left[\min_{\substack{(n_i, i \in {}^\bullet p) : M_p + \\ \sum_{i \in {}^\bullet p} n_i = \nu_{pa}(n)}} \max_{i \in {}^\bullet p} x_{i,n_i} \right] \right\} + y$$

est croissante. De plus, pour calculer X_a, on n'a besoin que d'un nombre fini des $\tau_b(i)$, et ce nombre ne dépend que du routage. En effet, soit $\overrightarrow{\sigma}$ l'image commutative (unique pour un routage donné) des transitions franchies de M à M_a après $n-1$ tirs de a. Le nombre $X_a(n)$ ne dépend que de $\tau_b(1), \ldots, \tau_b(|\overrightarrow{\sigma}|_b)$, $b \in \mathcal{T}$, comme les autres tirs ne peuvent arriver qu'après le n-ième tir de la transition a (voir [52, lemma 4.4]).

D'après le lemme 5.7, pour toute transition b, pour tout $n \in \mathbb{N}^*$, $\tau_{r,b}(n) \leq_{st} \tau_{B,b}(n)$, et comme Φ est croissante d'après le lemme 5.9, $\lambda_{\text{race},a} = \lim_{n\to\infty} \frac{n}{X_{r,a}(n)} \geq \lim_{n\to\infty} \frac{n}{X_{B,a}(n)} = \lambda_{\text{Ber},a}$, et $c_{\text{race}} \geq c_{\text{Ber}}$. $\square$

5.2 Calculs de débit avec des temporisations exponentielles

5.2.1 Politique de compétition

Le cas de la politique de compétition est connu et peut être modélisé par une chaîne de Markov à temps continu, d'espace d'états $\mathcal{R}(M_0)$, l'ensemble des marquages atteignables du réseau. Soit M un marquage atteignable et notons T_M l'ensemble des transitions habilitées en M. La première transition qui tire est $a \in T_M$ avec probabilité $\mu_a / (\sum_{a' \in T_M} \mu_{a'})$. Le temps de tir est exponentiellement distribué de paramètre $\sum_{a' \in T_M} \mu_{a'}$. On peut calculer la distribution stationnaire de ce processus comme le vecteur propre à gauche π_{race} correspondant à la valeur propre 0 de son générateur infinitésimal Q dont les termes sont calculés comme suit. Pour tout $M_1 \in \mathcal{R}(M_0)$, pour tout $a \in T_{M_1}$, $Q_{M_1,M_2} = \mu_a$ si $M_1 \xrightarrow{a} M_2$ et $Q_{M_1,M_1} = -\sum_{M_2 \in \mathcal{R}(M_0)} Q_{M_1,M_2}$. Le débit total est alors donné par la formule

$$c_{\text{race}} = \sum_{M_1 \in \mathcal{R}(M_0)} -\pi_{\text{race}M_1} . Q_{M_1,M_1} .$$

Illustrons ce calcul par un exemple.

Exemple 5.11 La figure 5.2 représente le réseau à choix libres que nous prendrons comme exemple pendant toute la section. Les places sont notées par des lettres $(a, \ldots, e)$ et les transitions par des chiffres $(1, \ldots, 5)$. Les paramètres des temporisations exponentiellement distribuées sont respectivement $\mu_1 = 2$, $\mu_2 = 2$, $\mu_3 = 3$, $\mu_4 = 5$ et $\mu_5 = 1$. L'espace des marquages atteignables est $\mathcal{R} = \{\{a,d\}, \{a,e\}, \{b,d\}, \{b,e\}, \{c,d\}, \{c,e\}\}$. Le générateur infinitésimal de la chaîne de Markov à temps continu est alors

$$Q = \begin{pmatrix} -2 & 0 & 2 & 0 & 0 & 0 \\ 1 & -3 & 0 & 2 & 0 & 0 \\ 2 & 0 & -5 & 0 & 3 & 0 \\ 0 & 2 & 1 & -6 & 0 & 3 \\ 0 & 5 & 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} .$$

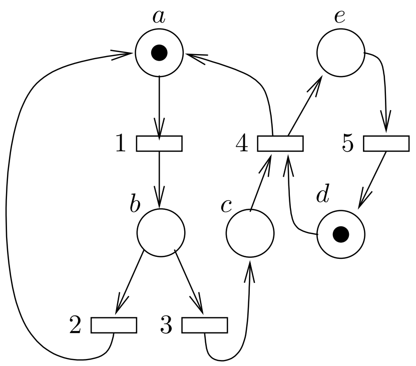


FIG. 5.2 – Réseau de Petri de l'exemple 5.11.

La distribution stationnaire de ce processus est le vecteur propre unitaire associé à la valeur propre 0 de Q, c'est-à-dire,

$$\pi_{\text{race}} = \frac{1}{337}(85, 90, 40, 30, 42, 90).$$

Le débit total du réseau est alors $c_{\text{race}} = \sum_i -\pi_i.Q_{ii} = 1120/377 \approx 2.97$.

5.2.2 Routage Bernoulli

Dans le cas des routages Bernoulli, l'évolution ne peut plus être modélisée par une chaîne de Markov sur l'espace des marquages atteignables. Une solution pour pallier ce problème est de rajouter une transition de temporisation nulle avant chaque transition conflictuelle. Cela mène à un processus semi-markovien sur l'espace des marquages du nouveau réseau. Une autre solution est de travailler directement sur le réseau considéré, mais de prendre un espace d'états étendu. On obtient alors un processus de Markov (chaîne de Markov à temps continu), qu'on peut ensuite étudier pour optimiser le routage.

Intéressons-nous tout d'abord à l'espace des états. Quand un jeton entre dans une place de choix, le temps de tir de la transition choisie dépend du routage (contrairement au cas de la politique de compétition). Si l'on ne prenait comme espace d'états que l'ensemble des marquages atteignables, on serait confronté à des difficultés dues à cette dépendance. En vue de séparer les routages et les temporisations, on considère un nouvel espace d'états : quand un jeton entre dans une place de choix, la transition qu'il peut franchir est déjà définie. Pour un marquage $M \in \mathcal{R}(M_0)$, on note $\mathcal{T}(M)$ l'ensemble des ensemble maximaux de transitions qui peuvent être franchies simultanément (dans le réseau non temporisé) avec le marquage M. L'espace d'états étendu est alors

$$\mathcal{E}_{\text{Ber}} = \{(M, T) \mid M \in \mathcal{R}(M_0), T \in \mathcal{T}(M)\}, \tag{5.8}$$

c'est-à-dire qu'un état correspond à un marquage atteignable et les transitions qui peuvent être franchies avec le routage.

Définissons maintenant le générateur infinitésimal de la chaîne de Markov.

Soit $(M_1, T_1) \in \mathcal{E}_{\text{Ber}}$ un état, soit $a \in T_1$. La transition a est franchie avec un taux μ_a et le nouvel ensemble des transitions habilités est $T_2 = (T_1 \setminus \{a\}) \cup T'$ où T' est l'ensemble des transitions nouvellement habilitées. Pour chaque place de choix p qui a un jeton de plus, $a' \in p^\bullet$, $a' \in T'$ avec probabilité $\mathbb{P}(u_p(1) = a')$ et si une transition non-conflictuelle a' est nouvellement habilitée, $a' \in T'$. Alors, le générateur infinitésimal de (M_1, T_1) à (M_2, T_2) est

$$Q_{(M_1,T_1),(M_2,T_2)} = \begin{cases} \sum_{a \in T_1 : M_1 \xrightarrow{a} M_2} \mu_a \prod_{a' \in T_2 \setminus (T_1 - \{a\})} \mathbb{P}(u_p(1) = a') & \text{si } T_1 \setminus \{a\} \subseteq T_2, \\ 0 & \text{sinon, et} \\ -\sum_{a \in T_1} \mu_a & \text{si } (M_1, T_1) = (M_2, T_2). \end{cases}$$

On note π_{Ber} la distribution stationnaire de ce processus ($\pi_{\text{Ber}} = \pi_{\text{Ber}} Q$).

Exemple 5.12 Considérons de nouveau le réseau de la figure 5.2. Un jeton arrivant dans la place b franchit la transition 2 avec probabilité p et la transition 3 avec probabilité $1-p$. L'espace d'états est $\{(ad, 1), (ae, 1, 5), (bd, 2), (bd, 3), (be, 2, 5), (be, 3, 5), (cd, 4), (ce, 5)\}$, et la matrice de transition

$$Q = \begin{pmatrix} -2 & 0 & 2p & 2(1-p) & 0 & 0 & 0 & 0 \\ 1 & -3 & 0 & 0 & 2p & 2(1-p) & 0 & 0 \\ 2 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 & 0 & 3 & 0 \\ 0 & 2 & 1 & 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -4 & 0 & 3 \\ 0 & 5 & 0 & 0 & 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}.$$

Pour avoir les mêmes probabilités de routage que pour la politique de compétition, prenons $p = 2/5$. La probabilité stationnaire est alors

$$\pi_{\text{Ber}} = \frac{1}{1184}(190, 270, 100, 141, 72, 108, 222, 81),$$

et le débit est $c_{\text{Ber}} = 1480/573 \approx 2.58$.

Optimisation des paramètres de Bernoulli

Pour l'instant, on a comparé le débit des réseaux avec la politique de compétition et des réseaux avec routages Bernoulli où les proportions de routage sont les mêmes. Dans ce paragraphe, nous étudions la dépendance du débit en fonction des paramètres de routages. Quand il n'y a qu'une place de routage, il existe des algorithmes efficaces pour calculer le routage qui optimise le débit. Nous verrons aussi un autre algorithme dans le cas général.

Optimisation d'un seul paramètre On suppose que tous les paramètres de routages sont fixés, sauf un, que l'on note p. Le temps de séjour dans un état ne dépend pas des paramètres de routages. Donc seule la distribution stationnaire de la chaîne de Markov incluse aux dates de sauts dépend de p.

Soit $Q(p)$ le générateur infinitésimal de la chaîne de Markov. Les coefficients de $Q(p)$ sont des polynômes de degré au plus 1. Pour tout $p \in]0, 1[$, $Q(p)$ est le générateur infinitésimal d'une chaîne de Markov à temps continu irréductible sur un espace d'états fini. L'espace propre associé à la valeur propre 0 est donc de dimension 1 et $Q(p)$ est le rang $r - 1$, où r est la dimension de la matrice $Q(p)$. La probabilité stationnaire $\pi(p)$ est la solution de l'équation

$$x.Q(p) = 0, \tag{5.9}$$

et toute solution de cette équation est proportionnelle $\pi(p)$.

Le théorème suivant ([72], Theorem 6.1) donne la complexité d'un algorithme récursif qui calcule la solution de $x.Q(p) = 0$ en temps cubique.

Théorème 5.13 *Soit $M \in F[p]^{n\times m}$ une matrice de rang r dont le degré en p des coefficients de la matrice est borné par d. Il existe un algorithme résolvant le système $x.M = 0$ en $O(nmrd^2)$ opérations élémentaires.*

Appliqué à notre problème, si n est la taille de l'espace d'états défini en (5.8), le coût pour résoudre $xQ(p) = 0$ est $O(n^2(n-1)) = O(n^3)$. Remarquons que l'espace d'états peut être exponentiellement grand comparé à la taille du réseau de Petri initial : il est plus grand que l'ensemble des marquages atteignables. La solution donnée par l'algorithme est un vecteur de polynômes dont les degrés sont au plus n. En fait, on peut aussi montrer que le degré maximum est au plus le nombre de lignes dans lequel le paramètre p apparaît.

Notons aussi que la complexité de cet algorithme de calcul de $\pi(p)$ est la même que pour résoudre un système linéaire avec des coefficients réels. Une méthode plus naïve aurait mené à une complexité en $O(n^6)$: comme il est possible de borner les degrés des polynômes d'une solution x, de l'équation 5.9, on peut écrire $x(p)$ comme $x_0 + x_1.p + \cdots + x_d.p^d$, où d est le degré maximal d'un coefficient de $x(p)$ et x_i sont des vecteurs réels. Maintenant, $Q(p)$ peut s'écrire en $P_0 + P_1.p$, où P_0 et P_1 sont des matrices réelles. En identifiant les coefficients, le système peut se résoudre comme un système linéaire à nd équations, ce qui donne une complexité en temps de $O((n.d)^3)$.

La probabilité stationnaire est le vecteur normalisé de cette solution : c'est une fraction rationnelle, de degré maximum n pour le numérateur et le dénominateur. Le débit est une fonction linéaire de $\pi(p)$. Alors, maximiser le débit consiste à étudier une fraction rationnelle entre 0 et 1 et trouver son maximum.

Exemple 5.14 On calcule le débit du réseau figure 5.2 avec routage Bernoulli en fonction de p. On trouve :

$$\pi_{\text{Ber}}(p) = \frac{1}{4p^2-17p+18}\left(\frac{2p+3}{2}, \frac{9(p-1)}{2}, \frac{5p}{2}, \frac{9-p^2-8p}{4}, 3p(1-p), 3(1-p)^2, \frac{4p^2-13p+9}{2}, \frac{9(p^2-2p+1)}{4}\right),$$

ce qui donne pour le débit $c_{\text{Ber}}(p) = \frac{60(4p^2-17p+18)}{139p^2-403p+414}$. Alors, on trouve que le débit maximal est atteint pour $p = \frac{846-30\sqrt{615}}{751} \approx 0.14$ et le débit maximal est approximativement $2,61$.

Notons que dans cet exemple, le débit maximum atteint par un routage Bernoulli est encore inférieur au débit atteint par la politique de compétition. Mais ce n'est pas toujours le cas. Pour le réseau de Petri donné en figure 5.3 (avec les paramètres des temporisations donnés en gras sur la figure), le débit atteint par un routage Bernoulli peut être bien meilleur que pour la politique de compétition. Avec la politique de compétition, on a $c_r \approx 2.07$. Si les paramètres de routages Bernoulli dans les places c et e sont les mêmes que pour la politique de compétition et si p est la probabilité de router les jetons dans a vers la transition 1, le débit est $c_{\text{Ber}}(p) = \frac{2300}{1400-1089p}$. Le débit maximal est atteint quand p est proche de 1, et le débit total est $c_{\text{Ber}}(1) \approx 7.40$.

Cas général Quand il y a plus d'un paramètre, il n'y a plus d'algorithme efficace connu pour calculer le débit. Mais la méthode plus naïve est toujours valable : il existe une solution de (5.9) telle que le degré de chaque coefficient est au plus n (car le degré des coefficients de la matrice est au plus 1). En identifiant les coefficients de même degré dans l'équation, l'équation peut être résolue comme un système linéaire réel à $n(n+1)^m$ équations, où m est le nombre de paramètres.

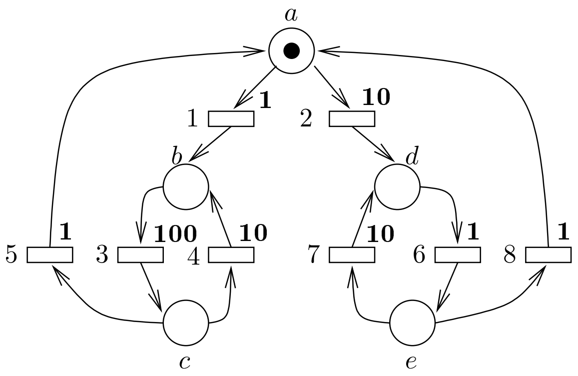


FIG. 5.3 – Exemple d'un réseau de Petri où le débit maximum avec les routages Bernoulli est meilleur que pour la politique de compétition.

La complexité est alors $O(n^{3m+3})$. Le temps de calcul peut devenir très grand dès que l'espace d'états augmente.

5.2.3 Débit pour les routages périodiques

On numérote les places du réseau : $\mathcal{P} = \{p_1, \ldots, p_k\}$ et les p_i ont un routage périodique de périodes respectives d_i, $i \in \mathbb{N}_k$. Alors, l'espace d'états est au pire $|\mathcal{R}(M_0)|.d_1 \ldots d_k$.

On peut modéliser ce réseau par une chaîne de Markov à temps continu de générateur infinitésimal Q de la manière suivante :

- $(M_0, 0, \ldots, 0)$ est un état ;
- pour tout état $e_1 = (M, r_1, \ldots, r_k)$, si t est une transition habilitée en sortie des places $p_{i_1}, \ldots, p_{i_\ell}$ alors on peut atteindre l'état $e_2 = (M', r'_1, \ldots, r'_k)$ si $M \xrightarrow{t} M'$ et si pour tout $j \in \{i_1, \ldots, i_\ell\}$, $r'_j = (r_j + 1) \mod p_j$ et pour tout $j \notin \{i_1, \ldots, i_\ell\}$, $r'_j = r_j$. Alors, $Q_{e_1,e_2} = \mu_t$.

Le nombre d'états devient rapidement très grand quand la longueur des périodes augmente. Des calculs en Maple pour le réseau de la figure 5.2 donnent les débits représentés en figure 5.4. Les débits maximaux pour une même proportion de routage sont atteints pour les routages périodiques *équilibrés*. On dit qu'un routage est équilibré en une place p si quand on prend deux facteurs de même longueur de la suite u_p, la différence du nombre d'occurrence d'une transition dans ces sous-suites diffère au plus de 1. Autrement dit, l'apparition des transitions est la plus régulière possible. La figure 5.5 compare les débits pour des routages périodiques équilibrés et des routages Bernoulli, avec les mêmes proportions de routage. Dans cet exemple, le routage équilibré est meilleur dans tous les cas que le routage Bernoulli.

Nous allons maintenant étudier différents routages optimaux. Suivant les choix que l'on autorise, on obtient plusieurs routages optimaux, que nous pouvons comparer.

5.2.4 Politiques qui maximisent le débit

Dans ce paragraphe, on étudie des politiques qui maximisent le débit Une solution pour le routage optimal est de se ramener à un problème de décision de Markov. Pour simplifier l'exposé, on se limite au cas des réseaux à choix libres vivants 1-bornés, mais les résultats sont aussi valables pour les réseaux non 1-bornés.

Puisque les temporisations sont exponentielles, c'est-à-dire sans mémoire, pour un état, la

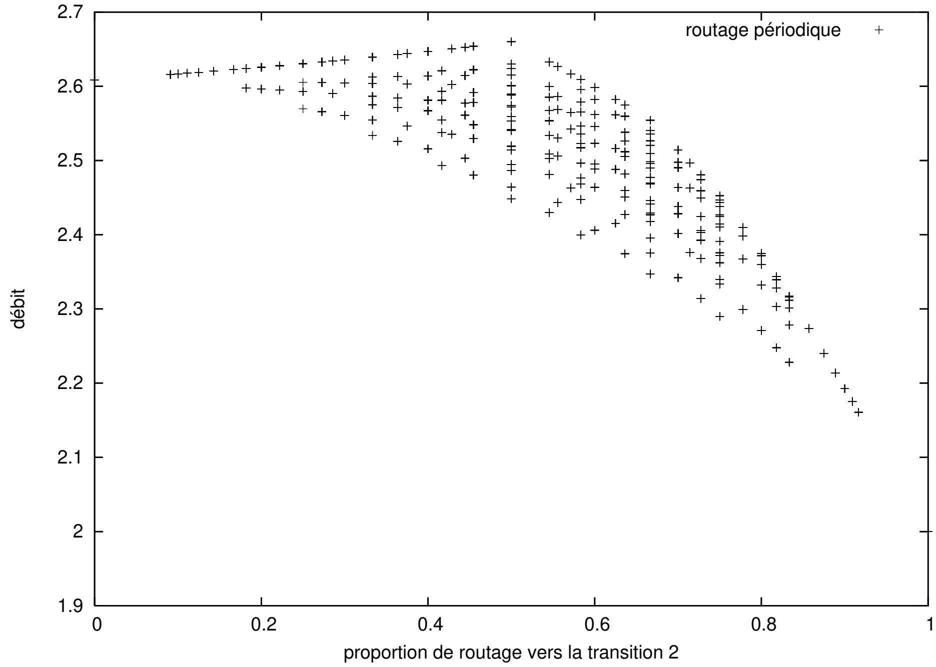


FIG. 5.4 – Débit pour les routages périodiques. En abscisse est représenté la proportion de routage vers la transition 22. Pour une abscisse donnée, il y a plusieurs débits, puisque plusieurs périodes possibles. Pour chaque abscisse représentée, le point donnant le débit le plus élevé correspond au routage équilibré pour cette proportion.

décision à prendre ne dépend que de l'état courant, donc la décision optimale pour un état est toujours la même et est stationnaire. Une fois ces décisions connues, trouver le débit revient à calculer une probabilité stationnaire d'une chaîne de Markov à espace d'état fini et irréductible. Cette probabilité existe, et donc le débit existe. Définissons maintenant plus précisément le problème de décision de Markov étudié.

Les instants de tir potentiel pour une transition a sont donnés par un processus de Poisson de taux μ_a. À chaque instant de tir potentiel, le tir de la transition peut effectivement avoir lieu si et seulement si la transition est habilitée. Dans ce dernier cas, un « contrôleur » choisit de tirer ou non la transition. Le but du contrôleur est de maximiser le débit total.

Comme il est plus aisé de travailler en temps discret pour les processus de décision de Markov, on uniformise la superposition des processus de Poisson, ce qui revient à remplacer ce processus par une suite i.i.d. de variables aléatoires dans $\mathcal{T}$ et de distribution $(\mu_a/\Lambda)_{a\in\mathcal{T}}$, avec $\Lambda = \sum_a \mu_a$. Maintenant, à chaque unité de temps, une transition a la possibilité de tirer. Ce tir a lieu

1. si cette transition est habilitée
2. et si le contrôleur est d'accord.

Le gain immédiat est de 1 si la transition tire et de 0 sinon.

Le gain cumulé à partir de l'étape k pour chaque état peut se calculer en fonction du gain cumulé à partir de l'étape $k+1$.

Préallocation optimale des jetons

Supposons qu'un jeton entre dans une place de choix au temps n. Le contrôleur doit choisir immédiatement une des transitions de sortie. L'information disponible est le marquage courant

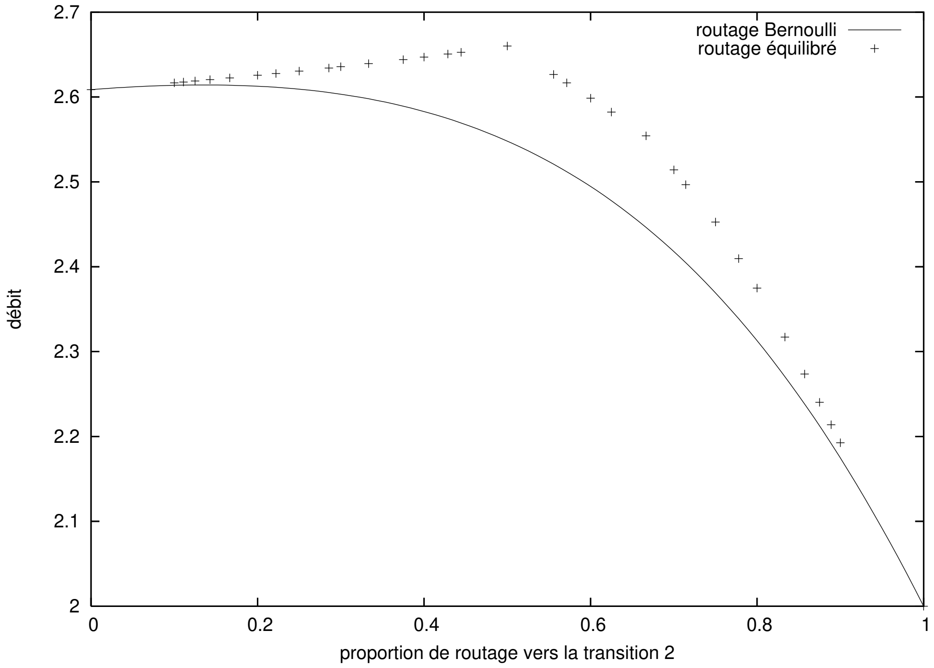


FIG. 5.5 – Comparaison des débits entre les routages Bernoulli et les routages périodiques équilibrés. Les routages équilibrés donnent un débit plus élevé que les routages Bernoulli avec les mêmes proportions de routage. Les points obtenus pour les routages équilibrés semblent former une courbe comparable à celle obtenue pour des routages équitables avec des temporisations constantes ([25]).

et les décisions déjà prises. Le jeton tirera la transition quand la transition choisie sera potentiellement habilitée. Mais le contrôleur ne sait pas quelle est la première transition qui sera potentiellement habilitée. C'est ce que l'on appelle la *préallocation sur les jetons*. Les routages Bernoulli et périodiques en sont des cas particuliers.

Comme pour le routage Bernoulli, il est nécessaire de « garder en mémoire » les décisions de routage dans l'état du réseau. L'espace des états est donc $\mathcal{E}_{\text{Ber}}$, le même espace que pour le routage Bernoulli.

Un état est caractérisé par un marquage et les transitions habilitées (le routage étant effectué). Entre l'étape k et l'étape $k+1$, il y a un gain de 1 pour l'état (M, ρ) si le marquage à l'étape $k+1$ est $M + \vec{t}C$, $t \in \rho$ et où C est la matrice de transition du réseau. Alors, les transitions habilitées ne changent pas pour les jetons qui n'ont pas bougé et une transition est choisie par jeton placé dans une place en sortie de t. La transition t n'est plus habilitée.

La probabilité de passer du marquage M au marquage $M + \vec{t}C$ est μ_t/Λ. Soit $\text{rout}(M, \rho)$ l'ensemble des décisions de routage que l'on peut prendre après franchissement d'une transition franchissable à partir de (M, ρ) : un élément de cet ensemble est un ensemble de transitions nouvellement habilitées et pour une place de choix nouvellement marquée, il y a exactement une transition en sortie de cette place de choix. Il existe un élément de $\text{rout}(M, \rho)$ qui maximise l'espérance des gains (le réseau est à choix libres, donc les décisions des places sont indépendantes les unes des autres). Pour une décision de routage $r \in \text{rout}(M, \rho)$, l'espérance du gain est

$$J^k(M, \rho) = \sum_{t \in \rho} \left[(J^{k+1}(M + \vec{t}C, (\rho \setminus \{t\}) \cup t^{\bullet\bullet} \cap r) + 1) \frac{\mu_t}{\Lambda} + \frac{\Lambda - \sum_{t \in \rho} \mu_t}{\Lambda} J^{k+1}(M, \rho) \right].$$

La politique optimale consiste donc à maximiser l'espérance du gain, soit,

$$J^{k+1}(M, \rho) = \max_{r \in \text{rout}(M,\rho)} \sum_{t \in \rho} \left[(J^k(M + \vec{t}C, (\rho \setminus \{t\}) \cup t^{\bullet\bullet} \cap r) + 1) \frac{\mu_t}{\Lambda} + \frac{\Lambda - \sum_{t \in \rho} \mu_t}{\Lambda} J^k(M, \rho) \right]. \quad (5.10)$$

D'après le théorème 1.14, il existe $\lambda^{\text{pré}}$, un gain moyen optimal et h, une politique optimale tel que

$$\lambda^{\text{pré}} + h(M, \rho) = \max_{r \in \text{rout}(M,\rho)} \sum_{t \in \rho} \left[(h(M + \vec{t}C, (\rho \setminus \{t\}) \cup t^{\bullet\bullet} \cap r) + 1) \frac{\mu_t}{\Lambda} + \frac{\Lambda - \sum_{t \in \rho} \mu_t}{\Lambda} h(M, \rho) \right], \quad (5.11)$$

et $\lambda^{\text{pré}}$ est le gain moyen optimal. Le débit du réseau est le gain moyen optimal multiplié par la constante d'uniformisation. Par itération des valeurs, on trouve $\lambda^{\text{pré}}$. Dans l'exemple, la préallocation optimale est : choisir l'état $(b, e, 2)$ et router les jetons vers la transition 2 dans l'état (a, e) et choisir l'état $(b, d, 3)$ et router les jetons vers la transition 3 dans l'état (a, d). Le débit trouvé est alors 2,8.

Allocation optimale des jetons

Dans chaque place de choix marquée, le jeton est alloué à une transition de sortie de cette place, mais à chaque instant, l'allocation peut être modifiée par le contrôleur.

1. Le contrôleur connaît le marquage courant.
2. Il ne sait pas quelle transition va être habilitée.

On appelle cette politique *allocation des jetons*. La préallocation est un cas particulier de cette politique.

De nouveau, on peut résoudre le problème de trouver l'allocation optimale par un processus de décision de Markov, mais cette fois le contrôle ne dépend plus que du marquage atteint, puisque les choix peuvent être remis en cause à chaque tir de transition.

L'espace d'états du processus de décision de Markov est donc l'ensemble des marquages atteignables, et l'équation permettant de trouver le gain moyen optimal est :

$$\lambda^{\text{alloc}} + h(M) = \max_{r \in \mathcal{T}(M)} \sum_{t \in r} \left[(h(M + \vec{t}C) + 1)\frac{\mu_t}{\Lambda} + \frac{\Lambda - \sum_{t \in r} \mu_t}{\Lambda} h(M) \right], \tag{5.12}$$

où $\mathcal{T}(M)$ est l'ensemble des ensembles maximaux des transitions qui peuvent être franchies simultanément dans le réseau de Petri non temporisé et dans le marquage M. En effet, pour un ensemble de décisions $r \in \mathcal{T}(M)$, la transition $t \in r$ est franchie avec probabilité μ_t/Λ et le gain est alors de 1. Sinon, avec une probabilité $1 - \sum_{t \in r} \mu_t/\Lambda$, aucune transition n'est franchie et le gain à cette étape est nul.

Dans l'exemple, l'allocation optimale est : router vers la transition 2 dans l'état (b, e) et vers la transition 3 dans l'état (b, d). Le débit trouvé est alors 3,05.

Le débit est meilleur que celui obtenu avec la politique précédente. En effet, les politiques possibles avec la méthode précédente sont strictement incluses dans celles possibles avec cette méthode.

Politique optimale non-anticipative

À chaque instant, si la transition qui peut potentiellement tirer est habilitée, le contrôleur peut choisir de franchir cette transition ou non. L'information disponible est le marquage courant. On peut aussi voir ce modèle comme un modèle où le contrôleur peut réallouer le jeton à chaque instant connaissant

1. le marquage courant ;
2. la transition qui devient potentiellement habilitée.

C'est l'information maximale dont on peut disposer si l'on ne connaît aucun des événements à venir. On appelle une telle politique *non-anticipative*. L'allocation des jetons est clairement une politique non-anticipative, ainsi que la politique de compétition.

Pour une politique stationnaire, dans un marquage M, il existe un ensemble $T(M)$ de transitions que le contrôleur accepte de franchir et un ensemble $\bar{T}(M)$ de transitions que le contrôleur refuse de franchir. La politique non-anticipative revient alors, pour le marquage M, à une politique de compétition entre les transitions de $T(M)$.

Une fois encore, on peut trouver la politique optimale par un processus de décision de Markov dont l'espace d'états est l'ensemble des marquages atteignables. On a

$$\lambda^{\text{opt}} + h(M) = \max_{r \in \mathcal{S}_T(M)} \sum_{t \in r} \left[(h(M + \vec{t}C) + 1)\frac{\mu_t}{\Lambda} + \frac{\Lambda - \sum_{t \in r} \mu_t}{\Lambda} h(M) \right], \tag{5.13}$$

où $\mathcal{S}_T(M)$ est l'ensemble des sous-ensembles des transitions habilitées en M qui contiennent au moins une transition de chaque cluster qui contient une transition habilitée.

Le débit obtenu est de 3,2. Dans le marquage (b, e), on route vers la transition 2, et dans le marquage (b, d), on fait jouer la politique de compétition entre les transitions 2 et 3.

Les différentes politiques optimales peuvent s'interpréter d'une manière alternative. En effet, pour le premier cas, la décision de routage se fait dès l'arrivée d'un jeton dans la place de choix, et ce choix est irrévocable jusqu'au tir de la transition habilitée. Pour le deuxième cas, on peut remettre en cause cette décision à chaque changement d'état (s'il n'y a pas de changement d'état, se tenir à la décision optimale est la meilleure chose à faire, puisque les temps de tirs sont exponentiellement distribués). Ainsi, cette deuxième politique consiste à retarder le moment de la décision de routage. Mais, dès que le tir d'une transition est terminé, la transition tire. Enfin, dans le dernier cas étudié, le choix de la transition est encore retardé : on attend de savoir quelle est la première transition qui peut être franchie pour décider si on franchit ou non cette transition. La politique optimale non-anticipative consiste alors à choisir un ensemble de transitions dont on accepte le tir. Cette politique de résolution de conflits consiste donc à prendre une décision de tir au dernier moment, tout pouvant toujours être remis en cause. Ainsi, on voit sur l'exemple des temporisations exponentielles, que plus la décision de routage est tardive, ou plus on a d'information disponible sur le système, meilleur est le débit. Cette dernière classe de résolution de conflits contient à la fois les politiques d'allocation des jetons, $T(M)$ contient une et une seule transition de chaque cluster habilité pour chaque marquage M, et la politique de compétition, $T(M)$ contient toutes les transitions habilitées en M. La politique optimale non-anticipative est donc meilleure que ces politiques.

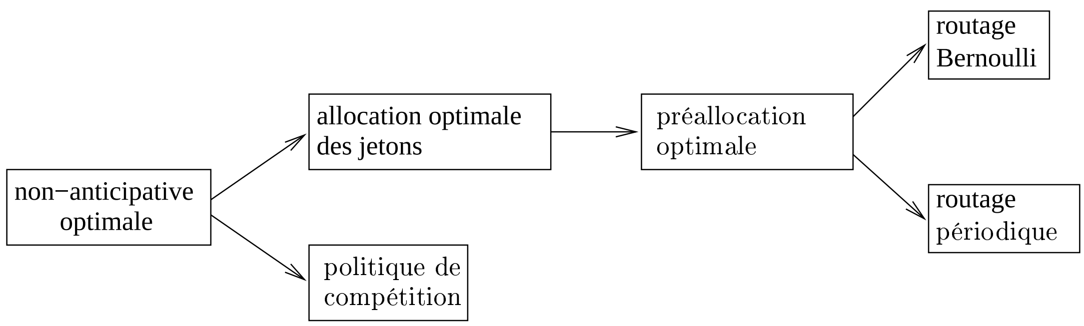
5.3 Comparaison entre les politiques de routage

Dans le deuxième exemple, le routage Bernoulli optimal étant atteint pour un routage 0-1, tous les routages optimaux sont les mêmes.

Politique de routage	Débit exemple 1	Débit exemple 2
politique de compétition	2,97	2,07
Bernoulli	2,61	7,40
périodique	2.66	7,40
préallocation optimale	2,8	7,40
allocation optimale	3,05	7,40
non-anticipative optimale	3,2	7,40

La figure 5.6 représente pour chacune des optimisations décrites les choix effectués en chaque marquage.

On a déjà prouvé que l'on a le diagramme suivant (les flèches signifie « a un débit supérieur à ») :



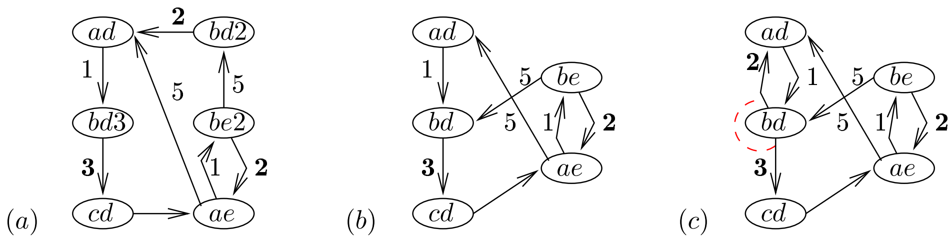


FIG. 5.6 – Politiques optimales pour le réseau de la figure 5.2. Les numéros sur les flèches sont les numéros des transitions franchies. Les numéros en gras correspondent aux décisions optimales prises. En (a) on a représenté la préallocation optimale des jetons. Les états ont donc en mémoire les choix effectués avant franchissement. En (b) on a représenté l'allocation optimale des jetons, et en (c), la politique optimale non-anticipative. L'arc de cercle en pointillé signifie qu'on franchit la première transition possible entre les transitions 2 et 3 dans le marquage bd.

L'exemple de la figure 5.7 montre que la politique de compétition n'est pas comparable avec la politique optimale sur les marquages : on sait déjà, par l'exemple étudié, que le routage optimal sur les marquages peut donner un débit meilleur que la compétition. Mais cet exemple montre que l'inverse peut aussi être vrai.

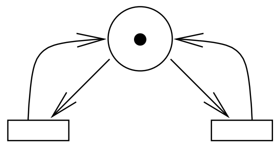
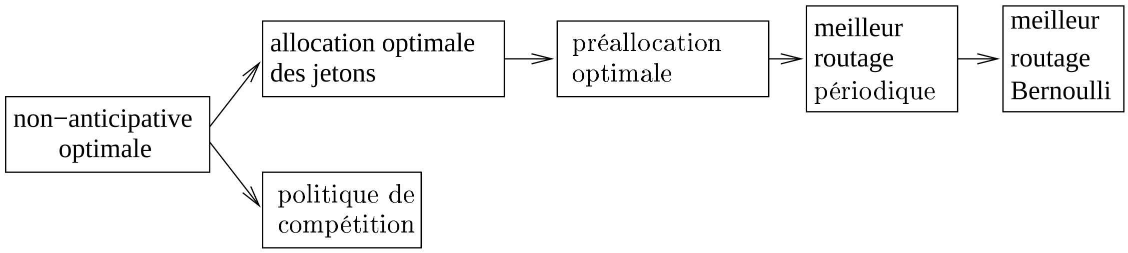


FIG. 5.7 – Réseau pour lequel la politique de compétition est meilleure que le routage optimal sur les marquages : le routage optimal consiste à franchir toujours la même transition. Le débit est donc l'inverse de l'espérance du temps de tir d'une des deux transitions. Pour la compétition, le débit est l'inverse de l'espérance du minimum des temps de tirs.

De plus, dans l'exemple étudié, le routage périodique optimal est meilleur que le routage Bernoulli optimal (voir figure 5.5), ce qui donne :



*
* *

Dans ce chapitre, nous avons comparé les différentes politiques de résolution des conflits pour des réseaux de Petri stochastiques. Pour les classes de routages étudiées, nous avons montré que l'existence du débit et que le rapport entre les débits des différentes transitions ne dépendait que des proportions de routage.

Les calculs peuvent être effectués pour des temporisations exponentielles, ce qui nous permet d'établir, au moins sur certains exemples une hiérarchie de ces différents routages. Toutefois, la comparaison entre routage périodique optimal et routage Bernoulli optimal n'a pas été effectuée. Dans le cas où il n'y a qu'une place de choix, les routages périodiques équilibrés semblent de bons candidats pour être les routages périodiques optimaux.

Chapitre 6

Couplage arrière dans les réseaux de Petri

Dans ce chapitre, nous étudions la simulation parfaite des réseaux à choix libres. Lors de l'analyse de réseaux dont le nombre d'états est grand, les calculs peuvent s'avérer très complexes. De plus, nous avons vu, dans les chapitres précédents, comment calculer le débit dans les réseaux à choix libres dans certains cas. Les calculs sont possibles dans deux cas : quand les temporisations sont constantes et quand elles sont exponentiellement distribuées. Pour des réseaux plus généraux, on peut prouver l'existence du débit, mais pas le calculer. La solution pour étudier de tels systèmes est alors de faire des simulations pour approximer le comportement des réseaux, par exemple, pour trouver le comportement stationnaire des réseaux. La simulation parfaite, qui a été introduite par Propp et Wilson dans les années 1990 ([80, 81]), permet d'obtenir des échantillons indépendants ayant pour loi commune la distribution stationnaire d'une chaîne de Markov. Ces techniques ont d'abord été développées pour l'étude de la physique statistique ([76]), pour laquelle les systèmes sont constitués d'un très grand nombre de particules. Nous appliquons ces techniques au cas des réseaux à choix libres. De plus, pour le cas des graphes d'événements, nous montrons comment améliorer les calculs en exhibant des états extrémaux, qui sont en fait les marquages de blocage. Ensuite, nous appliquons la simulation parfaite au cas des tas de pièces, et aux matrices (max,plus). Comme il est possible de modéliser les graphes d'événements comme un système (max,plus), nous avons une deuxième méthode d'analyse des graphes d'événements. Nous comparons donc ces deux méthodes (nombre d'étapes de simulations et temps de simulation). Enfin, nous appliquons les techniques de couplage arrière des tas de pièces pour la simulation parfaite des réseaux à choix libres 1-bornés avec des temporisations plus générales.

Les résultats de ce chapitre ont été déposés en tant que rapport de recherche du LIP, [19].

6.1 Simulation parfaite des chaînes de Markov

6.1.1 Simulation des chaînes de Markov

Nous allons voir dans cette section comment générer des valeurs distribuées selon la probabilité stationnaire d'une chaîne de Markov.

Génération de variables aléatoires

Lorsqu'on connaît la fonction de distribution d'une variable aléatoire, il existe des techniques simples pour générer une variable aléatoire selon cette distribution. On suppose que l'on dispose d'un générateur de nombres aléatoires qui permet de générer des suites $U_1, U_2, \ldots$ de variables aléatoires indépendantes et uniformément distribuées sur $[0, 1[$. La méthode la plus simple pour générer une variable aléatoire selon une distribution donnée est la *méthode de l'inverse* (voir [55, partie 4.11], par exemple).

Théorème 6.1 *Soit F une fonction de répartition et U une variable aléatoire uniformément distribuée sur* $[0, 1[$.
- *Si F est une fonction continue, la variable aléatoire $X = F^{-1}(U)$ est distribuée selon F.*
- *Si F est la fonction de répartition d'une variable aléatoire à valeurs dans $\mathbb{N}$, alors la variable aléatoire X donnée par*

$$X = k \Leftrightarrow F(k-1) \leq U < F(k)$$

est distribuée selon F.

Exemple 6.2 Prenons l'exemple d'une variable aléatoire exponentiellement distribuée de paramètre μ, qui va nous servir par la suite. On a $F(x) = 1 - \mathrm{e}^{-\mu x}$, alors

$$F^{-1}(u) = \frac{-\ln(1-u)}{\mu}.$$

D'autres méthodes existent et sont utilisées lorsqu'on ne peut pas calculer l'inverse de la fonction de répartition d'une distribution de probabilités. Parmi celles-ci, on peut citer la méthode du rejet.

Simulation de chaînes de Markov

De nombreux systèmes à événements discrets peuvent être modélisés par des chaînes de Markov. Mais analyser ces chaînes n'est pas toujours aisé, notamment calculer de la distribution stationnaire quand l'espace des états est très grand. Ce qui nous intéresse donc maintenant c'est de simuler une variable aléatoire dont on ne connaît pas la distribution sans toutefois chercher à calculer cette distribution. Plusieurs techniques de simulation ont été développées.

Une première méthode consiste à d'abord modéliser la distribution que l'on cherche comme la distribution stationnaire d'une chaîne de Markov. Ensuite, on utilise le théorème ergodique des chaînes de Markov : soit x un état de la chaîne de Markov, l'état x_1 est généré selon les probabilités de transition à partir de x, x_2 est généré selon les probabilités de transition à partir de x_1 et ainsi de suite. La suite $(x_n)_{n \in \mathbb{N}}$ converge en loi vers la distribution stationnaire de la chaîne de Markov, sous hypothèse d'ergodicité de la chaîne, d'après le théorème ergodique.

Cette méthode est appelée la méthode de Monte-Carlo ([59]). Elle permet d'approcher la distribution stationnaire d'une chaîne de Markov en un temps fini. Mais cette méthode a quelques inconvénients : tout d'abord, l'état généré par l'algorithme n'est pas exactement distribué selon la distribution stationnaire de la chaîne de Markov. Il faut alors trouver combien de pas il faut effectuer pour obtenir une bonne approximation et trouver un compromis entre efficacité du calcul (faire un nombre raisonnable de pas de calculs) et précision du calcul.

Dans la section suivante et la suite du chapitre, nous décrivons une méthode permettant de simuler les chaînes de Markov de manière exacte et nous l'appliquons aux réseaux de Petri. C'est la méthode de *simulation parfaite*.

6.1.2 La simulation parfaite : l'algorithme de Propp et Wilson

La simulation parfaite a été introduite par Propp et Wilson ([80]). Cette méthode permet de générer une variable aléatoire selon la distribution stationnaire d'une chaîne de Markov à espace d'états finis de manière exacte, contrairement à la méthode précédente.

Représentation fonctionnelle des chaînes de Markov

Soit $\{X_n\}$ une chaîne de Markov définie sur un espace d'états fini $\mathcal{E}$.

On peut représenter une chaîne de Markov homogène par une équation de récurrence $X_{n+1} = \Phi(X_n, U_{n+1})$, où $(U_n)_{n\in\mathbb{N}}$ est une suite de variables aléatoires indépendantes et uniformément distribuées sur $[0, 1[$. L'idée principale de la simulation parfaite est de partir de tous les états possibles et de tirer des variables aléatoires U_n jusqu'à ce que toutes les trajectoires se rejoignent. C'est ce que l'on appelle le *couplage*.

Une première difficulté est de trouver une bonne représentation fonctionnelle de la chaîne. De plus, il faut trouver un moyen d'atteindre le couplage en un état distribué selon la distribution stationnaire. Les exemples ci-dessous sont repris de [59],qui constitue une bonne introduction aux chaînes de Markov à espace d'états fini, et notamment aux techniques de simulation.

Importance de la représentation fonctionnelle choisie Il se peut que l'on ne puisse jamais atteindre le couplage avec certaines représentation, comme le montre l'exemple ci-dessous.

Exemple 6.3 Soit X la chaîne de Markov représentée à la figure 6.1. Cette chaîne admet plusieurs représentations fonctionnelles, par exemple,

$$\Phi_1(s,x) = \begin{cases} 1 & \text{si } x \in [0, 0.5[\\ 2 & \text{si } x \in [0.5, 1[\end{cases} ,$$

ou

$$\Phi_2(1,x) = \begin{cases} 1 & \text{si } x \in [0, 0.5[\\ 2 & \text{si } x \in [0.5, 1[\end{cases} ,$$

et

$$\Phi_2(2,x) = \begin{cases} 2 & \text{si } x \in [0, 0.5[\\ 1 & \text{si } x \in [0.5, 1[\end{cases} .$$

Il est facile de voir qu'avec la représentation Φ_1, on couple en une étape avec probabilité 1, alors qu'avec la représentation Φ_2, on ne couple jamais.

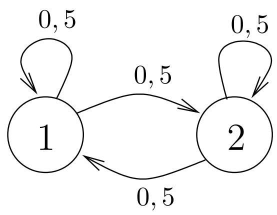


FIG. 6.1 – Exemple de chaîne de Markov

Le temps de couplage de la simulation parfaite dépend donc de la représentation choisie. L'exemple montre la nécessité de bien choisir sa représentation.

Comment faire le couplage ? Il y a plusieurs manières d'effectuer le couplage. La méthode la plus naturelle semble être celle qui consiste à partir de tous les états et générer les trajectoires à partir du temps 0 en tirant à chaque étape une variable aléatoire qui permet de calculer l'état suivant pour chaque trajectoire jusqu'à ce que toutes les trajectoires se rejoignent. C'est ce qu'on appelle le *couplage avant*. Malheureusement, l'exemple 6.4 montre que cette manière de faire ne permet pas de générer une variable aléatoire selon la bonne distribution.

Exemple 6.4 Considérons la chaîne de Markov représentée sur la figure 6.2. Sa probabilité stationnaire est $[2/3, 1/3]$. Considérons maintenant le premier instant de couplage. L'état dans lequel il y a eu couplage est nécessairement 1 : si on couplait pour la première fois dans l'état 2, alors l'état précédent ne peut pas être 2. Alors il y avait couplage au pas d'avant. Appliquer la méthode de couplage avant génère 1 avec probabilité 1.

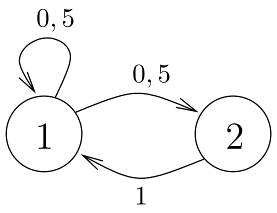


FIG. 6.2 – Exemple de chaîne de Markov

La seconde astuce de l'algorithme de Propp et Wilson consiste alors à faire un *couplage par l'arrière*. La méthode de simulation par l'arrière n'est pas nouvelle. Elle était déjà connue sous le nom de *schéma de Loynes*, qui est fréquemment utilisée pour montrer des résultats d'existence d'états stationnaires de systèmes à événements discrets, voir par exemple [67, 46]. La principale innovation de l'algorithme de Propp et Wilson est donc d'utiliser ce schéma pour le calcul effectif de ces états stationnaires. L'exemple suivant illustre le principe de la simulation par couplage arrière.

Exemple 6.5 Reprenons la chaîne de Markov de la figure 6.2 et prenons comme représentation fonctionnelle

$$\Phi(1, x) = \begin{cases} 1 & \text{si } x \in [0, 0.5[\\ 2 & \text{si } x \in [0.5, 1[\end{cases} \qquad \text{et} \qquad \Phi(2, x) = 1.$$

Le couplage par l'arrière s'effectue depuis des temps négatifs vers le temps 0. Pour la première étape, on tire U_0 une variable aléatoire uniformément distribuée sur $[0, 1[$, et on calcule les états au temps 0, $X_0(s)$ en partant du temps -1, $X_{-1}(s) = s$, avec $X_0(s) = \Phi(s, U_0)$. Si $X_0(2) = X_0(1)$, alors il y a couplage. Sinon, on génère U_{-1} une nouvelle variable aléatoire, et partant de tous les états au temps -2, on calcule les trajectoires au temps 0 : $X_0(s) = \Phi(\Phi(s, U_{-1}), U_0)$. On continue ce procédé jusqu'à ce que toutes les trajectoires terminent dans le même état au temps 0. Par exemple, si $U_0 = 0,7$ et $U_{-1} = 0,3$, le couplage a lieu en deux étapes et l'état obtenu est 2. Ceci n'était pas possible avec le couplage par l'avant.

Les théorèmes suivants prouvent l'algorithme 6.1 sous réserve de terminaison et donnent une condition suffisante sur la représentation fonctionnelle de la chaîne de Markov pour prouver que l'algorithme termine presque sûrement.

Théorème 6.6 *Soit P une matrice de transition d'une chaîne de Markov irréductible et apériodique d'espace d'états $S = \{s_1, \ldots, s_n\}$ et de probabilité stationnaire $\pi = (\pi_1, \ldots, \pi_n)$. Soit*

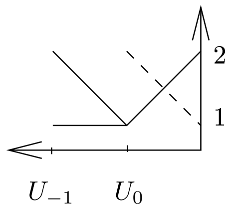


FIG. 6.3 – Couplage par l'arrière de l'exemple 6.5.

Φ une représentation de P. Supposons que l'algorithme 6.1 termine avec une probabilité 1 et retourne Y. Alors pour tout $i \in \mathbb{N}_n$, $\mathbb{P}\{Y = s_i\} = \pi_i$.

L'idée de la preuve est que s'il y a couplage arrière, comme la distribution itérée partant d'une distribution quelconque tends vers la distribution stationnaire, on aurait obtenu le même état en partant du temps $-\infty$.

Théorème 6.7 *Supposons que pour tout $(x, y) \in \mathcal{E}^2$, il existe $n \in \mathbb{N}$ tel que*

$$\mathbb{P}[\Phi(\Phi(\cdots(\Phi(x, U_1), \ldots, U_{n-1}), U_n) = \Phi(\Phi(\cdots(\Phi(y, U_1), \ldots, U_{n-1}), U_n)] > 0 \tag{6.1}$$

alors l'algorithme 6.1 termine en un temps presque sûrement fini.

Algorithme 6.1 : Algorithme de Propp et Wilson

Données : Φ une représentation fonctionnelle d'une chaîne de Markov finie ergodique : $X_{n+1} = \Phi(X_n, U_{n+1})$, une suite strictement croissante d'entiers $N_1, N_2, \ldots$ et une suite $U_0, U_{-1}, U_{-2}, \ldots$ de v.a. i.i.d. uniformément distribuées sur $[0, 1[$.
Résultat : État de la chaîne distribué selon la probabilité stationnaire de la chaîne.
début
  $m \leftarrow 1$;
  répéter
    pour chaque *état s* **faire**
      pour chaque *i de 1 à N_m* **faire**
        $X_{-N_m+i}(s) = \Phi(X_{-N_m+i-1}(s), U_{-N_m+i})$;
      finprch
    finprch
    $m \leftarrow m + 1$;
  jusqu'à $X_0(s) = X_0(t)$, *pour tous t, s états de la chaîne*;
  retourner X;
fin

Ce dernier théorème est très utile pour montrer qu'avec une certaine représentation, l'algorithme termine.

Remarque 6.8 Il faut garder en mémoire la suite des U_{-i} : sinon, la distribution n'est pas la distribution stationnaire.

Remarque 6.9 Une suite presque optimale N_m à choisir est 2^m. En effet, si on prend $N_m = i$, on a un nombre quadratique de pas à effectuer en fonction du temps de couplage, puisqu'il faut recalculer à chaque fois les trajectoires de N_m à 0. Avec 2^m, ce nombre de calculs n'est que linéaire. Une étude plus précise a été effectuée dans [80], section 5.2.

6.1.3 Techniques de réduction des états

Pour des chaînes dont l'espace d'états est gros, les temps de calculs sont aussi très grands. On cherche alors à diminuer le nombre des états de départ : existe-t-il un sous-ensemble $F \subset \mathcal{E}$ tel que si à la date N il y a couplage des trajectoires issues de F, alors il y a couplage de toutes les trajectoires ?

Par exemple, si l'espace d'états est $\{0, \ldots, k\}$, et s'il existe une représentation Φ telle que si $0 \leq i \leq j \leq k$, $\Phi(i, x) \leq \Phi(j, x)$ pour tout $x \in [0, 1[$, alors si les trajectoires issues de 0 et de k se rejoignent, alors il y aura eu couplage. Cette technique a été utilisée par de nombreux auteurs, par exemple [92] pour la simulation parfaite de fonctionnelles de chaînes de Markov, et en particulier pour des modèles de partage de ressources.

6.2 Simulation parfaite des graphes d'événements

6.2.1 Principe de simulation des réseaux à choix libres

Soit $(\mathcal{N}, M, \tau)$ un réseau à choix libres vivant borné temporisé, où les temporisations $(\tau_t)_{t\in\mathcal{T}}$ sont des suites de variables aléatoires i.i.d. exponentiellement distribuées de paramètre μ_t. Le réseau obéit à la politique de la compétition. On peut alors modéliser son évolution par une chaîne de Markov à temps continu sur l'espace des marquages atteignables. Le générateur infinitésimal est $Q = (Q_{M_1,M_2})_{M_1,M_2\in\mathcal{R}(M)}$ avec :

$$Q_{M_1,M_2} = \begin{cases} k.\mu & \text{si on peut passer de } M_1 \text{ à } M_2 \text{ en} \\ & \text{ franchissant une transition de paramètre } \mu \\ & \text{ et de degré d'habilitation } k, \\ 0 & \text{si l'on ne peut pas passer de } M_1 \text{ à } M_2 \text{ en} \\ & \text{ franchissant une transition,} \\ -\sum_{M'\neq M_1} Q_{M_1,M'} & \text{si } M_1 = M_2. \end{cases}$$

Pour effectuer la simulation parfaite, on transforme cette chaîne à temps continu en une chaîne à temps discret par uniformisation (cette transformation préserve la probabilité stationnaire de la chaîne, voir section 1.2.2). On prend comme coefficient d'uniformisation $q = \sum_{t\in\mathcal{T}} k_t\mu_t$, où k_t est le degré d'habilitation maximal de la transition t.

On choisit la représentation de la chaîne de Markov Φ comme suit :

$$\text{si } u \in \left[\frac{\sum_{j=1}^{i-1} k_j\mu_j + (k-1)\mu_i}{q}, \frac{\sum_{j=1}^{i-1} k_j\mu_j + k\mu_i}{q}\right[, \text{ alors}$$

$$\Phi(M, u) = \begin{cases} M' & \text{si } t_i \text{ est habilitée avec degré supérieur ou égal à } k \text{ et } M \xrightarrow{t_i} M' \\ M & \text{sinon} \end{cases}$$

Cette représentation paraît clairement la plus adaptée au problème, car le tir d'une variable aléatoire uniformément distribuée sur $[0, 1[$ correspond au tir potentiel d'une transition ayant un degré d'habilitation minimum, qui ne dépend pas du marquage.

Proposition 6.10 *Avec cette représentation, l'algorithme de Propp et Wilson termine en un temps presque sûrement fini.*

Démonstration. Pour montrer ce résultat, on applique le théorème 6.7, et il suffit alors de montrer que pour tout couple d'états de la chaîne de Markov considérée, il existe un entier N tel que les trajectoires issues de ces deux états se rejoignent en un temps au plus N avec une probabilité strictement positive.

Soient M_1 et M_2 deux marquages atteignables et a une transition du réseau. Il existe une suite de tirs qui a une probabilité strictement positive d'être tirée qui mène de M_1 à M_a, le marquage de blocage de la transition a. Cette suite de tirs mène M_2 en un marquage M_3. Il existe alors une suite de tirs qui mène de M_3 à M_a sans tirer la transition a (théorème 3.29). Donc la trajectoire issue de M_1 reste en M_a tandis que celle issue de M_2 arrive aussi en M_a, ce qui achève la preuve. $\square$

Mais le nombre de marquages atteignables peut être exponentiel en la taille du réseau de Petri. On cherche donc maintenant à diminuer le nombre d'états à considérer pour obtenir le couplage.

6.2.2 Réduction du nombre d'états initiaux pour la simulation parfaite des graphes d'événements

Le réseau $\mathcal{N}$ est maintenant un graphe d'événements temporisé, dont les temporisations pour chaque transition sont i.i.d. exponentiellement distribuées et dont le degré d'habilitation de chaque transition est au plus 1 ($\min_{p\in{}^\bullet t} M(p) \leq 1$ pour tout marquage atteignable M). On va montrer qu'on peut réduire le nombre d'états pour la simulation parfaite : il suffit de prendre le marquage de blocage de chaque transition (il y en a un par transition).

Avec la modélisation décrite ci-dessus, comme représentation de la chaîne de Markov uniformisée, avec la constante $q = \sum_{i=1}^{|\mathcal{T}|} \mu_i$, la représentation fonctionnelle devient alors $X_{n+1} = \Phi(X_n, U_{n+1})$ avec :

$$\Phi(M,u) = \begin{cases} M' & \text{si } u \in [(\sum_{i=1}^{k-1}\mu_i)/q, (\sum_{i=1}^{k}\mu_i)/q[\quad \text{et } M \xrightarrow{t_i} M', \\ M & \text{si } u \in [(\sum_{i=1}^{k-1}\mu_i)/q, (\sum_{i=1}^{k}\mu_i)/q[\quad \text{et } t_i \text{ non habilitée par } M. \end{cases}$$

Ainsi, quelle que soit la valeur de u, pour chaque marquage, on essaie de franchir la même transition. Si c'est possible, alors le franchissement a lieu, sinon, rien ne se passe.

On veut maintenant montrer le théorème suivant :

Théorème 6.11 *Soit $\mathcal{G}$ un graphe d'événements vivant borné temporisé pour lequel toutes les transitions ont un degré d'habilitation 1 avec des temporisations i.i.d. exponentiellement distribuées. L'algorithme de simulation parfaite en partant des marquages de blocage termine avec probabilité 1 et sa sortie est distribuée selon la distribution stationnaire du processus des marquages.*

Montrons tout d'abord quelques lemmes préliminaires.

Soit $M_{a,b}$ un marquage pour lequel les seules transitions habilitées sont a et b. En partant de $M_{a,b}$, on fait évoluer le réseau en bloquant la transition a jusqu'à obtenir le marquage de blocage de a, M_a. La suite de tirs pour atteindre M_a est notée σ_a. On fait de même pour la transition b, et on définit ainsi M_b et σ_b. Noter que les marquages de blocage sont obtenus sans franchir la transition bloquée et donc les vecteurs de Parikh $\vec{\sigma_a}$ et $\vec{\sigma_b}$ sont uniquement définis ([52], lemme 4.4).

Rappelons qu'à chaque instant, on essaie de tirer une transition. Si c'est possible, on fait évoluer le marquage, sinon, on ne fait rien. Pour une suite aléatoire de transitions σ, on note $\#\text{franch}_\sigma(t, M)$ le nombre de franchissement de la transition t à partir du marquage M en franchissant les transitions de σ lorsque c'est possible.

Lemme 6.12 *Pour une suite de transitions σ, on a*

$$\#\mathit{franch}_\sigma(t, M_{a,b}) = \min\left(\#\mathit{franch}_\sigma(t, M_a) + |\vec{\sigma_a}|_t, \#\mathit{franch}_\sigma(t, M_b) + |\vec{\sigma_b}|_t\right). \tag{6.2}$$

Démonstration. On procède par récurrence sur la longueur de σ.

Initialisation : $|\sigma| = 0$. Alors, $\#\mathrm{franch}_\sigma(t, M_{a,b}) = \#\mathrm{franch}_\sigma(t, M_a) = \#\mathrm{franch}_\sigma(t, M_b) = 0$, et il suffit de montrer que $|\vec{\sigma_a}|_t = 0$ ou que $|\vec{\sigma_b}|_t = 0$. Pour ce faire, on montre que σ_a et σ_b ont des supports disjoints.

Supposons qu'ils aient une transition en commun. Considérons la première transition t rencontrée par la suite σ_a qui apparaît aussi dans σ_b. Cette transition n'est ni a ni b par construction de σ_a et σ_b. Dans chaque place initialement vide en entrée de t, un jeton au moins a été ajouté. Mais alors, comme chaque place n'a qu'une transition en entrée (c'est un graphe d'événements), la transition en amont d'une place qui a été remplie a aussi été traversée par les deux suites σ_a et σ_b, ce qui est en contradiction avec le fait qu'on considérait la première transition traversée en commun. Les suites σ_a et σ_b ont donc bien des supports disjoints. On note aussi que σ_a commence nécessairement par b, et σ_b par a.

Hypothèse de récurrence : On suppose le lemme vrai pour les suites de tirs de longueur n.

Récurrence : Si $|\sigma| = n + 1$, on note $\sigma = \sigma' t$, où $t \in \mathcal{T}$ et $|\sigma'| = n$. Il suffit de montrer l'égalité 6.2 pour la transition t. En fait,

$$\#\mathrm{franch}_\sigma(t, M_a) + |\vec{\sigma_a}|_t = \#\mathrm{franch}_{\sigma_a.\sigma}(t, M_{a,b}).$$

Pour un marquage initial M quelconque, on peut franchir t pour la k-ième fois ou la transition k a été franchie au moins k fois après le tir d'une suite σ' si pour place en entrée de t a été marquée par au moins k jetons pendant l'évolution correspondant au tir de σ', c'est-à-dire, si $\forall p \in {}^\bullet t$, on a $\#\mathrm{franch}_{\sigma'}({}^\bullet p, M) + M(p) \geq k$ (comme dans un graphe d'événements une place a une unique transition en entrée, par abus de notation, on note ${}^\bullet p$ cette transition). Mais alors, si

$$\min_{p \in {}^\bullet t}(\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_{a,b}) + M_{a,b}(p)) > \#\mathrm{franch}_{\sigma'}(t, M_{a,b}), \tag{6.3}$$

la transition t aurait pu être franchie strictement plus de fois qu'elle ne l'a été. Cette transition est donc habilitée et est franchie. On a donc

$$\#\mathrm{franch}_\sigma(t, M_{a,b}) = \#\mathrm{franch}_{\sigma'}(t, M_{a,b}) + 1$$

et

$$\#\mathrm{franch}_\sigma(t, M_{a,b}) = \#\mathrm{franch}_{\sigma'}(t, M_{a,b}) \qquad \text{sinon.}$$

En utilisant l'hypothèse de récurrence,

$$\begin{aligned}\min_{p \in {}^\bullet t}(\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_{a,b}) + M_{a,b}(p)) = \min_{p \in {}^\bullet t} \min(&\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_a) + |\vec{\sigma_a}|_{{}^\bullet p} + M_{a,b}(p), \\ &\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_b) + |\vec{\sigma_b}|_{{}^\bullet p} + M_{a,b}(p)).\end{aligned}$$

Or, puisque $M_{a,b} \xrightarrow{\sigma_a} M_a$, $M_a(p) = M_{a,b}(p) + |\sigma_a|_{{}^\bullet p} - |\sigma_a|_{p^\bullet}$ et puisque $p^\bullet = t$, on a

$$|\sigma_a|_{{}^\bullet p} + M_{a,b}(p) = M_a(p) + |\sigma_a|_t.$$

La même égalité s'obtient en remplaçant a par b. En injectant cette égalité dans la précédente, on obtient

$$\min_{p\in{}^\bullet t}(\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_{a,b}) + M_{a,b}(p)) =$$
$$\min_{p\in{}^\bullet t}\min \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_a) + |\vec{\sigma_a}|_t + M_a(p), \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_b) + |\vec{\sigma_b}|_t + M_b(p)).$$

et

$$\min_{p\in{}^\bullet t}(\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_{a,b}) + M_{a,b}(p)) =$$
$$\min(|\vec{\sigma_a}|_t + \min_{p\in{}^\bullet t} \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_a) + M_a(p), |\vec{\sigma_b}|_t + \min_{p\in{}^\bullet t} \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_b) + M_b(p)).$$

Supposons que

$$\min_{p\in{}^\bullet t}(\#\mathrm{franch}_{\sigma'}({}^\bullet p, M_{a,b}) + M_{a,b}(p)) < |\vec{\sigma_b}|_t + \min_{p\in{}^\bullet t} \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_b) + M_b(p).$$

Alors $\#\mathrm{franch}_\sigma(t, M_{a,b}) = \#\mathrm{franch}_{\sigma'}(t, M_{a,b}) + 1$ augmente si et seulement si $\#\mathrm{franch}_\sigma(t, M_a) = \#\mathrm{franch}_{\sigma'}(t, M_a) + 1$. Si

$$|\vec{\sigma_a}|_t + \min_{p\in{}^\bullet t} \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_a) + M_a(p) = |\vec{\sigma_b}|_t + \min_{p\in{}^\bullet t} \#\mathrm{franch}_{\sigma'}({}^\bullet p, M_b) + M_b(p),$$

alors $\#\mathrm{franch}_\sigma(t, M_{a,b}) = \#\mathrm{franch}_{\sigma'}(t, M_{a,b}) + 1$ si et seulement si $\#\mathrm{franch}_\sigma(t, M_a) = \#\mathrm{franch}_{\sigma'}(t, M_a) + 1$ et $\#\mathrm{franch}_\sigma(t, M_b) = \#\mathrm{franch}_{\sigma'}(t, M_b) + 1$.

□

Remarquons que si la transition t peut être franchie pour la trajectoire issue de $M_{a,b}$, alors elle peut aussi l'être pour la trajectoire issue de M_a ou de M_b. Si elle ne peut pas être franchie, alors, elle ne peut pas l'être non plus pour les trajectoires issues de M_a et de M_b. Donc si les trajectoires issues des marquages M_a et M_b se rejoignent, la trajectoire issue de $M_{a,b}$ aura aussi rejoint les deux autres. En effet, par définition du couplage, exactement les mêmes transitions sont franchies pour les trajectoires issues des marquages M_a et M_b après couplage. D'après le lemme, après couplage, les mêmes transitions sont aussi franchies pour la trajectoire issue de $M_{a,b}$. Mais alors, si on franchit les transitions de sorte à obtenir un marquage de blocage pour les trajectoires issues de M_a et de M_b, c'est aussi le marquage atteint pour l'autre trajectoire : exactement les mêmes transitions sont franchissables, une seule, et ce marquage de blocage est unique. Ce marquage a été atteint en franchissant exactement les mêmes transitions. C'est donc que dès le couplage, le marquage était le même pour toutes les trajectoires.

On généralise maintenant le lemme précédent pour des ensembles de transitions :

À l'instant initial, les transitions habilitées forment un ensemble T. Soit T_1, T_2 une partition non triviale de T. Comme précédemment, on fait évoluer le marquage en bloquant les transitions de T_1 (resp. T_2) jusqu'à obtenir un blocage et le marquage M_1 (resp. M_2). La suite des transitions franchies est σ_1 (resp. σ_2). On montre comme précédemment que σ_1 et σ_2 ont des supports disjoints et le lemme :

Lemme 6.13 *Pour une suite de transitions σ, on a :*

$$\#\mathit{franch}_\sigma(t, M) = \min\left(\#\mathit{franch}_\sigma(t, M_1) + |\vec{\sigma_1}|_t, \#\mathit{franch}_\sigma(t, M_2) + |\vec{\sigma_2}|_t\right).$$

Démonstration. Idem lemme 6.12 □

On peut maintenant prouver le théorème 6.11.

Démonstration. (du théorème 6.11) On procède de nouveau par récurrence, sur le nombre de transitions habilitées pour un marquage atteignable. Pour un marquage atteignable M. On pose T l'ensemble des transitions habilitées pour M.

Initialisation : $|T| = 1$. Le marquage initial est un marquage de blocage, par définition et unicité du marquage de blocage.

Hypothèse de récurrence : On suppose le lemme vrai pour les marquages habilitant au plus k transitions.

Récurrence : Soit a une transition de T. On pose $T_1 = \{a\}$ et $T_2 = T \setminus \{a\}$. Comme dans le lemme 6.13, on note M_1 le marquage atteint pour lequel seule a est habilitée, et M_2 le marquage atteint pour lequel seules les transitions de T_2 sont habilitées obtenu en bloquant la transition a. On applique l'hypothèse de récurrence à M_2. Si les trajectoires des marquages de blocage se rejoignent, alors la trajectoire issue de M_2 les a aussi rejoint. On applique le lemme 6.13 : si les marquages M_2 et $M_1 = M_a$ se rejoignent, alors celle issue de M les a aussi rejoint. Donc si les trajectoires issues des marquages de blocage se rejoignent, la trajectoire issue de M les aura aussi rejointes, ce qui achève la preuve.

□

Calcul du débit

En itérant un grand nombre de fois la simulation, on peut obtenir une bonne approximation de la probabilité stationnaire π. Ainsi, par exemple, on peut calculer l'activité d'un graphe d'événements (nombre moyen de transitions franchies par unité de temps) :

$$D = \Big(\sum_{M' \in \mathcal{R}(M)} \frac{\pi_{M'}}{\mu_{M'}} \Big)^{-1},$$

où $\mu_{M'}$ est la somme des paramètres des lois exponentielles des temporisations des transitions habilitées (comptées autant de fois que leur degré d'habilitations).

6.2.3 Contre-exemples pour la réduction des états dans les cas plus généraux de réseaux de Petri

Dans cette section, nous montrons que les marquages de blocage ne sont plus des états extrémaux pour la simulation parfaite dans des cas plus généraux.

Cas où le degré d'habilitation est supérieur à 1

Les marquages de blocages ne suffisent pas pour la simulation parfaite, comme le montre l'exemple suivant.

Exemple 6.14 On considère le réseau de la figure 6.4, qui est un circuit de trois places et trois transitions.

On effectue les tirs suivants (les nombres entre parenthèse indiquent le degré d'habilitation minimum que doit avoir la transition pour tirer) : $3(1), 3(1), 2(1), 1(2), 3(1)$. À la suite de cette suite de tirs, partant des marquages de blocage, on obtient le marquage $(1, 1, 0)$, et partant du marquage initial, on obtient $(2, 0, 0)$.

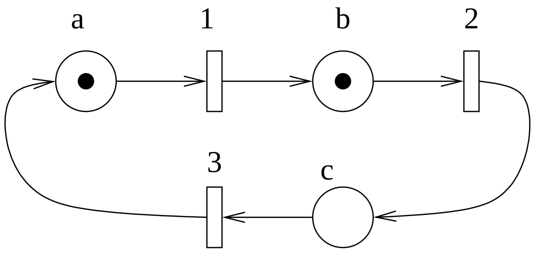


FIG. 6.4 – Exemple de réseau avec degré d'habilitation supérieur à 1.

Cas des machines d'états

On suppose maintenant que les degrés d'habilitation des transitions sont toujours 1, quitte à rajouter des boucles sur chaque transition (elles ne sont pas représentées dans la figure 6.5). Nous allons montrer que même dans ce cas-là, on ne peut pas effectuer la simulation parfaite en ne partant que des marquages de blocage (le marquage de blocage est, pour chaque transition t, le marquage où la seule place marquée est ${}^{\bullet}t$, qui contient n jetons, le nombre initial de jetons dans le réseau). L'exemple suivant le montre.

Exemple 6.15 On considère le réseau de la figure 6.5. On effectue les tirs suivants : $1, 1, 2, 2, 7, 5,$

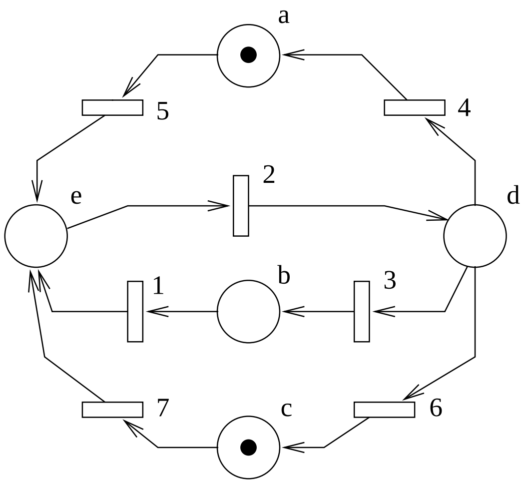


FIG. 6.5 – Exemple de machine d'états.

$2, 4, 4, 2, 3, 7, 2, 4$. À la suite de cette suite de tirs, partant des marquages de blocage, on obtient les deux jetons dans la place a, alors que partant du marquage initial, on obtient un jeton en place a et un jeton en place b. La figure 6.6 montre les trajectoires issues des marquages de blocage ainsi que celle issue du marquage initial.

Cas des réseaux à choix libres 1-bornés

L'exemple ci-dessus n'est valable que pour les machines d'états contenant plus de 1 jeton : dans le cas d'une machine d'états 1 bornée, les marquages atteignables sont exactement les marquages de blocages. L'exemple suivant montre que la réduction du nombre d'états en ne prenant que les marquages de blocage n'est pas valide.

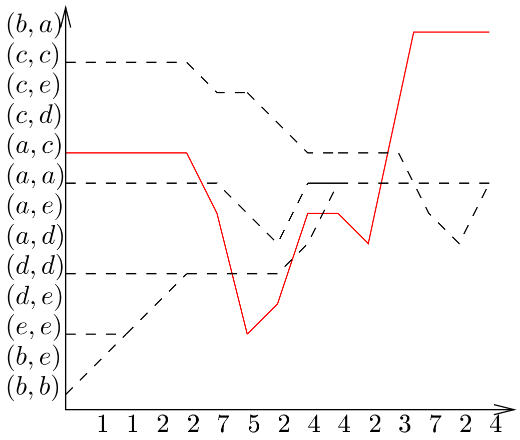


FIG. 6.6 – Trajectoires pour la figure 6.15.

Exemple 6.16 On considère le réseau de la figure 6.7.

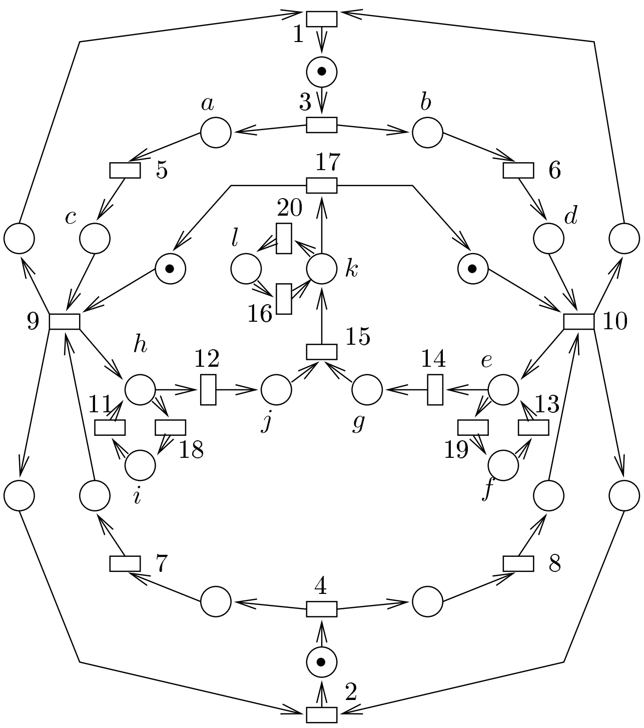


FIG. 6.7 – Exemple de réseau à choix libres 1-borné.

On commence par tenter de franchir toutes les transitions possibles à l'exception des transitions 1, 2, 3 et 4. Comme les seules transitions franchissables à partir du marquage représenté sur la figure 6.7 sont les transition 3 et 4, ce marquage reste inchangé. Regardons comment ont évolué les marquages issus des marquages de blocage. Les seuls qui n'ont pas changé sont les marquages de blocage des transitions 1, 2, 3 et 4, par construction. En essayant de franchir dans l'ordre les transitions 5, 6, 7, 8, 9, 10, aucune de ces transitions n'est plus habilitée, et ne le sera plus tant qu'on s'interdit de franchir les transitions 1, 2, 3 et 4. Si l'on tire maintenant la suite de transitions 11, 12, 13, 14, 15, 16, 17 aucune de ces transitions n'est plus habilitée. Dans les marquages autres que le marquage initial et les marquages de blocage des transitions 3 et 4, les seules transitions habilitées sont parmi les transitions 16, 1 et 2. De plus, les places de a à l ne sont pas marquées.

Dans un deuxième temps, on fait évoluer le marquage initial en franchissant les transitions 3 et 4. On ne franchit toujours pas les transitions 1 et 2, donc les marquages de blocage de ces transitions n'évoluent pas. Grâce aux transitions franchies précédemment, les autres trajectoires ne vont pas évoluer non plus. Essayons ensuite de franchir la suite 5, 8, 9, 10, 18, 19. La trajectoire issue du marquage initial tire effectivement les transitions 3, 4, 5, 8. La trajectoire issue du marquage de blocage de la transition 3 tire effectivement les transitions 3, 5, 9, 18 et celle issue du marquage de blocage de la transition 4 tire 4, 8, 10, 19. Pour ces deux dernières trajectoires, la place i ou f est marquée. Tant que les transitions 11 et 13 ne sont pas tirées, la transition 15 ne pourra jamais être franchie pour ces transitions. Maintenant, tirons les transitions 6, 7, 9, 10, 12, 14. La trajectoire issue du marquage initial tire effectivement ces transitions. La transition 15 est donc habilitée. C'est la seule trajectoire qui habilite cette transition après toutes les suites de tirs, et les places k et l sont vides. Tirons maintenant les transitions 15 et 20. La place l est marquée dans le marquage de la trajectoire issue du marquage initial, et cette place n'est marquée dans aucun des autres marquages atteints. Or, si l'on enlève au réseau la place l et les transitions 16 et 20, le réseau est toujours vivant 1-borné. On peut donc faire évoluer le marquage des autres trajectoires vers un même marquage sans franchir les transitions 16 et 20. Donc ce marquage obtenu pour les trajectoires issues des marquages de blocage ne sera pas atteint par la trajectoire issue du marquage initial.

La méthode utilisée ne permet donc de faire de la simulation parfaite avec une diminution du nombre d'états que pour des graphes d'événements avec degré d'habilitation des transitions 1. Nous allons maintenant voir que l'on peut aussi faire de la simulation parfaite pour certains systèmes (max,plus), et grâce à la modélisation par tas de pièces, présenter une méthode alternative de la simulation des réseaux à choix libres vivants 1-bornés.

6.3 Simulation parfaite avec l'algèbre (max,plus)

On peut modéliser la dynamique temporelle de certaines classes de réseaux de Petri avec l'algèbre (max,plus). On peut distinguer d'une part les graphes d'événements, qu'on peut représenter par des matrices (max,plus) (leur évolution s'obtient par multiplications successives de matrices), et les réseaux à choix libres 1-bornés, qu'on peut représenter par des tas de pièces et des automates (max,plus).

Nous définissons tout d'abord quelques notions supplémentaires sur les matrices (max,plus) et les tas de pièces.

Définition 6.17 (profil d'un vecteur) *Soit $u \in \mathbb{R}^k$. On appelle profil de u le vecteur noté Prof(u) défini par Prof$(u)_i = u_i - \min_{j \in \mathbb{N}_k} u_j$, $u \in \mathbb{R}^k$.*

Si deux vecteurs diffèrent d'une constante additive, alors ils ont le même profil. On définit de la même manière le *profil d'un tas de pièces* : c'est le profil du vecteur de la hauteur du tas en chacune de ses colonnes.

Définition 6.18 (matrice de rang 1) *Une matrice $A \in \mathbb{R}_{\max}^{k \times k}$ est de rang 1 si*

1. *$\forall u \in \mathbb{R}^k$, $A \otimes u \in \mathbb{R}^k$ (i.e. aucune ligne n'a toutes ses coordonnées égales à $-\infty$) ;*
2. *$\forall u \in \mathbb{R}^k$, $u \otimes A \in \mathbb{R}^{1 \times k}$ (i.e. aucune colonne n'a toutes ses coordonnées égales à $-\infty$) ;*

3. $\forall u, v \in \mathbb{R}^k$, *Prof*$(A \otimes u) =$ *Prof*$(A \otimes v)$.

Lemme 6.19 *On a les équivalences :*

$$A \text{ est de rang 1 } \Leftrightarrow \forall i, j,\ \mathit{Prof}(A_{.i}) = \mathit{Prof}(A_{.j}) \Leftrightarrow \forall i, j,\ \mathit{Prof}(A_{i.}) = \mathit{Prof}(A_{j.}).$$

La simulation parfaite par les matrices (max,plus) se fait sur l'espace d'états des profils, et une étape de la simulation consiste à multiplier un profil par une matrice. Le lemme précédent montre que si une matrice est de rang 1, alors, quel que soit le vecteur à coefficients réels par lequel on la multiplie, le profil du résultat est le même. Il y a donc couplage. Il reste donc maintenant à montrer que l'on obtient presque sûrement en un temps fini une matrice de rang 1. Nous montrons ce résultat dans deux cas : pour des matrices modélisant des graphes d'événements et des matrices modélisant des tas de pièces.

6.3.1 Une méthode alternative pour la simulation des graphes d'événements.

Soient (Hyp) les hypothèses suivantes sur une suite de matrices $(A(n))_{n\in\mathbb{N}}$:

$$(Hyp) \qquad \begin{cases} \forall i, j,\ (A(n)_{ij})_{n\in\mathbb{N}} \text{ est i.i.d.,} \\ \forall i, j,\ \mathbb{P}(A(1)_{ij} = -\infty) \in \{0, 1\}, \\ \text{le support de } A_{11} \text{ est } \mathbb{R}_+, \\ \text{le graphe sous-jacent à } A(1) \text{ est fortement connexe.} \end{cases}$$

On peut représenter l'évolution des graphes d'événements par une équation de récurrence d'ordre 1 (max,plus) (voir chapitre 3) :

$$X(n) = A(n) \otimes X(n-1),$$

où $(A(n))$ est une suite de matrices satisfaisant (Hyp).

L'algorithme 6.2 décrit la simulation parfaite de graphes d'événements par l'algèbre (max,plus).

Algorithme 6.2 : Simulation parfaite pour des matrices (max,plus)

Données : Une suite de matrices (max,plus), $(A(n))_n$ vérifiant (Hyp) et une suite strictement croissante d'entiers positifs $N_1, N_2, \ldots$

début
- $m \leftarrow 1$;
- **répéter**
  - $B_m \leftarrow A(-m) \otimes B_{m-1}$;
  - $m \leftarrow m + 1$;
  - **retourner** *Prof*$((B_m)_{.1})$;
- **jusqu'à** B_m *est de rang 1* ;

fin

Sous les hypothèses (Hyp), il est prouvé en [69] que le système $X(n) = X(n-1) \otimes A(n)$ admet un régime stationnaire, c'est-à-dire que $\mathrm{Prof}(X(n))$ converge en loi vers un profil stationnaire Prof_∞ indépendant des conditions initiales, $X(0)$.

Théorème 6.20 *Si l'algorithme de simulation parfaite par (max,plus) termine, alors sa sortie est distribuée selon la distribution stationnaire du profil du système (max,plus).*

Démonstration. En utilisant le lemme 6.19, si la matrice est de rang un, et si X_∞ est un état ayant un profil stationnaire, alors $X \otimes B_m = X_\infty \otimes B_m$ pour tout X. Puisque $\mathrm{Prof}(X_\infty)$ est stationnaire, alors $\mathrm{Prof}(X_\infty \otimes B_m) = \mathrm{Prof}(X \otimes B_m)$ l'est aussi. □

Le reste de cette section est consacré à la preuve, sous les hypothèses (Hyp), que l'algorithme 6.2 termine avec probabilité 1. Le lemme et le théorème suivants sont une spécialisation du théorème 6.15 de [69]. Nous en redonnons ici une preuve pour le cas qui nous intéresse.

Lemme 6.21 *Soit $(A_n)_{n\in\mathbb{N}}$ une suite de matrices de $\mathbb{R}_{\max}^{k\times k}$. Le produit $A(1) \otimes \cdots \otimes A(n)$ est de rang 1 avec probabilité strictement positive dès que $n > k$.*

Démonstration.

Pour tout n, l'événement $E_{n,\varepsilon}$ est défini par

$$E_{n,\varepsilon} = \{\omega \mid A_{i,j}(m) \in [h_{ij} - \varepsilon, h_{ij} + \varepsilon] \ \forall m \leq n\},$$

où h_{ij} est dans le support de $A_{i,j}(n)$ pour tout $n \in \mathbb{N}$ et l'ensemble $\{h_{ij} \mid i \leq k, j \leq k\}$ ne satisfait aucune équation linéaire avec des coefficients dans $\{1, -1\}$ impliquant h_{ij} dès que le support de $A_{i,j}(n)$ est continu. De plus, $h_{11} > k(\max_{(i,j)\neq(1,1)} h_{ij} + \varepsilon)$.

Par construction de la matrice déterministe $H = (h_{ij})$ et en utilisant la théorie des matrices (max,plus) déterministes (voir [11, 18] par exemple), pour tout i, il existe une suite $i_1 \dots i_\ell$ et pour tout j, il existe une suite $j_1 \dots j_r$ telle que pour tout $n > k$,

$$H_{ij}^n = h_{ii_1} + \cdots + h_{i_\ell 1} + (n - \ell - r - 2)h_{11} + h_{1j_1} + \cdots + h_{j_r j}.$$

D'après la forme de H^n, il est facile de voir que H^n est de rang 1. En effet, la différence $H_{ij}^n - H_{i'j}^n$ ne dépend pas de j. De plus, d'après les hypothèses, il y a un unique chemin simple de poids maximal d'un nœud à un autre. Soit η le minimum, pour toute paire de nœuds, de la différence entre le poids maximal d'un chemin entre ces deux nœuds et le poids du deuxième chemin le plus lourd entre ces deux nœuds.

Maintenant, si $n > k$ et $\varepsilon < \eta$, le produit $A(1) \otimes \cdots \otimes A(n)$ est aussi de rang 1 sur $E_{n,\varepsilon}$ pour la même raison :

$$\begin{aligned}(A(1) \otimes \cdots \otimes A(n))_{ij} = {} & A(1)_{ii_1} + \cdots A(\ell+1)_{i_\ell 1} \\ & + A(\ell+2)_{11} + \cdots + A(n-r-1)_{11} \\ & + A(n-r)_{1j_1} + \cdots + A(n)_{j_r j},\end{aligned}$$

de sorte que $(A(1) \otimes \cdots \otimes A(n))_{ij} - (A(1) \otimes \cdots \otimes A(n))_{i'j}$ ne dépend pas de j.

Pour finir la preuve, il suffit de remarquer que sous les hypothèses (Hyp), $\mathbb{P}(E_{n,\varepsilon}) > 0$ pour tout $n \in \mathbb{N}$ et tout $\varepsilon > 0$. □

Théorème 6.22 *Sous les hypothèses (Hyp), la simulation parfaite des matrices (max,plus) termine avec probabilité 1.*

Démonstration. C'est une conséquence du lemme de Borel-Cantelli : soit (Ev_n) une suite d'événements indépendants. Si $\sum \mathbb{P}(Ev_n) = +\infty$, alors presque sûrement un nombre infini de Ev_n se réalisent. Le lemme précédent montre que l'événement $E_{n,\varepsilon}$ a une probabilité non nulle, donc la simulation parfaite termine avec probabilité 1. □

À partir des profils stationnaires trouvés par l'algorithme 6.2, on peut trouver le marquage stationnaire du graphe d'événements. L'algorithme 6.3 a pour sortie un marquage distribué selon la distribution stationnaire des marquages.

Algorithme 6.3 : Trouver le marquage stationnaire

Données : $X(0)$ un représentant d'un profil stationnaire tel que $\max_i X(0)_i = 0$.
début
  $k \leftarrow 1$; $M \leftarrow M_0$;
  répéter
    $X(k) \leftarrow X(k-1) \otimes A(k)$;
    $k \leftarrow k+1$;
  jusqu'à $X_i(k) > 0,\ \forall i$;
  pour chaque *transition* t_i **faire**
    $n_i \leftarrow \max\{n | X_i(n) < 0\}$;
    $M \leftarrow N.(n_1, \ldots, n_n) + M_0$;
  finprch
  retourner M ;
fin

Comparaison des méthodes de simulation parfaite des graphes d'événements

Nous avons implémenté les deux méthodes de simulation parfaite des graphes à événements présentées, afin de les comparer. Les programmes ont été faits en Caml.

Pour la première méthode de simulation parfaite, il faut d'abord calculer les marquages de blocage pour toutes les transitions. Trouver un marquage de blocage se fait en $O(|\mathcal{T}|^3)$ ([52, Proposition 3.1]), donc la complexité de cette première étape est $O(|\mathcal{T}|^4)$. Ensuite, à chaque étape, il faut générer la transition à franchir si possible. En utilisant la méthode de l'anti-aliasing, ceci peut se faire en temps constant (le temps d'accès à une case d'un tableau). Pour une trajectoire (il y en a $|\mathcal{T}|$), le calcul du marquage atteint se fait en temps constant k, où k est le maximum du nombre de places d'entrée ou de sortie des transitions (en fait on a au pire $k = |\mathcal{P}|$, donc un temps linéaire). Chaque étape se fait donc en temps $O(k|\mathcal{T}|)$ et si le couplage a lieu en N étapes, la complexité de l'algorithme est $O(2Nk|\mathcal{T}|)$.

La deuxième méthode consiste à multiplier des matrices (max,plus). En général, la multiplication se fait en $O(n^3)$, où n est la dimension de la matrice. Si l'algorithme couple en N' étapes, la complexité est $O(N'n^3)$. Dans les simulations que nous avons effectuées, les matrices étaient creuses, donc la multiplication peut se faire en $O(n^2)$. Si dans le marquage initial, beaucoup des places sont vides, alors générer la matrice est coûteux : il faut calculer $A(0)^*$ (en reprenant les notations du chapitre 1). Si des places au contraire contiennent beaucoup de jetons, n sera plus grand que le nombre de transitions (la matrice à générer est une matrice compagnon). De plus, il faut garder en mémoire les temporisations de quelques tirs précédents. Ainsi, la complexité d'une étape dépend beaucoup du marquage initial. Pour les simulations que nous avons effectuées, sur des circuits, la taille de la matrice est $\max(|\mathcal{T}|, m)$, où m est le nombre de jetons, d'où la courbe dans la figure 6.9.

Les figures 6.8 et 6.9 représentent le nombre de pas moyen et le temps d'exécution de 10000 simulations parfaites, avec les marquages de blocage et les matrices (max,plus) pour un circuit de 40 transitions et un nombre m de jetons variant entre 1 et 80. Pour la simulation parfaite, le nombre d'états est de C_{n+m-1}^{n-1}. Donc le nombre d'états peut aller jusqu'à 3.819 10^{31}. Pour la simulation par matrices (max,plus), la taille des matrices est $\max(n, m)$.

Alors que le nombre d'itérations est plus petit pour les cas (max,plus), le temps de calcul est plus grand, et augmente alors très rapidement lorsque le nombre de jetons devient plus grand que le nombre de transitions. Cela est en partie dû à l'augmentation de la taille des matrices. Le

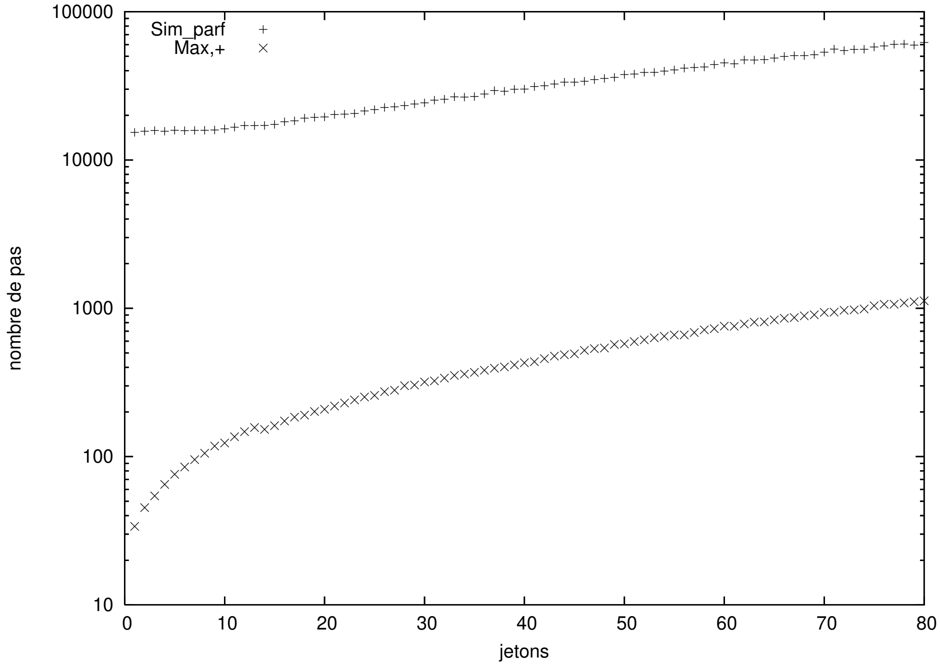


Fig. 6.8 – Nombre d'étapes moyen pour un circuit de longueur 40 en fonction du nombre de jetons.

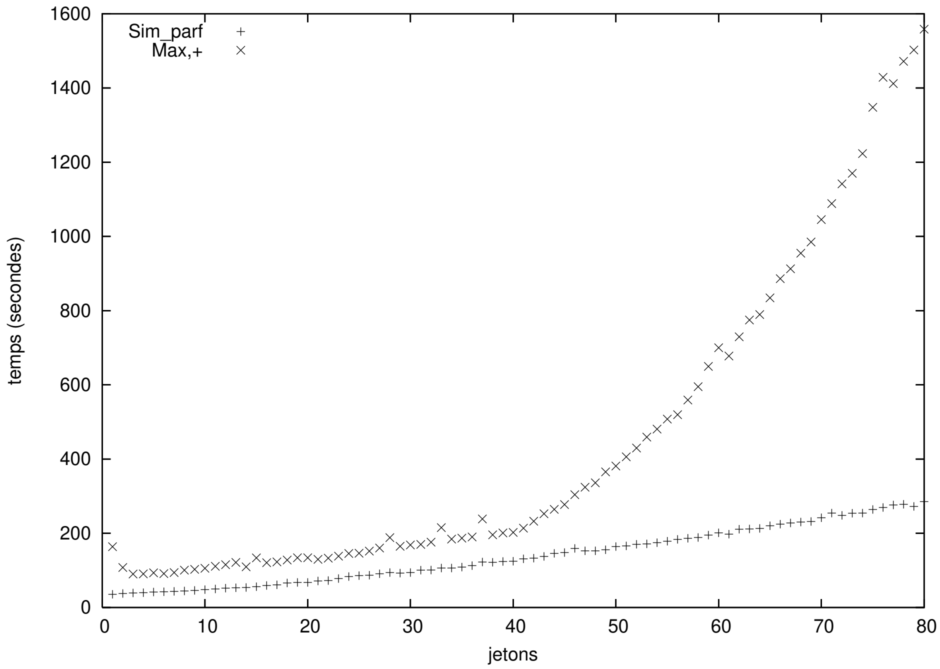


Fig. 6.9 – Temps d'exécution pour un circuit de longueur 40 en fonction du nombre de jetons (10000 simulations).

même comportement est observé pour différentes valeurs de n (même pour des petites valeurs).

Alors que la méthode par simulation parfaite est plus rapide, la méthode (max,plus) est plus générale : elle est valable pour des distributions de services plus générales. Cette méthode est aussi valable pour des matrices dont le support des matrices n'est pas fixé. En particulier, on peut faire de la simulation parfaite des tas de pièces grâce à la modélisation (max,plus). C'est ce que nous allons voir dans la section suivante.

6.3.2 Simulation parfaite des tas de pièces

Soit $\mathcal{A}$ un ensemble de pièces sur R un ensemble de colonnes (voir section 1.1.2. On définit le graphe de dépendance des pièces par $(\mathcal{A}, D)$, avec aDb si $R(a) \cap R(b) \neq \emptyset$. On suppose que ce graphe de dépendance est connexe.

On veut obtenir la probabilité stationnaire pour les profils des empilements. Ceci nous permettra en effet de calculer ensuite le débit, comme l'inverse de l'augmentation moyenne de la hauteur du tas lorsqu'on empile une nouvelle pièce.

Le lemme 6.19 montre que l'on peut effectuer les simulations uniquement à partir des profils extrémaux $(\mathrm{Prof}_i)_{i \in R}$, où Prof_i est le vecteur-ligne dont toutes les coordonnées sont égales à $-\infty$ sauf la i-ième, égale à 0.

Lemme 6.23 ([49], lemme 3.2) *Soit R un ensemble de colonnes, $\mathcal{A}$ un ensemble de pièces, représentées dans l'algèbre (max,plus) par les matrices $M(a), a \in \mathcal{A}$. On suppose que*

1. *$\forall r \in R$, $\exists a \in \mathcal{A}$ tel que $r \in R(a)$,*
2. *$(\mathcal{A}, D)$ est connecté.*

Soit $a_1, \ldots, a_n$ un chemin de $(\mathcal{A}, D)$ passant au moins une fois par chaque sommet. Alors $M(a_1) \otimes M(a_2) \otimes \cdots \otimes M(a_{n-1}) \otimes M(a_n) \otimes M(a_{n-1}) \otimes \cdots \otimes M(a_2) \otimes M(a_1)$ est une matrice de rang 1.

Dans la suite, on supposera toujours les hypothèses du lemme précédent (Hyp') satisfaites :

$$(Hyp') \qquad \begin{cases} \forall r \in R,\ \exists a \in \mathcal{A} \text{ tel que } r \in R(a), \\ (\mathcal{A}, D) \text{ est connecté.} \end{cases}$$

Une conséquence de ce lemme est la proposition :

Proposition 6.24 ([49], théorème 3.1) *Sous les hypothèses (Hyp'), si $(A_i)_{i \in \mathbb{N}}$ est une suite i.i.d. de variables aléatoires à valeur dans $\mathcal{A}$ et pour toute pièce a, $\mathbb{P}\{A_1 = a\} = p_a > 0$, alors la chaîne de Markov $(\mathit{Prof}(x_n))_n$ a une unique distribution stationnaire et converge en loi vers cette distribution stationnaire.*

Pour la simulation parfaite, on peut déjà se restreindre, pour l'espace des profils de départ aux profils extrémaux, grâce au lemme 6.19. Ou alors, on n'effectue que les multiplications de matrices, et on s'arrête quand on obtient une matrice de rang 1. Les deux méthodes sont équivalentes (elles ont la même complexité).

Proposition 6.25 *L'algorithme de la simulation parfaite s'arrête en un temps presque sûrement fini.*

Démonstration. On utilise à nouveau le lemme de Borel-Cantelli. On sait d'après le lemme 6.23 qu'il existe une suite σ de taille finie, k telle que $M(\sigma)$ est de rang 1. On considère comme événement A_n l'ensemble des suites de longueur k correspondant au tirage des pièces $kn+1$ à $k(n+1)$ qui donnent une matrice de rang 1. Toutes les pièces ont une probabilité strictement positive d'être choisies, donc $\mathbb{P}(A_n) = \mathbb{P}(A_0) > 0$, et comme on choisit les pièces de manière indépendantes, on en déduit que les événements A_n sont indépendants et équiprobables, et donc que $\sum \mathbb{P}(A_n) = +\infty$. On peut appliquer le lemme de Borel-Cantelli : on obtient presque sûrement une matrice de rang 1. Or, dès qu'on a une matrice de rang 1 et qu'on la multiplie à droite ou à gauche par d'autres matrices représentant des pièces (aucune des lignes ou des colonnes n'est identiquement nulle), le résultat reste une matrice de rang 1. L'algorithme de simulation parfaite s'arrête donc au bout d'un temps presque sûrement fini. □

L'algorithme de Propp et Wilson ne s'applique que dans le cas de chaînes de Markov à espace d'états fini. Il faut donc montrer la correction de cet algorithme pour le tas de pièces.

Proposition 6.26 *L'algorithme de Propp et Wilson est correct pour la simulation parfaite des tas de pièces.*

Démonstration. La preuve est exactement la même que celle du théorème 6.20. □

Ce résultat peut aussi être appliqué pour des hauteurs de pièces aléatoires. En effet, la terminaison de l'algorithme ne dépend pas des hauteurs des pièces mais seulement de leur support, et la preuve du théorème précédent ne dépend pas non plus de la hauteur des pièces, pourvu qu'il existe une distribution stationnaire pour le processus des profils, ce qui est le cas.

Application à la simulation parfaite des réseaux 1-bornés

On a vu que les réseaux à choix libres temporisés 1-bornés peuvent être modélisés par des tas de pièces. Comme les tas de pièces peuvent être simulés de manière exacte pour des hauteurs de pièces variables, on cherche maintenant à simuler les réseaux 1-bornés pour des distributions des temporisations plus générales. La suite des hauteurs de chaque pièce est une suite de variables aléatoires i.i.d. strictement positives.

Soit $\mathcal{H}$ le tas de pièces associé à $\mathcal{N}$ ayant une colonne par place. On a vu (lemme 3.30) que le marquage est déterminé de manière unique en fonction du profil du tas de pièces. Soit $v \in \mathbb{R}^{\mathcal{P}}$ un profil. On note $\mathrm{Marq}(v)$ le marquage associé à ce profil. Sur ce profil, les seules pièces que l'on peut empiler sont les transitions habilitées en $\mathrm{Marq}(v)$. Contrairement à ce qui se passe dans le cas des tas de pièces, on ne peut pas empiler les mêmes pièces sur deux profils différents. Pour qu'il y ait couplage arrière, il faut que partant de tous les profils possibles, un même profil soit obtenu par empilement de pièces admissibles. L'idée est de partir de tous les marquages atteignables M, auxquels on associe un profil v_M tel que $\mathrm{Marq}(v_M) = M$, et d'empiler les transitions possibles d'une suite de tirs aléatoirement générée jusqu'à obtenir un même profil, et que pour tous les empilements effectués, la matrice des pièces vraiment empilées soit de rang 1.

On considère toujours le réseau à choix libres vivant 1-borné temporisé et routé $(\mathcal{N}, M_0, \tau, u)$. On suppose que M_0 est un marquage atteignable à partir de tout marquage atteignable (il en est alors de même pour tous les marquages atteignables). Soit M un marquage atteignable et σ une suite de tirs telle que $M \xrightarrow{\sigma} M$ et telle que toutes les transitions apparaissent au moins une fois. Si l'on empile les pièces de hauteur 1 correspondant à la suite σ, le profil v du tas obtenu satisfait $\mathrm{Marq}(v) = M$.

On suppose que le routage u est un routage Bernoulli. On définit le processus de l'empilement des pièces. À chaque étape, on génère aléatoirement une pièce de la façon suivante :

- la pièce empilée correspond à la transition t avec probabilité $\frac{1}{C}\mathbb{P}(u_{\bullet t}(1) = t)$, où C est le nombre de clusters du réseau ;
- une fois la pièce t choisie, la hauteur de la pièce est aléatoirement générée selon la distribution de τ_t.

On empile effectivement la pièce ainsi générée sur un tas si son profil correspond à un marquage qui habilite t. On note $\mathcal{M}_{t,\varphi}$ la matrice représentant la pièce correspondant à la transition t de temporisation φ.

L'algorithme de couplage arrière pour les réseaux à choix libres 1-bornés est l'algorithme 6.4.

Algorithme 6.4 : Simulation par couplage arrière d'un réseau à choix libres 1-borné.

Données : $(\mathcal{N}, M_0, \tau, u)$ un réseau à choix libres vivant 1-borné,
$(t_0, \varphi_0), (t_1, \varphi_1), (t_2, \varphi_2), \ldots$ une suite de transitions avec leur temps de tir générés aléatoirement selon le procédé décrit ci-dessus, $N_1, N_2, \ldots$ une suite strictement croissante d'entiers.

début
  pour chaque *marquage atteignable M* **faire**
    $v_M \leftarrow$ profil tel que $\mathrm{Marq}(v_M) = M$;
  finprch
  $m \leftarrow 1$;
  répéter
    $\mathcal{M}_M \leftarrow \mathcal{E}$;
    pour chaque *marquage atteignable M* **faire**
      pour chaque *i de 1 à N_m* **faire**
        $v_M(-N_m + i) \leftarrow$ empiler si c'est possible $t_{N_m - i+1}$ sur $v_M(-N_m + i - 1)$;
        $\mathcal{M}_M \leftarrow \mathcal{M}_M \otimes \mathcal{M}_{t_{N_m-i+1}, \varphi_{N_m-i+1}}$;
      finprch
    finprch
    $m \leftarrow m + 1$;
  jusqu'à $v_M(0) = v_{M'}(0)$, *$\forall M, M'$ marquages atteignables et $\mathcal{M}_M$ est de rang 1 $\forall M$*;
  retourner $v_M(0)$;
fin

Lemme 6.27 *Avec le modèle défini ci-dessus et si le routage est équitable, l'algorithme de simulation parfaite termine en un temps fini avec probabilité 1.*

Démonstration. On peut considérer le couplage en deux étapes : tout d'abord, le couplage au niveau du marquage uniquement. Avec probabilité 1, en un temps fini, les profils obtenus $v_M(0), M \in \mathcal{R}(M_0)$ ont la même image par Marq. C'est une conséquence directe de la première partie du théorème 6.11. Ensuite, il y a couplage au niveau des profils. Mais, si tous les profils correspondent au même marquage, les pièces empilées sont les mêmes sur tous les profils.

Si le marquage est équitable, le graphe de dépendance des pièces est connexe, mais on ne peut pas appliquer le lemme 6.23 directement. L'idée est donc de se fabriquer une suite de tirs admissible dont la matrice est de rang 1.

Soit t une transition non conflictuelle et M_t son marquage de blocage. On peut atteindre en un temps presque sûrement fini M_t et donc supposer que le marquage atteint soit M_t. Soit σ une suite de tirs admissible de M_t vers M_t sans repasser par M_t. On note T_σ le tas de pièces correspondant à σ. On peut choisir un nombre fini de telles suites de tirs dont le support total est l'ensemble de toutes les transitions. On peut donc appliquer le lemme de Borel-Cantelli avec ces pièces. Elles ont toutes une probabilité non nulle d'être choisies, et peuvent être empilées les unes sur les autres sans problème d'admissibilité, puisqu'on part toujours du marquage M_t. □

Des deux lemmes précédents, on déduit :

Algorithme 6.5 : Trouver le marquage stationnaire d'un réseau à choix libres 1-borné

Données : v un profil stationnaire tel que $\max_i v(i) = 0$.
début
Empiler sur v des pièces générées aléatoirement comme décrit au-dessus jusqu'à ce que toutes les colonnes aient une hauteur supérieur à 0 ;
$v' \leftarrow$ profil obtenu en enlevant les pièces dont le profil supérieur est au dessus de 0 ;
retourner Marq(v) ;
fin

Théorème 6.28 *Exécuter l'algorithme 6.4 puis l'algorithme 6.5 permet de simuler de manière exacte le marquage stationnaire des réseaux à choix libres 1-bornés routés.*

Ce théorème est la conséquence des deux lemmes précédents. On peut donc utiliser les techniques de simulation parfaite pour évaluer le marquage stationnaire (et le débit) dans les réseaux à choix libres vivants 1-bornés routés. Comme nous l'avons vu au chapitre 5, on peut aussi calculer le marquage stationnaire et le débit de cette manière pour les réseaux obéissant à la politique de compétition. Il suffit de changer les distributions des temporisations et de calculer les fonctions de routage.

*
* *

Dans ce chapitre nous avons donné deux méthodes pour simuler de manière exacte des réseaux de Petri. La première utilise les propriétés structurelles des réseaux de Petri, comme les marquages atteignables, et les marquages de blocage. Dans le cas des graphes d'événements, on peut accélérer l'algorithme en ne considérant qu'un nombre linéaire de trajectoires en la taille du réseau. Les marquages de blocages apparaissent alors comme des marquages extrémaux. Les contre-exemples dans les autres cas montrent que les marquages de blocage ne sont plus des marquages extrémaux. Cette méthode n'est valable que pour des temporisations exponentiellement distribuées.

La deuxième méthode utilise les propriétés de l'algèbre (max,plus), et ainsi peut être utilisé pour les graphes d'événements, les tas de pièces et les réseaux à choix libres 1-bornés. Elle est valable pour des temporisations et des hauteurs de pièces avec des distributions plus générales.

La comparaison de ces deux méthodes pour les graphes d'événements montre que si le nombre d'étapes pour atteindre un marquage stationnaire est moindre pour la méthode de simulation (max,plus), le temps d'exécution est plus court avec l'autre méthode.

Troisième partie

Gérer les cycles dans les réseaux

Chapitre 7

Stabilité des opérations du network calculus

Le *network calculus* est un outil puissant d'analyse de performances dans les réseaux de communication, en particulier pour obtenir des garanties déterministes des performances. Cette théorie est basée sur des outils mathématiques, dont l'algèbre (min,plus) que nous avons présentée au chapitre 1. Pour ce faire, on modélise les paramètres du système par des courbes, représentant les flots dans le réseau ou les courbes de services des serveurs. C'est en combinant ces différentes courbes et en utilisant des opérateurs définis pour le calcul dans les réseaux qu'on peut calculer les performances du réseau. Ces performances peuvent être de différentes natures : le délai de transmission de paquets ou le nombre de paquets en attente de service par exemple. Ces opérations sont décrites dans deux livres de référence, [30, 65]. Le principal domaine applicatif du *network calculus* est celui de la qualité de service. Par exemple, ces techniques ont été utilisées pour la régulation optimale des flux vidéo dans [65, 89] et au contrôle de flux par fenêtres de taille fixe ([2, 1])

Malgré tout, les questions algorithmiques, à savoir, comment implémenter les opérations du *network calculus*, n'ont pas été abordées dans ces deux références.

Plusieurs tentatives ont toutefois été effectuées pour pallier ce manque. Pour le calcul de la convolution (min,plus), certains ([77, 78]) ont tenté une approche similaire à l'approche du calcul de la convolution classique qui utilise la transformée de Fourier ou de Laplace. Malheureusement, cette transformation, de Legendre-Fenchel, n'est injective que sur l'ensemble des fonctions convexes et ne peut être utilisée dans le cas général. Des résultats ([77]) ont été obtenus pour le calcul de la convolution des fonctions affines par morceaux ayant au plus 2 changements de pentes. Mais ces études au cas par cas paraissent aussi difficiles à généraliser en utilisant les mêmes méthodes.

D'un point de vue plus pratique, il existe des implémentations des opérations du *network calculus*. Mais elles ne sont implantées que pour des fonctions ayant des domaines de définition bornés, et les algorithmes paraissent être des transcriptions directes des formules de [30, 65], ce qui mène à de grandes complexités. On peut citer par exemple [88] qui décrit une implémentation dans Mathematica.

Dans ce chapitre, nous avons une approche différente. Nous commençons par chercher une bonne classe de fonctions pour lesquels nous allons ensuite implémenter les opérations. Après avoir défini les opérations du *network calculus*, nous nous intéressons à la stabilité de certaines classes de fonctions par ces opérations. Les fonctions les plus utilisées dans les calculs sont les fonctions affines par morceaux. C'est pourquoi nous nous y intéressons d'abord. Nous montrons alors que la clôture de ces fonctions est incluse dans l'ensemble des fonctions ultimement périodiques. Ces fonctions peuvent se représenter de manière fini, et apparaissent constituer la bonne classe de fonctions à considérer pour étudier les opérations du *network calculus* de manière algorithmique. Dans une dernière partie, nous donnons donc une représentation finie des fonctions ultimement périodiques et une implémentation des opérations sur ces fonctions. Tout au long de ce chapitre, nous considérons des fonctions définies de $\mathbb{N}$ dans $\mathbb{R}$. Dans [30, 65], les fonctions sont tantôt définies sur $\mathbb{N}$, tantôt sur $\mathbb{R}_+$, mais cela pose parfois des problèmes de stabilité des classes fonctions étudiées. De plus, lorsque cela ne pose pas de problème de stabilité, les preuves sont presque identiques à celles que nous donnons. Un tableau récapitulatif en fin de section 7.2 résume la stabilité des classes de fonctions en fonction de leur domaine de définition.

7.1 Calcul de garanties dans les réseaux

Nous rappelons dans cette section la théorie du network calculus. Cette théorie fait intervenir l'algèbre (min,plus) sur les fonctions. Ceci nous amènera à étudier la fermeture de certaines classes de fonctions par les opérations de l'algèbre (min,plus) sur les fonctions : minimum, addition et convolution (min,plus) et la clôture sous-additive.

7.1.1 Calculs dans les réseaux

Fonctions supérieurement contraintes

Définition 7.1 *Une fonction A est f-supérieurement contrainte si pour tous $t, s \in \mathbb{N}$,*

$$A(t+s) - A(t) \leq f(s). \tag{7.1}$$

Si A est un processus d'arrivées, c'est-à-dire, $A(t)$ est le nombre de paquets arrivés à la date t ou avant, alors $A(t+s) - A(t)$ désigne le nombre de paquets arrivés entre les dates t et $t+s$. Ce nombre ne peut jamais dépasser $f(s)$.

Un cas particulier très important des fonctions f-supérieurement contraintes sont les fonctions contraintes par une fonction affine. On dit qu'une fonction est (σ, ρ)-supérieurement contrainte si elle est supérieurement contrainte par la fonction affine $f : t \mapsto \rho t + \sigma$. Le coefficient ρ représente alors le taux d'arrivée moyen maximum des clients, et la constante σ la rafale maximum, c'est-à-dire le nombre maximum de paquets qui peuvent arriver simultanément. Ce sont ces fonctions qui vont nous intéresser en priorité.

Serveurs

Un serveur est un élément d'un réseau qui permet de transmettre des paquets d'une partie d'un réseau vers une autre.

Définition 7.2 *Un f-serveur est un serveur qui a servi au moins $f(t)$ paquets à la date t. Un serveur est exact s'il ne sert jamais plus de $f(t) - f(t-1)$ paquets à la date t.*

Un cas particulier est celui du lien conservatif de capacité c : à chaque instant, le serveur peut servir c paquets : il sert exactement c paquets s'il y en a au moins c en attente, et sinon, tous les paquets en attente.

Exemple 7.3 Regardons maintenant comment se comporte le flux de sortie des paquets quand une arrivée A (σ, ρ)-supérieurement contrainte définie sur $\mathbb{N}$ (les arrivées sont discrétisées) entre dans un lien conservatif de capacité c. On note $B(t)$ le nombre de paquets sortis à la date t. On a :

$$B(t) = \min(A(t), A(t-1)+c, A(t-2)+2c, \cdots, A(0)+ct). \tag{7.2}$$

L'équation (7.2) fait apparaître deux opérations : le minimum et l'addition. Nous allons maintenant définir un cadre algébrique pour faire ces calculs.

Calculs dans l'algèbre des fonctions (min,plus)

Le semi-anneau (min,plus) On définit le semi-anneau (min,plus) de la même manière que le semi-anneau (max,plus). Par souci de clarté, nous redonnons ces définitions.

Définition 7.4 *Le semi-anneau (min,plus), noté* $(\mathbb{R}_{\min}, \oplus, \otimes)$ *est défini sur* $\mathbb{R}_{\min} = \mathbb{R} \cup \{\infty\}$, *et est muni de deux opérations internes,* $\oplus$, *l'opération minimum, et* $\otimes$, *l'addition.*

L'élément nul est ∞, l'élément unité est 0. Les opérations $\oplus$ et $\otimes$ sont commutatives et associatives. De plus, $\otimes$ est distributive par rapport à $\oplus$.

Nous nous intéressons maintenant aux fonctions définies sur $\mathbb{N}$ et aux opérations possibles avec les opérations $\oplus$ et $\otimes$.

Fonctions (min,plus) On note $\mathcal{F}$ l'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{R}_{\min}$. On définit sur $\mathcal{F}$ deux opérations :

1. **Minimum :** $h = f \oplus g$ si $\forall t \in \mathbb{N}$, $h(t) = f(t) \oplus g(t)$.
2. **Convolution :** $h = f * g$ si $\forall t \in \mathbb{N}$, $h(t) = \bigoplus_{0 \le s \le t}(f(s) \otimes g(t-s)) = \min_{0 \le s \le t}(f(s) + g(t-s))$.

La convolution est à rapprocher de celle définie dans l'algèbre usuelle :

$$h(t) = \sum_{0 \le s \le t} f(s).g(t-s).$$

L'élément nul de $\mathcal{F}$ est la fonction constante égale à ∞ et l'élément unité est la fonction égale à ∞ partout sauf en 0, où il est égal à 0.

Exemple 7.5 Si l'on reprend l'exemple 7.3, l'équation (7.2) peut s'écrire $B = A * f$, avec $f : t \mapsto ct$.

Le minimum et la convolution sont des opérations internes, associatives et commutatives. L'élément neutre pour le minimum est $\epsilon : t \mapsto \infty$ et pour la convolution, $\mathbf{e} : 0 \mapsto 0$; $t \mapsto \infty$. De plus, la convolution est distributive par rapport au minimum et $\forall f \in \mathcal{F}$, $f * \epsilon = \epsilon * f = \epsilon$. Donc $(\mathcal{F}, \oplus, *)$ est un semi-anneau.

Une autre opération peut aussi être définie dans $\mathcal{F}$, la clôture sous-additive d'une fonction.

Clôture sous-additive des fonctions Une fonction f est sous-additive si $\forall t, s \in \mathbb{R}_+$, $f(t+s) \le f(t) + f(s)$.

Soit $f \in \mathcal{F}$. On définit récursivement f^n par : $f^0 = \mathbf{e}$ et $f^{n+1} = f^n * f$.

Définition 7.6 *Soit* $f \in \mathcal{F}$. *La clôture sous-additive de* f *est* f^*, *définie par :*

$$f^*(t) = \min_{n \in \mathbb{N}} f^n(t).$$

On peut aussi définir la clôture sous-additive comme

$$f^*(t) = \begin{cases} 0 & \text{si } t = 0, \\ \min[f(t), \inf_{0<s<t}\{f^*(s) + f^*(t-s)\}] & \text{sinon.} \end{cases} \tag{7.3}$$

ou

$$f^*(t) = \min_{k \in \mathbb{N}} \min_{t_1,\dots,t_k \in \mathbb{N}^* : t_1+\cdots+t_k = t} f(t_1) + \cdots + f(t_k). \tag{7.4}$$

On peut donner une interprétation de cette opération. Intuitivement, si l'on considère un graphe dont l'ensemble des nœuds est $\mathbb{N}$ et un arc de t vers $t+s$, $s > 0$, a un poids $f(s)$, alors $f^*(t)$ est le chemin de poids minimum de 0 à t.

La fonction f^* est inférieure à f par définition, et si $f(0) \geq 0$, c'est la plus grande fonction sous-additive qui soit inférieure à f : si g est une fonction de $\mathcal{F}$ sous-additive et $g \leq f$, alors $g \leq f^*$.

On pourra aussi étudier d'autres opérations sur les fonctions de $\mathcal{F}$. L'addition de deux fonctions peut aussi être définie sur ces fonctions : c'est l'addition point à point, $f+g : t \mapsto f(t)+g(t)$. On peut aussi définir le maximum de deux fonctions. Sur les fonctions finies (de $\mathbb{N}$ dans $\mathbb{R}$, et non plus $\mathbb{R}_{\min}$), on peut aussi définir la soustraction.

Garanties de services

Nous pouvons donc maintenant décrire dans un cadre plus formel les réseaux et faire des calculs de garantie de services. Redéfinissons maintenant les f-serveurs.

Définition 7.7 *Un serveur garantit une courbe de service f pour A si pour un processus d'entrée A, le processus de sortie est $B \geq A * f$. On appelle ce serveur un f-serveur. Ce serveur est exact si $B = A * f$, et si l'inégalité est vraie pour tout processus d'entrée, alors le f-serveur est dit universel.*

Lemme 7.8 ([30], Theorem 2.3.3) *La concaténation d'un f_1-serveur pour A et d'un f_2-serveur pour la sortie du f_1-serveur est un f-serveur pour A avec $f = f_1 * f_2$.*

Proposition 7.9 ([30], Theorem 2.3.4) *Soit A un processus d'entrée f_1-supérieurement contraint dans un f_2-serveur. On note B le processus des sorties. La longueur de la file d'attente à la date t est $q(t) = A(t) - B(t)$. Le délai virtuel d'un paquet arrivé à la date t est $d(t) = \inf\{d \geq 0 : B(t+d) \geq A(t)\}$. Alors,*

1. *B est f-contraint supérieurement avec $f(0) = 0$ et $f(t) = \sup_{s \geq 0}[f_1^*(t+s) - f_2(s)]$.*
2. *La longueur de la file d'attente à la date t est bornée par $\sup_{0 \leq s \leq t}[f_1^*(s) - f_2(s)]$.*
3. *Le délai virtuel à la date t est borné par $\inf\{d \geq 0 : f_1^*(s) \leq f_2(s+d), s = 1, \cdots, t\}$.*

Si l'on trace les fonctions f_1 et f_2 sur un même repère, la longueur maximale de la file d'attente est donnée par la distance maximale verticale entre f_1^* et f_2 et le délai virtuel maximal est donné par la distance maximale horizontale entre ces deux fonctions.

La figure 7.1 illustre ce fait.

Exemple 7.10 On reprend l'exemple d'un processus d'arrivées (σ, ρ)-supérieurement contraint et d'un lien conservatif de capacité c. On pose $f_1 : t \mapsto \rho t + \sigma$ et $f_2 : t \mapsto ct$. Le processus des sorties est f_3-supérieurement contraint avec $f_3 = f_1 * f_2$ (f_1 est sous-additive). Plus précisément,

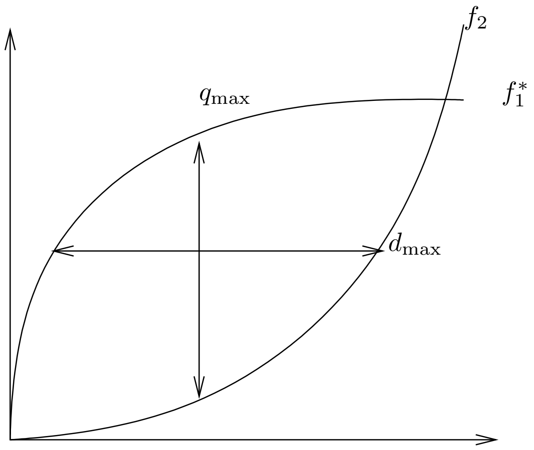


FIG. 7.1 – Bornes garanties.

$f_3(t) = \min_{0\leq s\leq t}[\rho.s + \sigma + c.(t-s)] = \min(\rho, c)t + \sigma$. La file d'attente est de taille au plus $\max_t[f_1(t) - f_2(t)]$. Cette valeur est finie si et seulement si $c \geq \rho$. Dans ce cas, $q(t) \leq \sigma - \lfloor c-\rho \rfloor$. Le délai virtuel est au plus $\sup_t \inf\{d \geq 0 : \rho s + \sigma \leq c.(s+d), s = 1, \ldots, t\}$. Ce nombre est aussi fini si $c \geq \rho$. On a alors $d(t) \leq \lceil \sigma/c \rceil$.

Une classe de fonctions utiles est la classe des fonctions affines. En effet, elle permet de modéliser pour un processus d'arrivées le taux moyen maximum d'arrivée des paquets, et la nombre maximal de paquets qui peuvent arriver simultanément. Nous allons maintenant étudier la fermeture de la classe des fonctions affines par toutes les opérations définies.

7.2 Fonctions ultimement affines et pseudo-périodiques

On s'intéresse maintenant à la stabilité de certains ensembles de fonctions par les opérations du *network calculus*. Nous commençons par les fonctions affines.

Lemme 7.11 *Soient f et g de $\mathbb{N}$ dans $\mathbb{R}$ affines. Alors, $f+g$ et $f*g$ sont affines.*
Soit $h : \mathbb{N} \to \mathbb{R}$ une fonction affine. Alors, h^* est affine.

Démonstration. C'est clair pour l'addition. Soient f_1 et f_2 deux fonctions affines : $f_1(t) = \rho_1 t + \sigma_1$ et $f_2(t) = \rho_2 t + \sigma_2$. Pour tout $t \in \mathbb{R}_+$,

$$\begin{aligned} f_1 * f_2(t) &= \inf_{0\leq s\leq t} (\rho_1 s + \sigma_1) + (\rho_2(t-s) + \sigma_2) && (7.5)\\ &= \min(\rho_1, \rho_2)t + \sigma_1 + \sigma_2. && (7.6)\end{aligned}$$

La convolution de deux fonctions affines est donc bien une fonction affine.

Soit f une fonction affine : $f(t) = \rho t + \sigma$. Alors $\forall t, s \in \mathbb{N}$,

$$f(t) + f(s) = \rho(t+s) + 2\sigma \geq f(t+s) \Leftrightarrow \sigma \geq 0, \qquad (7.7)$$

donc f est sous-additive si et seulement si $\sigma \geq 0$.

Sinon, $\sigma < 0$, $f(t) \geq tf(1) = t(\sigma + \rho)$, donc f^* est aussi affine. □

En revanche, la figure 7.2 montre que le minimum de deux fonctions affines n'est pas affine. De

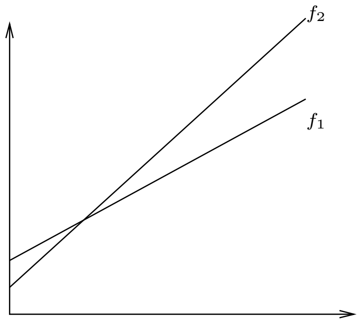


FIG. 7.2 – Minimum de deux fonctions affines.

plus, plus les coefficients directeurs des fonctions affines sont très proches, plus le rang à partir duquel le minimum des fonctions est affine est grand.

On étudie donc des classes plus larges de fonctions. La figure 7.2 suggère que nous étudiions les fonctions ultimement affines.

7.2.1 Clôture des fonctions ultimement affines par $+, -, *, \min, \max$

On appelle *support* d'une fonction $f \in \mathcal{F}$ l'ensemble $\{n \in \mathbb{N} \mid f(n) \neq \infty\}$.

Définition 7.12 *Une fonction f de $\mathbb{N}$ dans $\mathbb{R}$ est ultimement affine à partir de T si*

$$\exists \sigma, \rho \in \mathbb{R} \text{ tel que } \forall x \geq T, \quad f(x) = \rho x + \sigma.$$

Montrons tout d'abord un lemme qui va nous être très utile par la suite.

Lemme 7.13 *Soient f et g deux fonctions affines, de supports respectifs $[a, b] \subseteq \mathbb{N}$ et $[c, d] \subseteq \mathbb{N}$ et de coefficients directeurs sur ce support ρ_f et ρ_g avec $\rho_f \leq \rho_g$. Alors $f * g$ a pour support $[a + c, b + d]$ et*

$$f * g(a + c + x) = \begin{cases} f(a + x) + g(c) & \text{si } x \leq b \\ f(b) + g(c + x) & \text{si } b < x \leq b + d. \end{cases}$$

Démonstration. Par définition, $f * g(t) = \min_{0 \leq s \leq t} f(s) * g(t - s)$, et donc $f * g(t)$ est différent de $+\infty$ s'il existe s tel que $f(s) \neq +\infty$ et $g(t - s) \neq +\infty$. Donc il faut que $s \in [a, b]$ et $t - s \in [c, d]$. Le support de $f * g$ est donc bien $[a + c, b + d]$.

Soit $t \in [a + c, b + d]$. Alors,

$$\begin{aligned} f * g(t) = \min_{0 \leq s \leq t} f(s) + g(t - s) = \\ \min_{\max(a, t-d) \leq s \leq \min(b, t-c)} f(a) + \rho_f(s - a) + g(c) + \rho_g(t - s - c) \\ f(a) + g(c) - \rho_f(a) - \rho_g(c) + \rho_f \min(b, t - c). \end{aligned}$$

Donc,

$$f * g(t) = \begin{cases} f(a) + g(c) + \rho_f(t - a - c) & \text{si } t \leq b + c \\ f(a) + g(c) + \rho_f(b - a) + \rho_g(t - b - c) & \text{sinon.} \end{cases}$$

$\square$

Lemme 7.14 *Si f_1, f_2 sont deux fonctions ultimement affines de $\mathbb{N}$ dans $\mathbb{R}$ respectivement à partir de T_1 et T_2, alors*

1. $\min(f_1, f_2)$ *est affine par morceaux et ultimement affine à partir de* $T = \max(T_1, T_2, \frac{\sigma_2-\sigma_1}{\rho_1-\rho_2})$, *si* $\rho_1 \neq \rho_2$ *et* $\max(T_1, T_2)$ *sinon, et de coefficient directeur* $\min(\rho_1, \rho_2)$,
2. $\max(f_1, f_2)$ *est affine par morceaux et ultimement affine à partir de* $T = \max(T_1, T_2, \frac{\sigma_2-\sigma_1}{\rho_1-\rho_2})$, *si* $\rho_1 \neq \rho_2$ *et* $\max(T_1, T_2)$ *sinon, et de coefficient directeur* $\max(\rho_1, \rho_2)$,
3. f_1+f_2 *est affine par morceaux et ultimement affine à partir de* $\max(T_1, T_2)$, *et de coefficient directeur* $\rho_1 + \rho_2$,
4. f_1-f_2 *est affine par morceaux et ultimement affine à partir de* $\max(T_1, T_2)$, *et de coefficient directeur* $\rho_1 - \rho_2$,
5. $f_1 * f_2$ *est affine par morceaux et ultimement affine à partir de*

$$\max(T_1+T_2, \lceil \frac{\sigma_2 - \sigma_1 + \min_{0\leq s\leq T_1}(f_1(s) - \rho_2 s) - \min_{0\leq u\leq T_2}(f_2(u) - \rho_1 u)}{\rho_1 - \rho_2} \rceil),$$

si $\rho_1 \neq \rho_2$ *et* $T_1 + T_2$ *sinon, et de coefficient directeur* $\min(\rho_1, \rho_2)$.

Démonstration. On reprend les notations de l'énoncé. Il est clair que si f_1 et f_2 sont affines par morceaux, alors $\min(f_1, f_2)$, $\max(f_1, f_2)$, $f_1 + f_2$ et $f_1 - f_2$ le sont aussi.

En utilisant le lemme 7.13, la décomposition de f_1 et f_2 en fonctions affines définies sur un intervalle, et la distributivité du minimum par rapport à la convolution, on voit que $f_1 * f_2$ est affine par morceaux.

1. Supposons tout d'abord que f_1 et f_2 sont deux fonctions affines sur $\mathbb{R}$: $f_1(t) = \rho_1 t + \sigma_1$ et $f_2(t) = \rho_2 t + \sigma_2$. Si $\rho_1 \neq \rho_2$, alors il existe un unique point t_0 tel que $f_1(t_0) = f_2(t_0)$, et $t_0 = \frac{\sigma_2-\sigma_1}{\rho_1-\rho_2}$. Si $\rho_1 = \rho_2$, alors $\min(f_1, f_2)$ est soit f_1 si $\sigma_1 \leq \sigma_2$, soit f_2 sinon.
   Si f_1 et f_2 sont ultimement affines, alors il est maintenant facile de voir que $\min(f_1, f_2)$ est ultimement affine à partir de $T = \max(T_1, T_2, \lceil \frac{\sigma_2-\sigma_1}{\rho_1-\rho_2} \rceil)$ si $\rho_1 \neq \rho_2$ et $T = \max(T_1, T_2)$ si $\rho_1 = \rho_2$.
   Notons que T peut être arbitrairement grand, si ρ_1 et ρ_2 sont proches.
2. Idem 1.
3. L'addition de deux fonctions affines est une fonction affine.
4. Idem 3.
5. Soit $t \geq T_1 + T_2$. Calculons $f_1 * f_2(t)$.

$$\begin{aligned}
f_1 * f_2(t) &= \inf_{0\leq s\leq t} f_1(s) + f_2(t-s) \\
&= \inf_{0\leq s\leq T_1} f_1(s) + f_2(t-s) \oplus \inf_{T_1\leq s\leq t-T_2} f_1(s) + f_2(t-s) \\
&\oplus \inf_{t-T_2\leq s\leq t} f_1(s) + f_2(t-s) \\
&= \inf_{0\leq s\leq T_1} (f_1 s - \rho_2 s) + \rho_2 t + \sigma_2 \oplus \inf_{0\leq u\leq T_2} (f_2 u - \rho_1 u) + \rho_1 t + \sigma_1 \\
&\oplus \inf_{0\leq v\leq t-T_1-T_2} \rho_1(v+T_1) + \rho_2(t-v-T_1)\sigma_1 + \sigma_2
\end{aligned}$$

Les nombres $\inf_{0\leq s\leq T_1}(f_1 s-\rho_2 s)$ et $\inf_{0\leq u\leq T_2}(f_2 u-\rho_1 u)$ ne dépendent pas de t, et

$$\inf_{0\leq v\leq t-T_1-T_2} \rho_1(v+T_1)+\rho_2(t-v-T_1)\sigma_1+\sigma_2 =$$
$$\inf_{0\leq v\leq t-T_1-T_2} \rho_1(v)+\rho_2(t-T_1-T_2-v)+\rho_1 T_1+\rho_2 T_2+\sigma_1+\sigma_2 =$$
$$\min(\rho_1,\rho_2)t+(\rho_1-\min(\rho_1,\rho_2))T_1+(\rho_2-\min(\rho_1,\rho_2))T_2+\sigma_1+\sigma_2$$

est affine. Ainsi, $f_1 * f_2$ est ultimement le minimum de trois fonctions affines et est donc, par le point 1., elle-même une fonction ultimement affine.
Supposons, sans perte de généralité, que $\rho_1 \leq \rho_2$. Alors,

$$\inf_{0\leq u\leq T_2}(f_2 u-\rho_1 u) \leq \inf_{0\leq u\leq T_2}(f_2 s-\rho_2 s) \leq f_2(T_2)-\rho_2 T_2=\sigma_2. \tag{7.8}$$

Alors,

$$f_1 * f_2(t) = \inf_{0\leq s\leq T_1}(f_1 s-\rho_2 s)+\rho_2 t+\sigma_2 \wedge \inf_{0\leq u\leq T_2}(f_2 u-\rho_1 u)+\rho_1 t+\sigma_1, \tag{7.9}$$

et si $\rho_1 \neq \rho_2$, alors $f_1 * f_2$ est ultimement affine à partir de $\max(T_1+T_2,T)$ avec

$$T=\lceil \frac{\sigma_2-\sigma_1+\inf_{0\leq s\leq T_1}(f_1(s)-\rho_2 s)-\inf_{0\leq u\leq T_2}(f_2(u)-\rho_1 u)}{\rho_1-\rho_2}\rceil.$$

et à partir de T_1+T_2 sinon.

□

Lemme 7.15 *Soit $f:\mathbb{N}\to\mathbb{R}$ une fonction affine par morceaux ultimement affine. Alors, f^* n'est pas nécessairement ultimement affine.*

Démonstration. Exhibons un exemple. La fonction f définie par $f(x)=1/2+x/2$ si $x\leq 2$ et $f(x)=x-1/2$ sinon est représentée en figure 7.3. Alors, f^* n'est pas ultimement affine :

$$f^*(x)=\begin{cases} 3x/2 & \text{si } x \text{ est pair,}\\ 3(x-1)/2+1/2 & \text{sinon.}\end{cases} \tag{7.10}$$

On remarque que f^* est pseudo-périodique de période 2. Les calculs pour cette fonction seront détaillés dans l'exemple 7.20.

□

Nous allons voir dans le prochain paragraphe que l'étoile des fonctions ultimement affines sont ultimement pseudo-périodiques et que les fonctions ultimement pseudo-périodiques sont closes par le minimum, l'addition, la convolution et le passage à l'étoile.

7.2.2 Clôture des fonctions ultimement pseudo-périodiques par $+,-,*,\min$

Définition 7.16 *Une fonction ultimement pseudo-périodique est une fonction telle que*

$$\exists x_0\in\mathbb{N}, \exists(c,d)\in\mathbb{R}\times\mathbb{N}^* \text{ tel que } \forall x\geq x_0,\quad f(x+d)=f(x)+c. \tag{7.11}$$

Lemme 7.17 *Soient $f_1,f_2:\mathbb{N}\to\mathbb{R}$ deux fonctions pseudo-périodiques respectivement à partir de T_1 et T_2, de périodes respectives d_1 et d_2 et d'incréments respectifs c_1 et c_2. Alors*

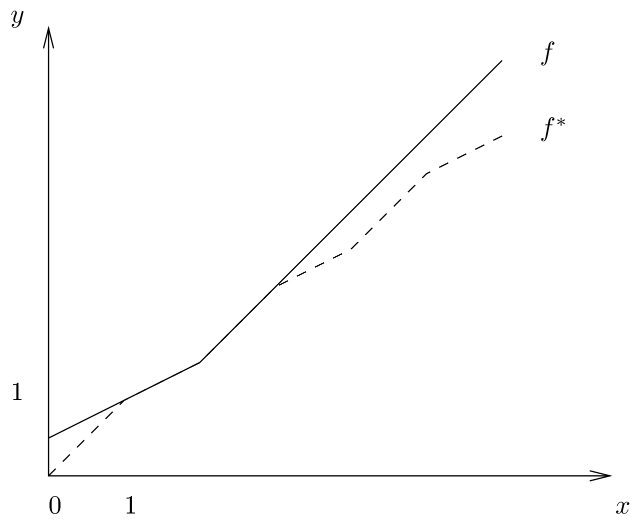


FIG. 7.3 – f^* n'est pas ultimement affine.

1. $\min(f_1, f_2)$ *est ultimement pseudo-périodique. Si* $d_2c_1 < d_1c_2$*, alors à partir d'un certain rang,* $\min(f_1, f_2) = f_1$ *(période* d_1*, incrément* c_1*). Si* $d_1c_2 < d_2c_1$*, alors ultimement* $\min(f_1, f_2) = f_2$ *(période* d_2*, incrément* c_2*). Sinon, la période est* $d_1 \vee d_2$ *et l'incrément* d_1c_2*.*
2. $\max(f_1, f_2)$ *est ultimement pseudo-périodique. Si* $d_2c_1 < d_1c_2$*, alors à partir d'un certain rang,* $\min(f_1, f_2) = f_2$ *(période* d_2*, incrément* c_2*). Si* $d_1c_2 < d_2c_1$*, alors ultimement* $\min(f_1, f_2) = f_1$ *(période* d_1*, incrément* c_1*). Sinon, la période est* $d_1 \vee d_2$ *et l'incrément* d_1c_2*.*
3. $f_1 + f_2$ *est ultimement pseudo-périodique à partir de* $T = \max(T_1, T_2)$*, et de période* $d = d_1 \vee d_2$*, et* $c = \frac{(d_2c_1+d_1c_2)}{d_1 \wedge d_2}$*,*
4. $f_1 - f_2$ *est ultimement pseudo-périodique à partir de* $T = \max(T_1, T_2)$*, et de période* $d = d_1 \vee d_2$*, et* $c = \frac{(d_2c_1-d_1c_2)}{d_1 \wedge d_2}$*,*
5. $f_1 * f_2$ *est ultimement pseudo-périodique de période* $d = d_1 \vee d_2$ *et d'incrément* $\min(d_1c_2, d_2c_1)$*,*

où $d_1 \vee d_2$ *est le plus grand multiple commun de* d_1 *et* d_2 *et* $d_1 \wedge d_2$ *est le plus petit diviseur commun de* d_1 *et* d_2*.*

Démonstration. On reprend les notations de l'énoncé. On pose $d = d_1 \vee d_2$, et $c'_1 = c_1 d/d_1$ et $c'_2 = c_2 d/d_2$. Les fonctions f_1 et f_2 sont toutes les deux ultimement pseudo-périodiques de période d et d'incréments respectifs c'_1 et c'_2.

1. Supposons tout d'abord que f_1 et f_2 sont pseudo-périodiques : $\forall x \in \mathbb{N}$, $f_i(x+d) = f_i(x)+c'_i$. Alors,

$$\begin{aligned} \min(f_1, f_2)(x + d) &= \min(f_1(x + d), f_2(x + d)) \\ &= \min(f_1(x) + c'_1, f_2(x) + c'_2). \end{aligned}$$

Si $c'_1 = c'_2$, alors

$$\min(f_1, f_2)(x + d) = \min(f_1, f_2)(x) + \frac{d}{d_2}c_2. \tag{7.12}$$

Sinon, supposons, sans perte de généralités, que $c'_1 < c'_2$. Alors,

$$\min(f_1(nd+i), f_2(nd+i)) = \min(f_1(i) + nc'_1, f_2(i) + nc'_2). \tag{7.13}$$

Pour tout $i \in \{0, \cdots, d-1\}$, il existe $n_i \in \mathbb{N}^*$ tel que pour tout $n \geq n_i$, $\min(f_1(i + nd), f_2(i+nd)) = f_1(i) + nc'_1$, et après calculs,

$$n_i = \left(\left\lceil \frac{(f_1(i) - f_2(i))}{(c'_2 - c'_1)} \right\rceil\right)_+ = \left(\left\lceil \frac{(f_1(i) - f_2(i))(d_1 \wedge d_2)}{d_1 c_2 - d_2 c_1} \right\rceil\right)_+. \tag{7.14}$$

Donc $\min(f_1, f_2)$ est ultimement pseudo-périodique à partir de $N = (\max_{i \in \{0,\cdots,d-1\}} n_i)_+ d$. Maintenant, si f_1 et f_2 sont ultimement pseudo-périodiques, le comportement des deux fonctions est pseudo-périodique à partir de $\max(T_1, T_2)$. Et donc, le comportement est pseudo-périodique à partir de $\max(T_1, T_2)$ si $c'_1 = c'_2$ et de

$$\begin{aligned} T = {} & \max(T_1, T_2) + \\ & \max_{i \in \{0,\cdots,d-1\}} \left(\left\lceil \frac{(f_1(\max(T_1,T_2)+i) - f_2(\max(T_1,T_2)+i))(d_1 \wedge d_2)}{d_1 c_2 - d_2 c_1} \right\rceil\right)_+ d \end{aligned}$$

sinon. À partir de T, le comportement est celui d'une des deux fonctions. Si $c'_1 < c'_2$, alors la période est d_1 et l'incrément c_1. Si $c'_1 = c'_2$, alors à partir de $\max(T_1, T_2)$, $\min(f_1, f_2)(x) = \min(f_1(x), f_2(x))$ est pseudo-périodique de période $d = d_1 \vee d_2$ et $c = \frac{d}{d_2} c_2$.

2. Idem 1. sauf que c'est max, pour les fonctions.
3. $\forall x \geq \max(T_1, T_2)$,

$$\begin{aligned} (f_1 + f_2)(x+d) &= f_1(x+d) + f_2(x+d) \\ &= f_1(x) + \frac{d}{d_1} c_1 + f_2(x) + \frac{d}{d_2} c_2 = (f_1+f_2)(x) + \frac{(d_2 c_1 + d_1 c_2)}{d_1 \wedge d_2}. \end{aligned}$$

4. Idem 3. sauf que $c = \frac{(d_2 c_1 - d_1 c_2)}{d_1 \wedge d_2}$.
5. Montrons tout d'abord que la convolution de deux fonctions pseudo-périodiques est ultimement pseudo-périodique. Pour tout $t \geq d$, on a

$$\begin{aligned} (f_1 * f_2)(t+d) &= \min_{0 \leq s \leq t+d} f_1(s) + f_2(t+d-s) \\ &= \min(\min_{0 \leq s \leq t} f_1(s) + f_2(t+d-s), \min_{d \leq s \leq t+d} f_1(s) + f_2(t+d-s)) \\ &= \min(\min_{0 \leq s \leq t} f_1(s) + f_2(t-s) + c'_2, \min_{0 \leq u \leq t} f_1(t+d-u) + f_2(u)) \\ &= \min(\min_{0 \leq s \leq t} f_1(s) + f_2(t-s) + c'_2, \min_{0 \leq u \leq t} f_1(t-u) + f_2(u) + c'_1) \\ &= \min(f_1 * f_2(t) + c'_2, f_1 * f_2(t) + c'_1) \\ &= f_1 * f_2(t) + \min(c'_1, c'_2). \end{aligned}$$

Donc, $f_1 * f_2$ est ultimement pseudo-périodique à partir de d et de période d et d'incrément $\min(c_1, c_2)$.

Considérons maintenant que les fonctions sont ultimement pseudo-périodiques à partir de T_1 et T_2 respectivement. Pour tout $t \geq T_1 + T_2 + d$,

$$\begin{aligned}
(f_1 * f_2)(t) &= \min_{0 \leq s \leq t} f_1(s) + f_2(t-s) \\
&= \min_{0 \leq s \leq T_1} f_1(s) + f_2(t-s) \\
&\oplus \min_{T_1 \leq s \leq t-T_2} f_1(s) + f_2(t-s) \\
&\oplus \min_{t-T_2 \leq s \leq t} f_1(s) + f_2(t-s) \\
&= \min(\min_{0 \leq s \leq T_1} f_1(s) + f_2(t-s), \min_{0 \leq u \leq T_2} f_1(t-u) + f_2(u)) \\
&\oplus \min_{T_1 \leq s \leq t-T_2} f_1(s) + f_2(t-s) \\
&= \min(\min_{0 \leq s \leq T_1} f_1(s) + f_2(t-s), \min_{0 \leq u \leq T_2} f_1(t-u) + f_2(u)) \\
&\oplus \min_{0 \leq s \leq t-T_2-T_1} f_1(T_1+s) + f_2(t-s-T_1).
\end{aligned}$$

Posons $f_1'(t) = f_1(t+T_1)$, $f_2'(t) = f_2(t+T_2)$ et $t' = t - T_1 - T_2$. La dernière ligne de l'équation précédente devient alors :

$$\min_{0 \leq s \leq t-T_2-T_1} f_1(T_1+s) + f_2(t-s-T_1) = \min_{0 \leq s \leq t'} f_1'(s) + f_2'(t'-s) = f_1' * f_2'(t').$$

Or, f_1' et f_2' sont pseudo-périodiques. Donc, $f_1' * f_2'$ est ultimement pseudo-périodique à partir de d. De plus,

$$\begin{aligned}
&\min(\min_{0 \leq s \leq T_1} f_1(s) + f_2(t+d-s), \min_{0 \leq u \leq T_2} f_1(t+d-u) + f_2(u)) = \\
&\qquad \min(\min_{0 \leq s \leq T_1} f_1(s) + f_2(t-s) + c_2', \min_{0 \leq u \leq T_2} f_1(t-u) + f_2(u) + c_1')
\end{aligned}$$

est aussi ultimement pseudo-périodique, en tant que minimum de fonctions ultimement pseudo-périodiques. Donc, $f_1 * f_2$ est le minimum de 3 fonctions ultimement pseudo-périodiques. C'est une fonction ultimement pseudo-périodique, de période $d = d_1 \vee d_2$, d'incrément $\min(c_1', c_2')$.

□

Soit $f : \mathbb{N} \to \mathbb{R}$ une fonction ultimement pseudo périodique de période d, d'incrément c à partir du rang T. On construit $\mathcal{G}(f)$ le graphe (orienté et pondéré) associé à f de la manière suivante :

- l'ensemble des nœuds de $\mathcal{G}(f)$ est $\{1, 2, \ldots, T+d-1\}$;
- pour tout $i \in \mathbb{N}_{T+d-2}$, il y a un arc du nœud i vers le nœud $i+1$ de poids 0 ;
- pour tout $i \in \mathbb{N}_{T+d-1}$, il y a un arc nœud i vers le nœud 1 de poids $f(i)$;
- il y a un arc de $T+d-1$ vers T de poids c.

La figure 7.4 représente la forme du graphe associé à une fonction f.

Proposition 7.18 *Soit $f : \mathbb{N} \to \mathbb{R}$ une fonction ultimement pseudo-périodique. Soit A la matrice (min,plus) associée au graphe $\mathcal{G}(f)$. Alors, pour tout $t \in \mathbb{N}$, $f^*(t) = A_{11}^t$.*

Démonstration. On rappelle la définition de f^* :

$$f^*(t) = \min_{k \in \mathbb{N}} \min_{t_1, \ldots, t_k \in \mathbb{N}^* : t_1 + \cdots + t_k = t} f(t_1) + \cdots + f(t_k). \tag{7.15}$$

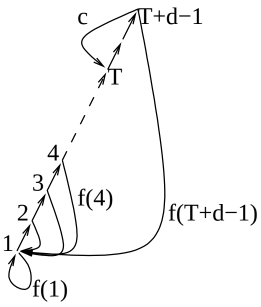


FIG. 7.4 – Graphe associé à une fonction ultimement pseudo-périodique.

Pour tout $t \in \mathbb{N}$, il existe un unique circuit de 1 vers lui-même, qui ne repasse pas par 1 de longueur t : il existe un unique entier t' tel que $0 \leq t - t' < d$ et $t = t' + kd$, $k \in \mathbb{N}$. Ce circuit est alors composé du circuit $1, 2, \dots, t'$ et de k circuits $T, T+1, \dots, T+d-1$. Le poids de ce circuit est alors $f(t') + kc = f(t' + kd) = f(t)$. Tout chemin de 1 vers lui-même peut se décomposer en circuits qui ne repassent pas par 1. Le poids d'un tel chemin, de longueur t, peut alors s'écrire sous la forme $f(t_1) + f(t_2) + \cdots + f(t_k)$ avec $t_1 + t_2 + \cdots + t_k = t$. Réciproquement, tout réel qui peut s'écrire $f(t_1) + f(t_2) + \cdots + f(t_k)$ est le poids d'un chemin de 1 vers lui-même de longueur $t_1 + t_2 + \cdots + t_k$.

Or, $A_{11}(t)$ est le poids minimum d'un chemin de 1 vers lui-même et de longueur t. Donc $f^*(t) = A^t_{11}$. □

Corollaire 7.19 *Soit $f : \mathbb{N} \to \mathbb{R}$ une fonction ultimement pseudo-périodique, alors f^* est ultimement pseudo-périodique.*

Démonstration. Par construction, le graphe $\mathcal{G}(f)$ est fortement connexe, et donc la matrice A qui lui est associé est irréductible. Soient d^* sa cyclicité et λ son unique valeur propre. Par le théorème fondamental des matrices (min,plus), il existe un rang T^* tel que $\forall t \geq T^*$, $A^{t+d^*} = A^t + \lambda d^*$. Donc $f^*(t + d^*) = f^*(t) + \lambda d^* = f^*(t) + c^*$ avec $c^* = \lambda d^*$. Donc f^* est ultimement pseudo-périodique de période d^* et d'incrément c^*.

La valeur propre de la matrice A est le poids moyen d'un circuit de poids moyen minimum. C'est donc, par construction $\min(c/d, \min_{1 \leq t \leq T+d-1} f(t)/t)$. Mais alors, c'est aussi $c^*/d^* = \inf_{t \in \mathbb{N}^*} f(t)/t$.

□

Exemple 7.20 Reprenons l'exemple de la figure 7.3. Le graphe construit est représenté en figure 7.5. Le cycle de poids moyen le plus faible est de longueur 2 et de poids total 3/2, donc $c^* = 3/2$ et $d^* = 2$.

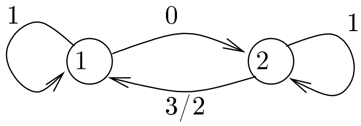


FIG. 7.5 – Graphe associé à la fonction de la figure 7.3.

Rang à partir duquel f^* est périodique

Dans le cas général, trouver le temps de couplage exact d'une matrice (min,plus) est NP-complet, mais il existe des bornes d'approximation ([18]). Le temps de couplage peut néanmoins être arbitrairement grand. Dans [18], le temps de couplage est étudié pour des matrices de cyclicité 1. Pour se ramener au cas d'une matrice de cyclicité d, il suffit alors de calculer le temps de couplage pour la puissance d-ième de cette matrice. Dans le cas qui nous occupe, la cyclicité de la matrice est rarement 1 et les formules pour la cyclicité 1 ne peuvent s'appliquer directement. Néanmoins, le graphe associé à la matrice étant très simple, de forme fixée, il paraît plus judicieux d'étudier le temps de couplage directement. De plus, nous ne nous préoccupons que du temps de couplage pour la coordonnée $(1, 1)$.

On distingue plusieurs cas, suivant le graphe critique de la matrice.

Le graphe critique est le circuit ne passant pas par 1**.** La cyclicité de la matrice (et la période de f^*) est alors d et l'incrément est c. Soit $c_1, \dots, c_k$ l'ensemble des circuits dont le poids moyen est le plus faible en dehors du graphe critique, soit ϵ tel que leur poids moyen est $c/d + \epsilon$. Le comportement de A^n devient périodique à partir d'un rang N, lorsque le chemin de poids minimal de longueur n de 1 vers 1 passe par T pour tout $n \geq N$.

Soit $C = \max_{T \leq t \leq T+d-1} f(t)$. Un chemin passe nécessairement par T lorsque sa longueur ℓ est telle que $\ell(f(t)/t + \epsilon) \geq \ell f(t)/t + C$, c'est-à-dire, $\ell \geq C/\epsilon$ et $N = C/\epsilon$.

Le graphe critique contient des circuits passant par 1 **et est connexe.** La cyclicité de la matrice est le plus grand diviseur commun d^* de la longueur de ces circuits et l'incrément est le poids moyen de ces circuits. Pour qu'il y ait couplage, il faut d'une part que tout multiple de d^* puisse se décomposer en la somme des longueurs des circuits critiques. Si t_1 est la taille maximale d'un circuit critique simple et t_2 la taille du plus petit circuit critique, alors ceci est possible à partir du rang $(t_1/d - 1)t_2$ ([33]). D'autre part, il faut que le chemin de poids minimal passe par les circuits critiques. Soit $C = \max_{1 \leq t \leq d^*-1} f(t)$. Soit ϵ tel que $c^*/d^* + \epsilon$ est égal au poids moyen des circuits de poids moyen le plus faible en dehors des circuits critiques. Alors le temps de couplage est aussi supérieur à C/ϵ. Le temps de couplage est donc $N = \max((t_1/d - 1)t_2, C/\epsilon)$.

Le graphe critique n'est pas connexe. Nécessairement, le circuit ne passant pas par 1 est critique, et les autres circuits critiques passent par 1 (il y en a au moins un) et sont de longueur inférieure à T. Soit d' le plus grand diviseur commun des circuits critiques passant par n_0. Si $d' = 1$, alors $d^* = 1$, et il y a couplage dès qu'on peut décomposer tout entier en somme de nombres longueur des circuits critiques, ce qui peut se faire à partir du rang $(t_1/d - 1)t_2$, en reprenant les mêmes notations que pour le cas précédent. Sinon, la cyclicité de A est alors $d^* = d \vee d'$. Le couplage est alors le maximum des temps de couplage des deux premiers cas.

Dans tous les cas, on a une formule du type $n = C/\epsilon$, où ϵ est la différence entre le deuxième poids moyen le plus faible et le poids moyen des circuits critiques, et $C \leq \max_{1 \leq t \leq T+d-1} f(t)$.

La figure 7.6 résume les différents cas et la figure 7.7 donne un exemple dans lequel la période de la fonction est le *ppcm* des longueurs des circuits critiques.

Contrairement au cas de l'addition, du minimum ou de la convolution de deux fonctions ultimement pseudo-périodiques, la période de l'étoile d'une fonction ultimement pseudo-périodique n'est pas liée à la période de cette fonction. On remarque aussi que le calcul de l'étoile d'une fonction ultimement affine est un cas particulier des fonctions ultimement pseudo-périodiques. L'exemple suivant montre aussi que l'étoile d'une fonction ultimement pseudo-périodique peut être ultimement affine.

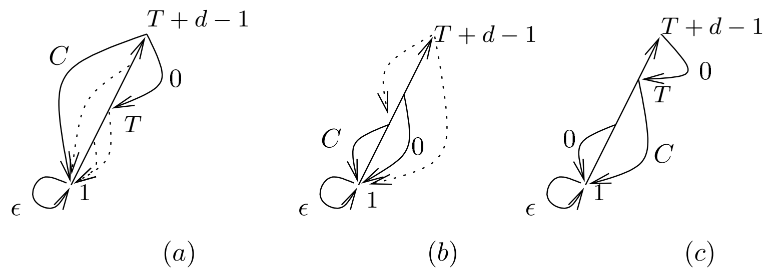


FIG. 7.6 – Temps de couplage pour la clôture sous-additive. (a) : Tant que $t\epsilon < C$, c'est le chemin correspondant à $tf(1)$ qu'il faut choisir pour minimiser le poids. (b) c'est la même chose que pour le cas (a), mais le circuit de poids C est à choisir parmi les circuits de taille inférieure au circuit critique. (c) le graphe critique n'est pas connexe, et le couplage doit être atteint pour les deux composantes connexes avant d'être atteint pour le graphe en entier.

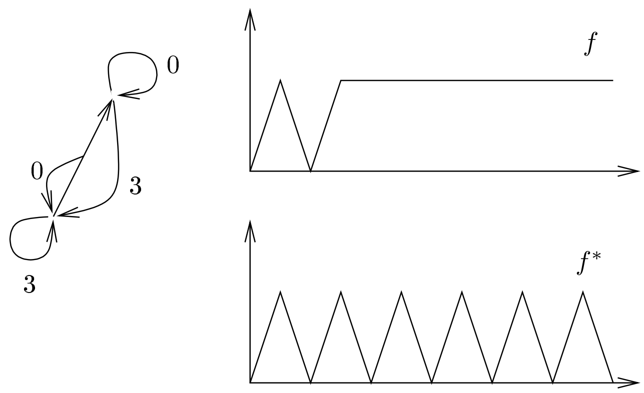


FIG. 7.7 – Cas où le graphe critique n'est pas connexe. Un circuit critique est de longueur 1 et l'autre de longueur 2, et la période de f^* est 2.

Exemple 7.21 La fonction f est représentée sur la figure 7.8 est ultimement pseudo-périodique de période 2 à partir du rang 3. On a $f(0) = f(1) = 1$, $f(2) = 4/3$ et $f(3) = 2$. À partir du rang 2, la fonction f^* est affine et $f^*(x) = 2x/3$.

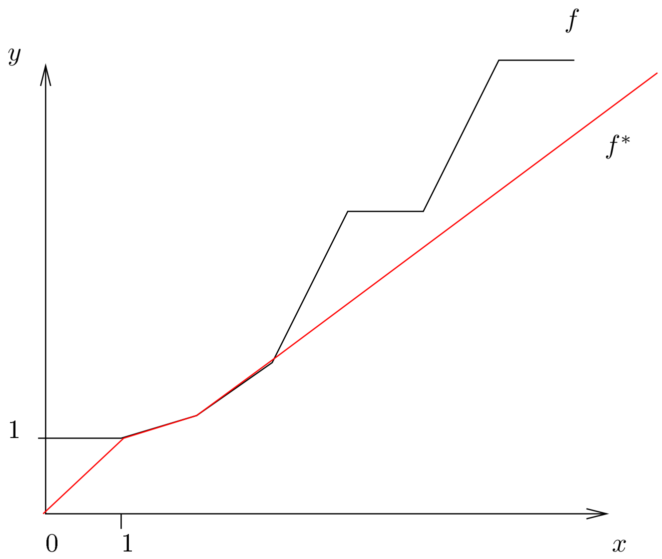


FIG. 7.8 – f^* est ultimement affine.

Les résultats obtenus dans cette section sont résumés en figure 7.9. On obtient les mêmes résultats pour les fonctions affines par morceaux ultimement affines et ultimement pseudo-périodiques si les fonctions sont définies de $\mathbb{Q}_+$ dans $\mathbb{Q}_+$. On met au même dénominateur les points de changement de pente et tout se passe de la même manière que pour les fonctions définies sur $\mathbb{N}$. Si les fonctions affines par morceaux ultimement affines sont définies de $\mathbb{R}_+$ vers $\mathbb{R}_+$, les opérateurs minimum, addition et convolution sont stables, mais pas la clôture sous-additive. Les fonctions ultimement pseudo-périodiques définies sur $\mathbb{R}_+$ ne sont stables par aucune opération du network calculus. Par exemple, le minimum de deux fonctions périodiques n'est pas périodique si les périodes sont incommensurables (leur rapport est irrationnel).

7.3 Représentation des fonctions affines par morceaux

Dans cette section, nous abordons les aspects algorithmiques des opérations du network calculus. La section précédente nous a permis d'exhiber des classes intéressantes de fonctions stables par ces opérations. Nous considérons ici les fonctions ultimement affines et ultimement pseudo-périodiques définies sur $\mathbb{N}$ à valeurs dans $\mathbb{R}$. Les algorithmes pour les fonctions définies sur $\mathbb{Q}_+$ et à valeurs dans $\mathbb{Q}$ peuvent se déduire des algorithmes présentés dans ce qui suit. Ces fonctions de $\mathbb{N}$ dans $\mathbb{R}$ admettent des représentations finies.

Une première solution consiste à représenter une fonction par ses premières valeurs (jusqu'au comportement régulier), les valeurs suivantes se déduisant ensuite. Mais cette manière de faire est très gourmande en mémoire : pour une fonction qui est le minimum de deux fonctions affines, le comportement régulier peut avoir lieu à partir d'un rang arbitrairement grand. Une autre manière de faire est de représenter les fonctions par leurs points de discontinuité. Ainsi, le minimum de deux fonctions affines serait représenté par deux points : la valeur en zéro, et en le point

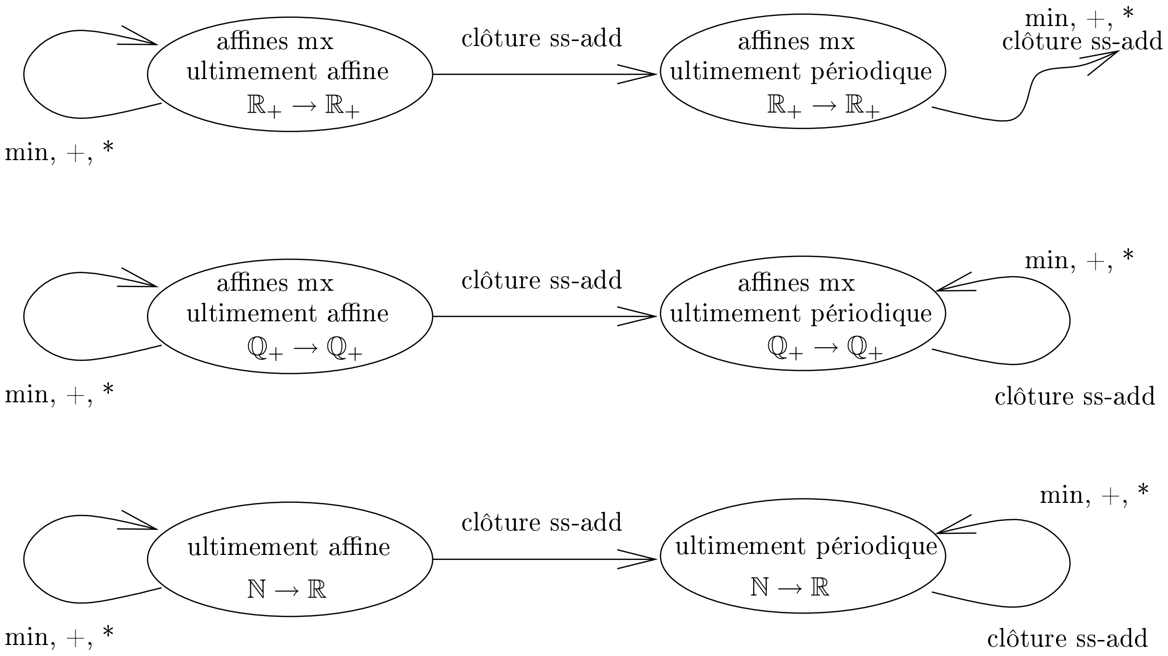


FIG. 7.9 – Stabilité des opérations min, $+, *$ et clôture sous-additive selon le domaine de définition des fonctions.

d'intersection des deux fonctions, s'il existe. La difficulté réside alors en trouver les points de changement de coefficient des fonctions, en particulier pour la convolution et le passage à l'étoile.

7.3.1 Représentation d'une fonction ultimement affine

Nous nous intéressons tout d'abord aux fonctions ultimement affines, et au calcul du minimum, de l'addition et de la convolution de telles fonctions.

Les fonctions sont définies sur $\mathbb{N}$, mais on utilise une extrapolation affine par morceaux sur $\mathbb{Q}$.

On représente une fonction par une liste de triplets (x, y, ρ), où (x, y) représente un point et ρ un coefficient directeur. Si $f := [t_1, \ldots, t_k]$, où $t_k = (x_k, y_k, \rho_k)$, alors pour tout $x \in \mathbb{Q}$, $f(x) = y_i + \rho_i \cdot (x - x_i)$ si $x \in [x_i, x_{i+1})$, avec $x_{k+1} = \infty$. Il est clair que pour toute fonction affine par morceaux (ultimement affine), cette représentation est finie.

Addition de deux fonctions ultimement affines

Soit $f_1 := [t_1, \ldots, t_k]$, avec $t_i = (x_i, y_i, \rho_i)$ et $f_2 := [t'_1, \ldots, t'_\ell]$, avec $t'_i = (x'_i, y'_i, \rho'_i)$. Pour additionner les deux fonctions, il suffit de prendre les points de changement de pente, et pour chacun de ces points, la fonction est la somme des valeurs des fonctions en ce point et le coefficient directeur, la somme des coefficients directeurs. Si tous les x_i et x'_i sont différents, la taille de la représentation est exactement $k + \ell - 1$, d'où le lemme suivant.

Lemme 7.22 *Soient f_1 et f_2 des fonctions de représentations de tailles respectives k et ℓ. La fonction $f = f_1 + f_2$ a une représentation de taille $k + \ell - 1$, qui peut se calculer en temps $O(k + \ell)$, est donnée par l'algorithme 7.1*

Algorithme 7.1 : Addition de deux fonctions

Données : $f_1 := [(x_1^1, y_1^1, \rho_1^1), \ldots, (x_k^1, y_k^1, \rho_k^1)]$ et $f_2 := [(x_1^2, y_1^2, \rho_1^2), \ldots, (x_\ell^2, y_\ell^2, \rho_\ell^2)]$.
Résultat : $f := [(x_1, y_1, \rho_1), \ldots, (x_n, y_n, \rho_n)]$ telle que $f = f_1 + f_2$.
début
    $f \leftarrow [\,]$;
    pour chaque $x \in \{x_i^1\} \cup \{x_j^2\}$ *pris dans l'ordre* **faire**
        $\rho_i(x) \leftarrow$ la pente de f_i au point x;
        $f \leftarrow f :: [(x, f_1(x) + f_2(x), \rho_1(x) + \rho_2(x))]$;
    finprch
    retourner f
fin

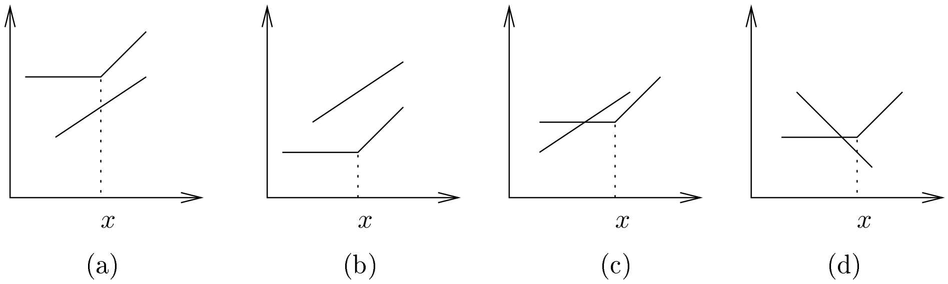


FIG. 7.10 – Minimum de deux fonctions. (a) x n'est pas ajouté dans la représentation du minimum ; (b) x est ajouté dans la représentation du minimum ; (c) on ajoute le point d'intersection des fonctions et x ; (d) on ajoute seulement le point d'intersection des courbes.

Cet algorithme peut se faire en temps linéaire car, en prenant les points de changement de pente dans l'ordre, on trouve $\rho_i(x)$ en temps constant. De plus, pour les représentations d'une fonction, les points de changement de pente sont triés.

Minimum de deux fonctions ultimement affines

Dans ce cas, on peut créer des nouveaux points de changement de pente, mais aussi en éliminer. Le principe de l'algorithme de calcul du minimum de deux fonctions est de regarder si, pour chaque segment de f_1 et chaque segment de f_2 ayant des supports non disjoints, ces segments se coupent ou pas. S'ils se coupent, alors il faut ajouter un nouveau point de changement de pente. On garde aussi les changements de pente des fonctions aux points où elles satisfont le minimum des deux fonctions.

L'algorithme 7.2 renvoie la représentation du minimum de deux fonctions. Il y a exactement $k + \ell$ parcours de la boucle et dans chaque passage, au plus deux points ajoutés. Le nombre de changements de pente est donc au plus de $2(k + \ell)$.

Lemme 7.23 *La complexité du calcul du minimum de deux fonctions de représentation taille k et ℓ est $O(k + \ell)$. Le minimum a une représentation de taille au plus $2(k + \ell)$.*

Lemme 7.24 *On peut calculer le minimum de n fonctions de représentation de taille k en $O(n^2k)$.*

Algorithme 7.2 : Minimum de deux fonctions ultimement affines

Données : $f_1 := [(x_1^1, y_1^1, \rho_1^1), \dots, (x_k^1, y_k^1, \rho_k^1)]$ et $f_2 := [(x_1^2, y_1^2, \rho_1^2), \dots, (x_\ell^2, y_\ell^2, \rho_\ell^2)]$.
Résultat : $f := [(x_1, y_1, \rho_1), \dots, (x_n, y_n, \rho_n)]$ telle que $f = \min(f_1, f_2)$.
début
    $f \leftarrow []$;
    $m \leftarrow 1$;
    pour chaque $x \in \{x_i^1\} \cup \{x_j^2\}$ *pris dans l'ordre* **faire**
        si $f_m(x) < f_{3-m}(x)$ *ou* ($f_m(x) = f_{3-m}(x)$ *et la pente de la fonction* f_m *en* x *est plus petite que celle de* f_{3-m}) **alors**
            si $x = x_i^m$ **alors**
                $f \leftarrow f :: [(x_i^m, y_i^m, \rho_i^m)]$;
            finsi
        sinon
            Ajouter le point d'intersection des deux fonctions à f;
            $m \leftarrow 3 - m$;
            si $x = x_i^m$ **alors**
                $f \leftarrow f :: [(x_i^m, y_i^m, \rho_i^m)]$;
            finsi
        finsi
    finprch
fin

Démonstration. Soit $X_n(k)$ la complexité pour calculer le minimum de n fonctions de taille k. On a $X_{2n}(k) = 2kn + X_n(4k)$. Alors $X_{2^i}(k) = \sum_{j=0}^{i} 2.2^j kn = 2^{i+1}kn$ Or, $i = \log n$, d'où le résultat. □

Convolution de deux fonctions ultimement affines

Ce cas est plus difficile, car on ne calcule plus point à point, mais en fonction de tous les points d'abscisse inférieure.

On utilise la distributivité de la convolution par rapport au minimum : pour des fonctions f, g, f', g', on a $(f \oplus g) * (f' \oplus g') = f * f' \oplus f * g' \oplus g * f' \oplus g * g'$. Une fonction f affine par morceaux peut se décomposer en fonctions *élémentaires* qui sont égales à $+\infty$ partout sauf sur un intervalle $[a, b]$, $a, b \in \mathbb{N} \cup \{+\infty\}$, sur lequel elles sont affines. La taille de la décomposition d'une fonction est égale à la taille de sa représentation.

Graphiquement, la convolution de deux segments consiste à mettre bout à bout ces segments dans l'ordre des pentes croissantes en commençant au point ayant pour abscisse la somme des abscisses des points de départ et pour ordonnée la somme des ordonnées des points de départ.

Un algorithme simple (algorithme 7.3) pour le calcul de la convolution de deux fonctions affines par morceaux est donc de calculer la convolution de chaque paire de segments affines, un élément de chaque paire étant un segment de la première fonction et l'autre de la deuxième fonction. Ensuite, calculer le minimum de toutes ces fonctions.

Le nombre de convolutions de segments est $k\ell$, si les fonctions ont des représentations respectives de taille k et ℓ. Il faut donc calculer le minimum de $k\ell$ fonctions de taille 3. La complexité de ce calcul serait donc $O((k\ell)^2)$. On peut réduire cette complexité en remarquant que certaines convolutions de segments ont des supports disjoints. En effet, si l'on prend k segments adjacents

Algorithme 7.3 : Convolution de deux fonctions

Données : $f_1 := [(x_1^1, y_1^1, \rho_1^1), \ldots, (x_k^1, y_k^1, \rho_k^1)]$ et $f_2 := [(x_1^2, y_1^2, \rho_1^2), \ldots, (x_\ell^2, y_\ell^2, \rho_\ell^2)]$.
Résultat : $f := [(x_1, y_1, \rho_1), \ldots, (x_n, y_n, \rho_n)]$ telle que $f = f_1 * f_2$.
début
  $f \leftarrow [\,]$;
  pour chaque (x_i^1, y_i^1, ρ_i^1) *du support de* f_1 **faire**
    $x_1' \leftarrow x_i^1$;
    pour chaque (x_j^2, y_j^2, ρ_j^2) *du support de* f_2 **faire**
      $x_2' \leftarrow x_i^2$;
      si $\rho_i^1 \leq \rho_j^2$ **alors**
        $f' \leftarrow$
        $[(x_i^1 + x_j^2, y_i^1 + y_j^2, \rho_i^1), (x_1' + x_j^2, y_i^1 + (x_1' - x_i^1)\rho_i^1 + y_j^2, \rho_j^2), (x_1' + x_2' + 1, \infty, \infty)]$;
      sinon
        $f' \leftarrow$
        $[(x_i^1 + x_j^2, y_i^1 + y_j^2, \rho_j^2), (x_2' + x_i^1, y_j^2 + (x_2' - x_j^2)\rho_j^2 + y_i^1, \rho_1^i), (x_1' + x_2' + 1, \infty, \infty)]$;
      finsi
      $f \leftarrow \min(f, f')$;
    finprch
  finprch
fin

de la première fonction et k autres segments adjacents de la deuxième fonction, la convolution des i-ièmes segments a un support disjoint de la convolution des j-ièmes segments dès que $i \neq j$. On peut alors se ramener au calcul du minimum de $k + \ell - 1$ fonctions, de taille au plus $2(k + \ell)$. La complexité devient donc $O((k + \ell)^3)$.

On peut améliorer encore cette méthode en ne décomposant plus les fonctions en parties affines, mais en parties convexes. En effet, la convolution de deux fonctions convexes affines par morceaux consiste à mettre bout à bout les segments dans l'ordre croissant des pentes. Comme les fonctions convexes ont déjà leurs segments dans l'ordre croissant, ordonner les segments se fait en temps linéaire.

7.3.2 Représentation des fonctions ultimement périodiques

Nous nous intéressons maintenant aux fonctions ultimement périodiques. On peut facilement modifier la représentation des fonctions affines par morceaux pour avoir une représentation des fonctions ultimement périodiques en ajoutant un triplet (T, p, c) qui indique à partir de quel rang T la fonction est pseudo-périodique, quelle est la période p et quel est l'incrément c. La représentation affine sert alors à représenter la fonction sur $[0, T + d]$. La représentation de f est notée $f := ((T, p, c), [(x_1, y_1, \rho_1), \ldots, (x_k, y_k, \rho_k)])$.

Grâce au lemme 7.17, on sait calculer les rangs à partir desquels l'addition, le minimum et la convolution de deux fonctions sont pseudo-périodiques, ainsi que la période et l'incrément. Il suffit alors, pour calculer l'addition, le minimum ou la convolution de deux fonctions, de calculer d'abord le rang à partir duquel le résultat est pseudo-périodique, la période et l'incrément. Ensuite, il suffit de faire le calcul comme si les fonctions étaient affines par morceaux. Pour cela, il faudra prolonger les fonctions, ce que l'on peut faire par l'algorithme 7.4 par exemple.

Nous nous intéressons maintenant à la dernière opération étudiée, la clôture sous-additive. Si

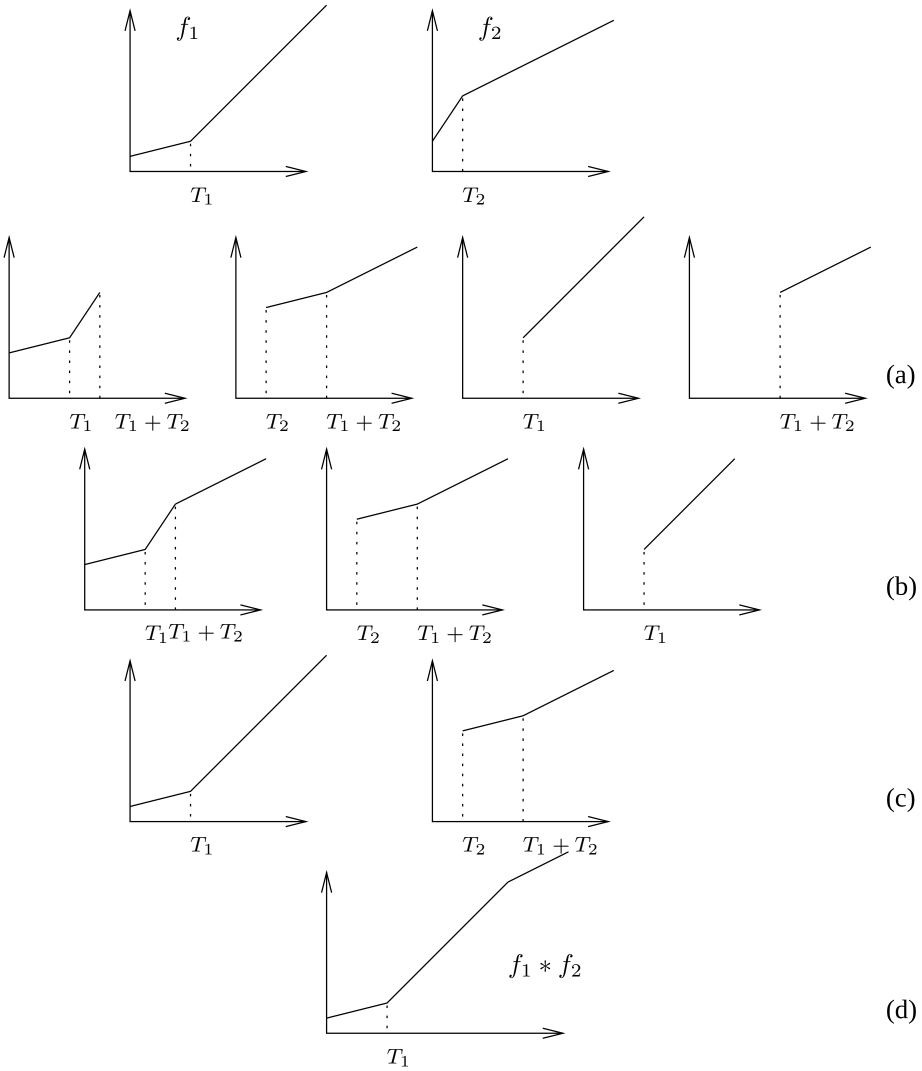


FIG. 7.11 – Convolution de deux fonctions : (a) convolution des segments deux à deux. Quatre calculs sont nécessaires, et on fait le minimum de quatre fonctions ; (b) on calcule en même temps les convolutions des segments qui ont des supports disjoints : les première et dernière fonctions de (a) sont superposées ; (c) f_1 est convexe, et donc on calcule la convolution de f_1 avec chacun des segments de f_2 : on ne calcule plus que le minimum de deux fonctions. (d) convolution de f_1 et de f_2.

Algorithme 7.4 : Prolongement

Données : $f := ((T,p,c),[(x_1,y_1,\rho_1),\dots,(x_k,y_k,\rho_k)])$, $T_1 \in \mathbb{N}$
Résultat : $f' := ((T',p,c),[(x'_1,y'_1,\rho'_1),\dots,(x'_\ell,y'_\ell,\rho'_\ell)])$, $T'-p < T_1 \leq T'$ et $f = f'$
début
  $n \leftarrow \lceil \frac{T_1 - T'}{d} \rceil$; $T' \leftarrow dn$;
  Calculer $f(T)$;
  Insérer $(T, f(T), \rho_k)$ dans la représentation de f;
  pour chaque *i de 1 à n* **faire**
    $f \leftarrow f :: [(T+id, f(T)+ic, \rho),\dots,(x_k+id, y_k+ic, \rho_k)])]$;
  finprch
  $T \leftarrow T'$;
  retourner *f;*
fin

l'on utilise l'étude faite précédemment, on sait calculer un rang à partir duquel la fonction est pseudo-périodique, mais avant ce rang, on n'a montré aucun moyen de faire les calculs, si ce n'est point par point, ce qui n'est pas satisfaisant. Comme dans le cas de la convolution, nous allons décomposer les fonctions pour simplifier les calculs et se ramener à des opérations connues.

Clôture sous-additive d'une fonction ultimement pseudo-périodique

L'idée est à nouveau de décomposer f en fonctions élémentaires f_i telles que $f = \min_i f_i$. On a alors $f^* = \bigstar_i f_i^*$, et il suffit de savoir calculer les f_i^*. On distingue deux types de fonctions élémentaires : les *segments*, comme dans le cas du calcul de la convolution, et les *segments répétés*, c'est-à-dire, il existe a, T, p, c, ρ, tel que la fonction f est définie sur les intervalles $[T+ip, T+ip+a], i \in \mathbb{N}$ (on suppose que $a \leq p$), et que sur un intervalle, $[T+ip, T+ip+a]$, f est un segment de pente ρ avec $f(T+ip) = f(T)+ic$.

Lemme 7.25 *Si f est une fonction élémentaire telle que*

$$f(x) = \begin{cases} \infty & si\ x \notin [a,b] \\ c + \rho(x-a) & sinon, \end{cases}$$

alors f^i est définie sur $[ia, ib]$ et

$$f^i(x) = \begin{cases} \infty & si\ x \notin [ia,ib] \\ ic + \rho(x-ia) & sinon. \end{cases}$$

Démonstration. Il suffit d'appliquer le lemme 7.13 à la fonction f et de procéder par récurrence. □

Lemme 7.26 *Si f est une fonction élémentaire telle que*

$$f(x) = \begin{cases} \infty & si\ x \notin [a,b] \\ c + \rho(x-a) & sinon, \end{cases}$$

alors :

si $f(a)/a \leq f(b)/b$, f^* est ultimement pseudo-périodique de période a à partir du rang $a\lceil a/(b-a)\rceil$, d'incrément $f(a)$. Sur la période $[ia,(i+1)a[$, on a $f^*(ia+x) = ica + \rho x$. Sur l'intervalle $[ia,(i+1)a[$, $i < \lceil a/(b-a)\rceil$, $f^* = f^i$.

Si $f(a)/a > f(b)/b$. Alors f^* est ultimement pseudo-périodique de période b à partir du rang $b\lceil b/(b-a)\rceil$, d'incrément $f(b)$. Sur la période $](i-1)b, ib]$, on a $f^*(ib-x) = idb - \rho x$. Sur l'intervalle $[(i-1)b, ib[$, $i < \lceil b/(b-a)\rceil$, $f^* = f^i$.

Démonstration. On suppose que $f(a)/a \leq f(b)/b$. Ceci est équivalent à $c/a \leq \rho$. L'autre cas s'en déduit facilement. On a $f^* = \min f^i$. Or, si $i < \lceil a/(b-a)\rceil$, le support de f^i est disjoint des supports des f^j, $j \neq i$ d'après le lemme précédent. Donc sur l'intervalle $[ia,(i+1)a[$, $f^* = f^i$. Si $i \geq \lceil a/(b-a)\rceil$, $f^{i+1}((i+1)a) = (i+1)c \geq ia + \rho a = f^i((i+1)a)$. Donc, $\min_j f^j(ia) = f^i(ia)$. Comme pour tout $j \in \mathbb{N}$ où les fonctions f^j sont définies, on a $f^j(ia+x) = f^j(ia) + \rho x$, on a bien $f^*(ia+x) = f^i(a) + \rho x$ si $x < a$. □

Ce lemme prouve que l'on peut facilement calculer les clôtures sous-additives des fonctions élémentaires qui sont des segments. La taille de la représentation d'une telle fonction est $\lceil a/(b-a)\rceil + 1$. Notons que si $c/a = \rho$, alors la clôture sous additive est affine par morceaux. Intéressons-nous maintenant aux segments répétés.

Lemme 7.27 *Soit f une fonction telle qu'il existe T, p, a, c, ρ tels que f est définie sur $[T + ip, T + ip + a]$, $i \in \mathbb{N}$, et que pour $x \in [0, a]$, $f(T + ip + x) = f(T) + ic + \rho x$. Alors on peut calculer f^* en un temps fini.*

Démonstration. On étudie tout d'abord f^k. La fonction f^k est définie sur les intervalles $[kT + ip, kT + ip + ka]$, $i \in \mathbb{N}$. Si $ka < p$, alors ces intervalles sont disjoints, et pour $x \in [0, ka]$, $f^k(kT+ip+x) = kf(T)+ic+\rho x$. En effet, la seule manière de décomposer $kT+ip+x$ en k une somme de k éléments du support de f est $T+i_1p+x_1, \dots, T+i_kp+x_k$, avec $i_1+\dots+i_k = i$ et $x_1+\dots+x_k = x$, car $ka < p$. On a alors $f(T+i_1p+x_1)+\dots+f(T+i_kp+x_k) = kf(T)+ic+\rho x$ quelle que soit la décomposition.

Si $ka \geq p$, alors les intervalles $[kT+ip, kT+ip+ka]$, $i \in \mathbb{N}$ se recoupent et il faut comparer ρ et c/p.

Si $c/p \leq \rho$: fixons x appartenant au support de f^k. Soient $i_1, \dots, i_k$ et $x_1, \dots, x_k \in [0, a[$ tels que $(\sum_{j\in\mathbb{N}_k} i_j)p + (\sum_{j\in\mathbb{N}_k} x_j) = x$. Alors $f(T+i_1p+x_1) + \dots + f(T+i_kp+x_k) = kf(T) + (\sum_{j\in\mathbb{N}_k} i_j)c + (\sum_{j\in\mathbb{N}_k} x_j)\rho$. Comme $c/p \leq \rho$, pour trouver une décomposition de x qui minimise cette expression, il vaut mieux maximiser $(\sum_{j\in\mathbb{N}_k} i_j)p$ par rapport à $(\sum_{j\in\mathbb{N}_k} x_j)$. Soit x' le reste de la division de $x - kT$ par p. On choisit donc $x_1, \dots, x_k$ tels que $\sum_{j\in\mathbb{N}_k} x_j = x$, et on a $f^k(x) = kf(T) + \lfloor (x-kT)/p \rfloor c + x'\rho$. La fonction f^k est donc ultimement pseudo-périodique à partir de T, de période p et d'incrément c.

Si $\rho < c/p$: un raisonnement similaire montre qu'il faut maintenant maximiser $(\sum_{j\in\mathbb{N}_k} x_j) = x$ par rapport à $(\sum_{j\in\mathbb{N}_k} i_j)p$. Donc, sur l'intervalle $[kT + ip + ka, kT + (i+1)p + ka[$, on a $f^k(kT + ip + x) = kf(T) + ic + \rho x$. Par conséquent, f^k est ultimement pseudo-périodique de période p d'incrément c à partir du rang $kT + ka$. Notons que sur l'intervalle $[kT, kT + ka]$, $f^k(x) = kf(T) + \rho(x - kT)$.

Nous nous intéressons maintenant à la clôture sous-additive de f, qui est le minimum de toutes les fonctions f^i que nous venons de calculer. Remarquons tout d'abord que l'incrément et la période de f^k ne dépendent pas de k. De plus (quitte à remplacer T par $t + a$ et a par $-a$

dans le deuxième cas), les intervalles sur lesquels les fonctions f^k sont affines ne dépendent pas de a. Les fonctions f^k ont donc le même comportement, décalé.

Il nous faut à nouveau regarder plusieurs cas :

Si $f(T)/T \leq \min(\rho, c/p)$: alors $f(kT) \geq kf(T)$ et $f^*(kT) = f^k(kT) = kf(T)$. f^* est donc ultimement pseudo-périodique à partir de kT, $k = \min\{j \in \mathbb{N} : ja > p\}$. Sur l'intervalle $[kT, (k+1)T[$, $f^* = f^k$.

Si $f(T)/T \geq \max(\rho, c/p)$: alors $f(kT) \leq kf(T)$ et $f^*(kT) = f(kT)$. Mais alors, pour tout k_1, k_2 tel que f^{k_i} sont définies entièrement à partir d'un certain rang, $f^{k_1} \leq f^{k_2}$ si $k_1 \leq k_2$ sur leur domaine commun de définition. Ainsi, $f^*(x) = f^j(x)$, où $j = \min\{k \in \mathbb{N} \mid f^k(x) \neq +\infty\}$. La fonction f^* est donc ultimement pseudo-périodique, de période p, d'incrément c à partir de $\lceil p/a \rceil T$.

Si $\rho \leq f(T)/T \leq c/p$: par raisonnement similaire au cas $\rho < c/p$, on a à partir d'un certain rang, $f^*(x) = f^k(x)$, pour $x \in [(k-1)(T+a), k(T+a)]$.

Si $c/p \leq f(T)/T \leq \rho$: Il existe $k \in \mathbb{N}$ à partir duquel $f^k(x) > f(x)$, et f^k est définie sur $[kT, \infty[$. Donc, il suffit de calculer un nombre fini de convolutions. □

Cette preuve donne un algorithme de calcul de la clôture sous-additive dans le cas général : on peut trouver f^* en décomposant d'abord f en fonctions élémentaires, puis en calculant la clôture sous-additive de ces fonctions élémentaires pour enfin calculer la convolution de ces clôtures sous-additives.

La figure 7.12 illustre le calcul de la clôture sous additive d'une fonction affine par morceaux, donc le résultat est une fonction ultimement pseudo-périodique.

On peut améliorer la complexité des calculs en décomposant les fonctions en fonctions concaves, car les calculs de clôture sous-additive sont assez simples. À titre d'exemple, calculons la clôture sous-additive d'une fonction convexe définie sur $[0, T]$.

Soit f une fonction concave sur $[0, T]$ et infinie sur $[T+1, \infty[$. Deux cas se présentent :

1. $f(1) \leq f(T)/T$: $f^*(t) = f(1)t$, $\forall t \in \mathbb{N}$;
2. $f > f(T)/T$: f^* est pseudo-périodique de période T et d'incrément $f(T)$.

Lemme 7.28 *$\forall (n, t) \neq (0, 0)$, $f^*(nT + i) = nf(T) + f(i)$.*

Démonstration. f est concave, donc c'est vrai sur $[1, T]$. Pour $t \in [T+1, 2T]$, on a :

$$f^*(T+1) = \min_{0<s<T+1} f^*(s) + f^*(T+1-s) \tag{7.16}$$
$$= \min_{0\leq u\leq T} f(1+u) + f(T-u) \tag{7.17}$$
$$= f(1) + f(T), \tag{7.18}$$

car f est concave sur $[0, T]$, donc les pentes de f sont décroissantes. Le même principe s'applique pour $t \in [T+1, 2T]$. Si $t \in \mathbb{N}$ et $t = nT + j$, $T+1 \leq j \leq 2T$, alors on a

$$f^*(t) = \min_{0<s<t} f^*(s) + f^*(t-s) \tag{7.19}$$
$$= nf(T) + \min_{0<s<j} f^*(s) + f^*(j-s) \tag{7.20}$$
$$= nf(T) + f^*(j). \tag{7.21}$$

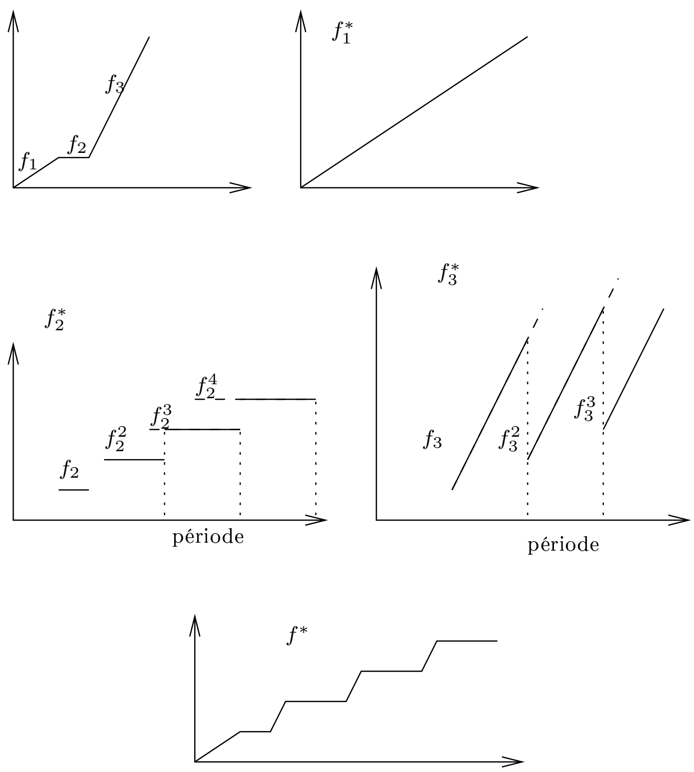


FIG. 7.12 – Calcul de la clôture sous-additive.

Ceci permet de conclure. □

*

* *

Dans ce chapitre, nous avons étudié la stabilité de certaines fonctions par les opérations du *network calculus* et donné des algorithmes en vue de l'implémentation de ces opérations. Dans la première partie, nous avons mis en évidence l'importance de trouver une bonne classe de fonctions à étudier, ainsi que l'importance des espaces de départ et d'arrivée de ces fonctions. Nous avons ainsi des résultats de stabilité pour les fonctions définies sur $\mathbb{Q}_+$ et sur $\mathbb{N}$, alors que des exemples montrent que les fonctions définies sur $\mathbb{R}$ ne sont par stables. Dans la deuxième partie, nous montrons que trouver des algorithmes efficaces pour le calcul des opérations nécessite à nouveau une analyse fine des opérations, et donnons des pistes en vue de nouvelles améliorations (étude des fonctions convexes ou concaves).

Chapitre 8

Interdire des angles pour casser les cycles

L'optimisation du routage dans les réseaux est un problème fondamental pour l'utilisation des performances des réseaux. Dans la partie II, nous avons étudié les performances de systèmes qui ne pouvaient pas se bloquer, mais ce n'est pas toujours le cas. En particulier, dans les routages de type « wormhole », qui sont largement répandus et étudiés dans [75, 45], les routages peuvent mener à des blocages de plusieurs types : soit aucun paquet ne peut plus être transmis, car toutes les ressources sont réservées de manière cyclique et aucune ressource libre ne permet d'amorcer la transmission, soit des paquets sont transmis indéfiniment dans le réseau sans jamais atteindre leur destination. Dans toutes ces situations, c'est l'existence de cycles dans les réseaux qui pose problème.

Pour résoudre ce problème et éliminer tous les blocages, plusieurs solutions existent : pour relier deux nœuds du réseau, on peut attribuer à chaque paire de nœuds un chemin prédéterminé qui sera toujours utilisé pour relier ces nœuds, c'est ce que l'on appelle un routage *déterministe*. Mais les performances du réseau se trouvent alors fortement réduite, puisqu'un unique itinéraire est autorisé. Une deuxième solution serait de mettre des priorités ou des préemptions sur les paquets. Mais alors le routage n'est pas équitable, et le temps de transmission de certains paquets pourraient être indéfiniment long si des paquets ayant une priorité plus forte arrivent en masse.

Une troisième solution est d'interdire certaines routes, de telle sorte qu'aucun chemin ne puisse suivre un cycle en entier. De cette manière, non seulement aucun paquet ne peut être transmis indéfiniment de nœuds en nœuds, mais on peut aussi éliminer les blocages. Au lieu d'interdire explicitement des chemins dans le réseau, on interdit des *angles*, c'est-à-dire, pour chaque nœud, on interdit certaines destinations (directes) aux paquets selon leur provenance. C'est le modèle *d'interdiction d'angles* (ou *turn prohibition*), et on cherche à « casser » tous les cycles. Il a été introduit par Glass et Ni dans [54], où le problème est étudié pour des topologies régulières, les grilles et les tores. La question se pose alors de savoir quel est le meilleur ensemble d'angles qu'il faut interdire, pour garder de bonnes performances. Intuitivement, plus on élimine d'angles, plus la longueur moyenne des chemins d'un point à un autre est grande et plus les performances sont mauvaises. Il est clair que l'interdiction d'un ensemble d'angles peut déconnecter deux nœuds. Le problème correspond alors à minimiser le nombre d'angles à interdire pour casser tous les cycles et tel que la connexité est conservée. Ce problème a récemment été étudié [87, 32, 43] et des heuristiques développées. Mais aucune méthode exacte efficace n'existe.

Dans ce chapitre, nous essayons donc de trouver des méthodes plus efficaces pour trouver un ensemble minimal d'angles à interdire. Dans une première partie, nous définissons plus précisément le concept d'angle interdit et posons le problème. Nous y répondons pour le cas des graphes orientés en montrant que minimiser le nombre d'angles pour casser les cycles est un problème NP-complet. Dans une deuxième partie, nous présentons certaines heuristiques, dont plusieurs ont été développées dans [87, 32] en tentant une analyse plus précise que celle qui y est développée. Enfin, nous étudions le problème pour certaines classes de graphes. Nous résolvons le problème pour le cas des graphes triangulés et des graphes 3-réguliers.

8.1 Interdire les angles

8.1.1 Définitions

Soit $G = (V, E)$ un graphe non orienté connexe. On ne considèrera par la suite que des graphes finis. On appelle *chemin* une suite de sommets $v_1 v_2 \dots v_n$ tels que pour tout $i \in \{1, n\}$, $(v_{i-1}, v_i) \in E$. On appelle *cycle* un chemin dont la première et la dernière arête sont la même, c'est-à-dire un chemin de la forme $v_0 v_1 \dots v_{n-1} v_0 v_1$ avec $v_i \neq v_{(i+2) \mod n}$ (on ne revient pas en arrière). Cette notion est légèrement différente de la notion classique de cycle (*i.e.* un chemin dont le premier sommet et le dernier sommet sont le même et qui ne passe pas deux fois par la même arête). Pour différencier ces deux notions, on appellera cette dernière cycle de sommets.

Définition 8.1 *Un* angle *est un triplet $(a, b, c) \in V^3$ de sommets distincts tel que $(a, b) \in E$ et $(b, c) \in E$. On dit que (a, b, c) est un angle autour de b. On note $\mathbb{A}$ l'ensemble des angles du graphe.*

Puisque le graphe est non orienté, si $(a, b, c) \in \mathbb{A}$, alors $(c, b, a) \in \mathbb{A}$. On considérera toujours que ces deux angles ne sont en fait qu'un seul et même angle. Soit $a \in V$ et d_a son degré. Le nombre d'angles autour de a est alors $d_a(d_a - 1)/2$.

On veut interdire des angles afin de casser les cycles du graphe : si un angle (a, b, c) est interdit, alors les chemins ne peuvent contenir le facteur abc.

Définition 8.2 *On appelle* chemin admissible *un chemin qui ne contient pas d'angle interdit comme facteur.*

Définition 8.3 *Soit $\mathbb{AI} \subset \mathbb{A}$ un sous-ensemble d'angles. Si en interdisant les angles de $\mathbb{AI}$, il n'y a plus de cycle admissible, alors on dit que $\mathbb{AI}$ casse les cycles.*

Définition 8.4 *Soit $\mathbb{AI} \subset \mathbb{A}$ un ensemble d'angles interdits. Si pour tout $i, j \in V$, $i \neq j$, il existe un chemin admissible de i à j, alors on dit que $\mathbb{AI}$ garde la connexité.*

On cherche des ensembles d'angles interdits de cardinal minimal qui cassent les cycles de G. La connexité est une propriété difficile à préserver dans ce problème d'optimisation. Nous nous sommes donc intéressés au cas sans cette contrainte de connexité.

Exemple 8.5 Soit G le graphe représenté en figure 8.1. La figure 8.2 représente le graphe et un ensemble d'angles qui casse les cycles et garde la connexité. La figure 8.3 représente le graphe et un ensemble d'angles qui casse les cycles, mais ne garde pas la connexité.

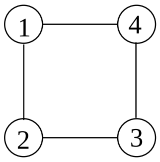


FIG. 8.1 – Exemple de graphe.

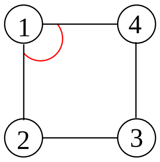


FIG. 8.2 – Exemple d'angles interdits cassant les cycles et gardant la connexité. Les angles interdits sont repérés par un arc de cercle. La graphe est consitué d'un cycle, et interdire l'angle (2,1,4) casse ce cycle.

8.1.2 Remarques sur les ensembles d'angles interdits

On verra par la suite, l'ensemble des heuristiques existantes ont une approche basée commune sur une numérotation des sommets. Rien ne dit que cette approche soit optimale. Nous avons examiné une approche basée sur la numérotation des arêtes, que nous formalisons par le lemme suivant. Il exprimerait une notion d'ordre sur les arêtes.

Lemme 8.6 *Soit G un graphe. Soit $\mathbb{AI}$ un ensemble non vide d'angles de G qui casse tous les cycles de G. Alors, il existe une arête (a, b) dont tous les angles autour de a de type (c, a, b) sont dans $\mathbb{A}$. Si de plus interdire $\mathbb{A}$ ne déconnecte pas le graphe, alors $G \setminus \{(a, b)\}$ est connexe.*

Démonstration. S'il n'existait pas de telle arête, on pourrait construire un chemin admissible infini. Comme $\mathbb{A}$ casse tous les cycles et que G est fini, c'est absurde.

Si interdire $\mathbb{A}$ ne déconnecte pas le graphe, tout couple de sommet est connecté par un chemin admissible. Un chemin admissible ne contient pas (a, b). □

Remarques sur le cas des graphes orientés

Dans ce chapitre, on considère principalement des graphes non orientés : on suppose que lorsqu'un lien existe entre deux nœuds d'un réseau, il existe dans les deux sens, ce qui est généralement le cas. Toutefois, on peut faire quelques remarques sur le problème d'interdiction d'angles dans les graphes orientés. Tout d'abord, trouver le nombre minimal d'angles à interdire

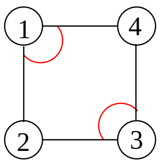


FIG. 8.3 – Exemple d'angles interdits cassant les cycles mais perdant la connexité. On interdit les angles (2,1,4) et (2,3,4). Alors, il n'y a plus de chemin admissible entre 2 et 4.

pour casser les circuits est un problème difficile dans le cas des graphes orientés, alors que la complexité dand le cas non orienté reste un problème ouvert.

Théorème 8.7 *Soit $G = (N, A)$ un graphe orienté et k un entier. Trouver s'il existe un ensemble d'angles qui casse les circuits de cardinal inférieur ou égal à k est NP-complet.*

Démonstration. On peut réduire le problème *Minimum feedback arc set* ([47], GT8), qui est NP-complet, à ce problème. Le problème *Minimum feedback arc set* est de trouver si c'est possible, étant donné un graphe orienté et un entier k, un ensemble de k arcs tel que si on élimine ces arcs, le graphe restant est sans circuit.

Soit $G = (N, A)$ un graphe orienté. On transforme ce graphe en un nouveau graphe G' de la manière suivante : pour tout arc $(x, y) \in A$, on crée un nouveau nœud n_{xy} tel que $x \to n_{xy} \to y$.

Soit $\mathcal{A}$ un ensemble d'arcs de G de cardinal inférieur ou égal à k qui casse tous les circuits de G. Alors $\mathbb{AI} = \{(a, n_{ab}, b) \mid (a, b) \in \mathbb{A}\}$ est clairement un ensemble d'angles de cardinal inférieur ou égal à k qui casse tous les circuits de G'.

Inversement, soit $\mathbb{AI}$ un ensemble d'angles de cardinal inférieur ou égal à k qui casse les circuits de G'. Si $(n_{ab}, b, n_{bc}) \in \mathbb{AI}$ est un angle autour d'un ancien nœud b, il peut être remplacé par l'angle (b, n_{bc}, c), autour du nouveau nœud n_{bc}. En effet, comme ce nœud est de degré 2, il casse les circuits contenant b et n_{bc} (ces circuits contiennent aussi nécessairement c). On transforme ainsi $\mathbb{AI}$ en un ensemble $\mathbb{AI}'$ d'au plus k angles de la forme (b, n_{bc}, c). L'emsemble des arcs $\mathcal{A} = \{(b, c) \mid (b, n_{bc}, c) \in \mathbb{AI}\}$ de cardinal inférieur ou égal à k casse les circuits de G.

La transformation de G en G' est polynomiale. Par conséquent, un algorithme polynomial pour décider si l'on peut casser les cycles d'un graphe en interdisant au plus k angles permettrait de décider polynomialement si on peut casser les circuits d'un graphe en enlevant au plus k arcs. □

La dernière remarque de ce paragraphe concerne concerne la connexité dans le cas orienté. En effet, si, comme on pourra le voir par la suite, dans le cas non orienté il existe toujours un ensemble d'angles cassant les cycles et maintenant la connexité, dans le cas orienté ce n'est pas toujours le cas.

L'exemple le plus simple est représenté en figure 8.4. Là, les nœuds 2 et 1 ne sont plus connectés.

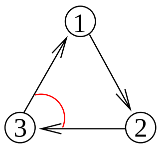


Fig. 8.4 – Casser les cycles casse la forte connexité.

Mais ce n'est pas le cas pour tous les graphes. En figure 8.5 on a représenté un ensemble d'angles qui permet de garder la forte connexité.

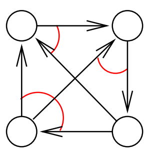


Fig. 8.5 – La forte connexité peut aussi être conservée.

Il semble délicat de trouver une caractérisation des graphes pour lesquels on peut conserver la connexité.

8.2 Étude de quelques heuristiques

Nous nous replaçons dans le cas non orienté. Dans cette partie, nous étudions quelques heuristiques données dans [87] et [32] pour calculer un ensemble d'angles à interdire pour casser les cycles. Nous en généralisons quelques-unes.

8.2.1 Algorithme de l'arbre couvrant

L'algorithme 8.1 dit de l'arbre couvrant (*spanning tree algorithm*) consiste à construire un arbre couvrant et à n'autoriser que les angles qui font partie de l'arbre. Ainsi, on casse tous les cycles et la connexité est conservée.

Algorithme 8.1 : Algorithme de l'arbre couvrant

Données : Graphe $G = (V, E)$
Résultat : $\mathbb{AI}$, ensemble d'angles qui casse les cycles et garde la connexité.
début
  Construire un arbre couvrant $T = (V, E_T)$ de G;
  $\mathbb{AI} = \{(a, b, c) \mid (a, b) \text{ ou } (b, c) \notin E_T\}$;
fin

Cet algorithme a une complexité en temps $O(|\mathbb{AI}| + |V|^2)$. Construire l'arbre couvrant peut se faire en temps linéaire (parcours du graphe en largeur ou en profondeur). Pour chaque nœud, on construit en $O(V)$ l'ensemble de ses voisins dans G (v_G), l'ensemble de ses voisins dans T (v_T) et ceux dans G mais pas dans T ($v_G - v_T$). On rajoute à $\mathbb{AI}$ $v_G \times (v_G - v_T)$.

Mais le nombre d'angles à interdire dépend de l'arbre couvrant choisi. Dans ce paragraphe, nous montrons que choisir l'arbre de manière à minimiser le nombre d'angles interdits est un problème NP-complet.

Exemple 8.8 Considérons le graphe de la figure 8.6.

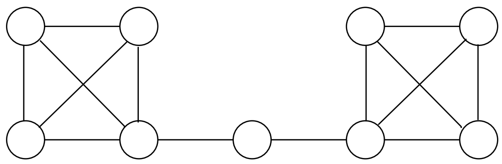


FIG. 8.6 – Graphe de l'exemple 8.8

On construit l'arbre couvrant en faisant un parcours en largeur à partir du sommet 1. Les angles interdits sont alors représentés en figure 8.7. Les arêtes de l'arbre sont représentées en traits pleins et les autres en pointillé. Le nombre d'angles interdits est 20, bien supérieur à 8, le minimum pour ce graphe.

Minimiser le nombre d'angles interdits est équivalent à maximiser le nombre d'angles à autoriser. Or, ce nombre est exactement $\sum_{x \in V} d_{T,x}(d_{T,x} - 1)/2$, où $d_{T,x}$ est le degré de x dans T.

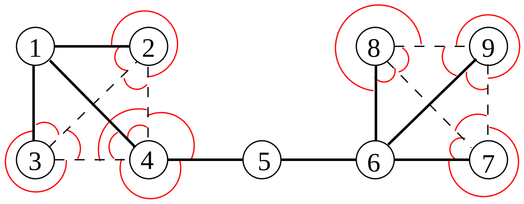


FIG. 8.7 – Angles interdits avec l'algorithme de l'arbre couvrant

Or, pour un arbre $T = (V, E_T)$, il y a $|V| - 1$ arêtes. Par conséquent, $\sum_{x \in V} d_{T,x} = 2|V| - 2$ et ainsi on veut maximiser $\sum_{x \in V} d_{T,x}^2$.

Un algorithme simple (glouton) consisterait à construire itérativement l'arbre en partant de la racine. On part du sommet de degré maximum ; on rajoute ses voisins qui deviennent ses fils ; on enlève dans le graphe initial toutes les arêtes qui créeraient un cycle dans ces arbres (une des extrémités est une feuille de l'arbre déjà construit) ; on met à jour les degrés ; on continue avec l'une des feuilles de l'arbre, parmi celles qui sont de degré maximum. Pour l'exemple de la figure 8.6, cela donnerait 18 angles interdits, ce qui est le minimum avec l'algorithme de l'arbre couvrant. Mais cette manière de faire ne minimise pas toujours le nombre d'angles interdits par la méthode de l'arbre couvrant, comme le montre le graphe de la figure 8.8. L'arbre de gauche est construit avec l'algorithme glouton. Le sommet du centre est de degré maximal, 8. On obtient alors $6*8 = 48$ angles interdits. Avec l'arbre de droite, on obtient $4*6+22 = 46$ angles interdit. En augmentant le degré du sommet central et de ses voisins, la différence entre le nombre d'angles à interdire par l'algorithme glouton et par l'arbre couvrant optimal peut être arbitrairement grande.

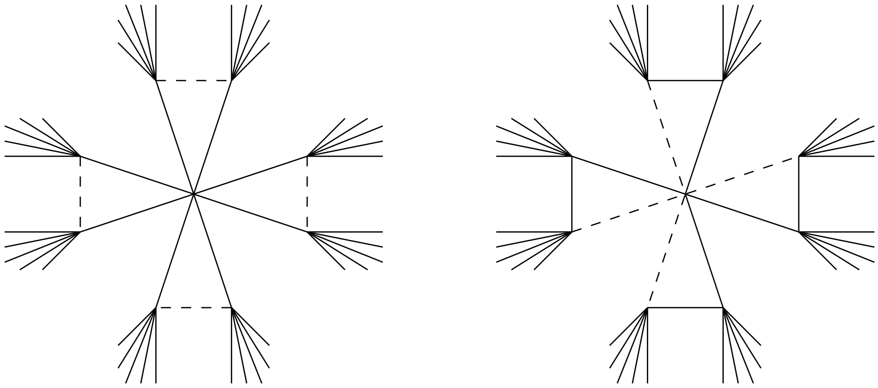


FIG. 8.8 – L'algorithme glouton pour l'arbre couvrant n'est pas optimal.

Nous montrons maintenant le théorème de NP-complétude pour le problème de trouver l'arbre couvrant qui minimise le nombre d'angles interdits.

Théorème 8.9 *Trouver l'arbre couvrant qui minimise le nombre d'angles interdits (c'est-à-dire la somme des carrés des degrés dans l'arbre) est NP-complet.*

Démonstration. On prouve par une réduction polynomiale du problème NP-complet *exact cover by 3-sets* (X3C) ([47], SP2) vers ce problème. Le problème X3C consiste, étant donné un ensemble S à $3q$ éléments et C une collection de sous-ensembles de trois éléments de S, à trouver, s'il existe, un sous-ensemble de C qui forme une partition de S.

Soit S un ensemble à $3q$ éléments, et C une collection de sous-ensembles de S à 3 éléments. On construit alors un graphe $G = (V, E)$ de la manière suivante.

L'ensemble V est constitué d'un sommet x et de trois ensembles, X_C, X_S et A. L'ensemble X_C contient $|C|$ sommets étiquetés par les éléments de C, l'ensemble X_S contient $|S|$ sommets étiquetés par les éléments de S et A sera défini ultérieurement. Les éléments de A sont de degré 1 et donc seulement reliés à x. Un sommet de X_C représente un élément de C, c'est-à-dire un sous-ensemble à trois éléments de S et chaque élément de X_C est relié aux éléments de X_S correspondants. Un exemple d'un tel graphe est montré figure 8.9. Pour tout arbre couvrant, la somme des carrés des degrés pour X_C et X_S est forcément inférieure ou égale à $\alpha = 4^2|X_C| + |X_C|^2|X_S|$. Il suffit donc de prendre $A = 2\alpha$ pour assurer que toutes les arêtes autour de x sont prises (s'il en manque une, on perd plus de 2α, ce qui ne peut être compensé par un meilleur choix autour de X_S et X_C, dont le gain serait inférieur ou égal à α).

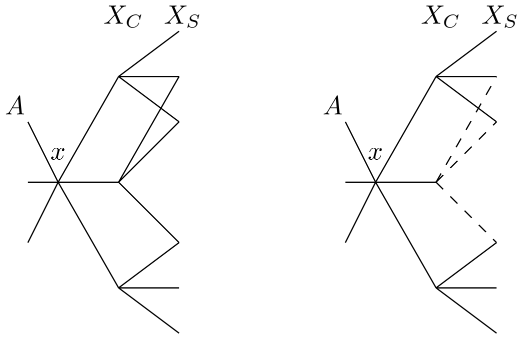


FIG. 8.9 – Réduction de X3C au problème des arbres couvrants.

Par construction du graphe, toutes les arêtes autour de x sont gardées. Les sommets de X_S ne peuvent avoir que degré 1 dans l'arbre couvrant, sinon, on crée un cycle dans l'arbre couvrant qui n'est donc plus un arbre. Il reste donc à trouver les arêtes à garder dans l'arbre couvrant entre X_C et X_S qui maximise la somme des carrés des degrés des sommets de X_C.

Les sommets de X_C dans l'arbre couvrant ont degré entre 1 et 4. Soit M_i le nombre de sommets de X_C ayant degré i dans l'arbre couvrant. On a :

(1) $M_1 + M_2 + M_3 + M_4 = |C|$,
(2) $M_2 + 2M_3 + 3M_4 = 3q$, et
(3) $M_1 + 4M_2 + 9M_3 + 16M_4$ est à maximiser.

Il faut donc aussi maximiser (3) + (1) - 5(2). Or, cela donne :

$$M_1 + 4M_2 + 9M_3 + 16M_4 + M_1 + M_2 + M_3 + M_4 - 5 * (M_2 + 2M_3 + 3M_4) = 2M_1 + 2M_4.$$

D'après (1), $2M_1 + 2M_4 \leq 2|C|$. Ce maximum est atteint si et seulement si $M_2 = M_3 = 0$, c'est-à-dire pour une solution ne contenant que des sommets de degrés 1 (non reliés à X_S) ou 4 (reliés à tous leurs voisins dans X_S). Tous les éléments de X_C devant être reliés à un élément de X_C pour appartenir à l'arbre couvrant, M_4 définit une partition de S. Ainsi trouver un arbre couvrant de G dont la somme des carrés des degrés est $2M_1 + 2M_4 + 5(2) - (1) = |C| + 15q$ est équivalent à trouver un sous-ensemble de C qui partitionne S, ce qui achève la réduction de X3C à notre problème. $\square$

8.2.2 Algorithme Up/Down

Un algorithme plus efficace a été introduit pour trouver un ensemble d'angles qui casse les cycles ([87], [32]). Cet algorithme (algorithme 8.2) est aussi basé sur une numérotation des sommets à partir d'un arbre couvrant.L'arbre couvrant permet d'assurer la connexité. Dans cette partie, nous décrivons l'algorithme Up/Down et étendons la technique de numérotation sans construction d'arbre couvrant. Nous proposons différentes numérotations pour le problème avec ou sans connexité. Les garanties que nous obtenons sont meilleures que celles obtenues avec l'algorithme Up/Down initial.

Algorithme 8.2 : Algorithme Up/Down basé sur un arbre couvrant

---

Données : Graphe $G = (V, E)$
Résultat : $\mathbb{AI}$, ensemble d'angles qui casse les cycles et garde la connexité.
début
  Construire un arbre couvrant enraciné $T = (V, T_E)$ de G;
  Ordonner totalement les sommets dans un ordre cohérent avec leur distance à la racine;
  $\mathbb{AI} \leftarrow \emptyset$;
  pour chaque *angle* (a, b, c) *tel que* $a < b$ *et* $c < b$ **faire**
    $\mathbb{AI} \leftarrow \mathbb{AI} \cup \{(a, b, c)\}$;
  finprch
fin

L'algorithme 8.2 se base sur le fait que pour tout cycle, il existe un angle (a, b, c) tel que $a < b$ et $c < b$: il suffit de considérer b le plus grand élément du cycle. La connexité est conservée car un angle défini par deux arêtes adjacentes de l'arbre couvrant est autorisé.

Théorème 8.10 ([32], Theorem 3) *L'algorithme 8.2 interdit au plus la moitié du nombre total d'angles.*

L'algorithme 8.3 présente une généralisation de l'algorithme Up/Down où la numérotation des sommets est quelconque. Une telle numérotation permet de diminuer le nombre d'angles interdits qui cassent les cycles, mais la connexité n'est plus garantie.

L'algorithme 8.3 pourrait être écrit de manière similaire à l'algorithme 8.2. La ligne concernant la construction d'un arbre serait enlevée, et l'ordre sur les sommets serait un ordre total quelconque. Nous en donnons ici une version totalement différente, mais qui donne une autre interprétation de la numérotation : tous les angles sont interdits autour du sommet le plus grand, et tous les autres angles mettant en jeu ce sommet sont autorisés. Cette remarque permet d'affirmer qu'après avoir considérer les angles où le sommet le plus grand apparaît, on peut traiter le graphe dans lequel ce sommet a été éliminé. Mais alors, la numérotation permet d'avoir un ordre pour l'élimination des sommets. C'est l'ordre inverse de la numérotation. L'algorithme 8.3 décrit cette approche. Comme la numérotation suit l'élimination des sommets, les angles interdits sont de la forme (a, x, b), où $x < a$ et $x < b$. Néanmoins, on continue d'appeler cet algorithme Up/Down.

Lemme 8.11 *L'algorithme 8.3 construit un ensemble d'angles* $\mathbb{AI}$ *qui casse tous les cycles.*

Démonstration. Quand on interdit tous les angles autour d'un sommet, on casse tous les cycles passant par ce sommet. Il ne reste alors dans le graphe que des cycles passant par les autres sommets. Cette remarque permet de montrer le lemme par récurrence. □

Algorithme 8.3 : Algorithme Up/Down généralisé

Données : Graphe $G = (V, E)$
Résultat : $\mathbb{AI}$, ensemble d'angles qui casse les cycles.
début
  $\mathbb{AI} \leftarrow \emptyset$;
  $\mathbb{AA} \leftarrow \emptyset$;
  Ordonner les sommets selon un ordre total;
  pour chaque $x \in V$ *pris dans cet ordre* **faire**
    pour chaque $(a, x, b) \in \mathbb{A} \setminus (\mathbb{AI} \cup \mathbb{AA})$ **faire**
      $\mathbb{AI} \leftarrow \mathbb{AI} \cup \{(a, x, b)\}$;
    finprch
    pour chaque $(x, a, b) \in \mathbb{A} \setminus (\mathbb{AI} \cup \mathbb{AA})$ **faire**
      $\mathbb{AA} \leftarrow \mathbb{AA} \cup \{(x, a, b)\}$;
    finprch
  finprch
fin

On peut aussi garantir la conservation de la connexité en choisissant les sommets dans un ordre $x_1 < x_2 < \cdots < x_n$ tel que pour tout $i \in \mathbb{N}_n$, $G - \{x_1, \ldots, x_i\}$ soit connexe. L'ordre dans lequel on considère les sommets est important pour minimiser le nombre d'angles interdits. On montre dans les deux prochaines propositions que l'on peut trouver un ordre sur les sommets en temps polynomial qui interdit au plus un tiers des angles du graphe, que ce soit en gardant la connexité ou non.

Proposition 8.12 *On peut choisir l'ordre des sommets dans l'algorithme 8.3 de sorte à interdire au plus un tiers du nombre total d'angles.*

Démonstration. Pour cela, il suffit de choisir les sommets dans l'ordre $x_1 < x_2 < \cdots < x_n$ de telle sorte que pour tout $i \in \mathbb{N}_n$, x_i soit un sommet de plus petit degré dans $G_{i-1} = G - \{x_1, \ldots, x_{i-1}\}$. À chaque étape, on interdit alors $I(x_i) = d_{G_{i-1}}(x_i)(d_{G_{i-1}}(x_i) - 1)/2$ angles, alors que $A(x_i) = \sum_{y \in V_{G_{i-1}}(x_i)} (d_{G_{i-1}}(y) - 1)$ angles ne seront jamais interdits. Comme pour tout $y \in V_{G_{i-1}}(x_i)$, $d_{G_{i-1}}(x_i) \leq d_{G_{i-1}}(y)$, on a $I(x_i)/(I(x_i) + A(x_i)) \leq 1/3$, au total,

$$\frac{|\mathbb{AI}|}{|\mathbb{A}|} = \frac{\sum_{i=1}^{|V|} I(x_i)}{\sum_{i=1}^{|V|} I(x_i) + A(x_i)} \leq \frac{1}{3}. \tag{8.1}$$

□

L'algorithme Up/Down généralisé exécuté avec l'ordre décrit dans cette preuve est appelé l'*algorithme glouton Up/Down*.

Proposition 8.13 *On peut choisir l'ordre des sommets dans l'algorithme 8.3 de sorte à conserver la connexité et à interdire au plus un tiers des angles .*

Démonstration. Comme pour la proposition précédente, on prouve que l'on peut trouver un sommet qui, quand il est traité, autorise au moins deux fois autant d'angles qu'il n'en interdit. On appelle *point d'articulation* un sommet dont la suppression déconnecte le graphe. Considérons les composantes connexes du graphe obtenu en enlevant tous les points d'articulations du graphe initial. Il existe une composante connexe qui est reliée à un seul point d'articulation : sinon, on aurait un cycle de composantes et de points d'articulations, ce qui n'est pas possible, puisque enlever un point d'articulation ne déconnecterait plus le graphe. Soit C une telle composante connexe du graphe. À l'intérieur de cette composante, soit x le sommet de plus petit degré et qui si possible n'est pas relié au point d'articulation. Si c'est possible, on a toujours $A(x) \geq 2I(x)$, avec les notations de la preuve précédente.

Sinon, on choisit x de plus petit degré et relié au point d'articulation. Soit m le nombre de voisins de x de même degré. Ces m voisins sont aussi reliés au point d'articulation, par choix de x. Donc, le point d'articulation a un degré au moins $2+m$. Les autres voisins de x ont degré au moins $d(x)+1$. Le nombre d'angles autorisés est alors

$$\begin{aligned}&m+2+m(d(x)-1)+(d(x)-m-1)d(x)\\&\qquad\qquad = 2+md(x)+d(x)(d(x)-1)-md(x) = 2+d(x)(d(x)-1).\end{aligned}$$

Le nombre d'angles interdits est $I(x) = d(x)(d(x)-1)/2$. On autorise donc bien au moins deux fois plus d'angles qu'on n'en interdit.

Trouver les points d'articulations peut se faire en temps linéaire en la taille du graphe, donc la preuve donne bien un algorithme polynomial pour trouver cet ordre. □

L'algorithme Up/Down généralisé exécuté avec l'ordre décrit dans cette preuve est appelé l'*algorithme glouton Up/Down avec connexité.*

Appliqué au graphe de la figure 8.6, l'algorithme glouton Up/Down donne le graphe en figure 8.10 et l'algorithme glouton Up/Down avec connexité le graphe en figure 8.11. Sur chacun de ces graphes, l'a numérotation des sommets correspond à l'ordre de leur élimination.

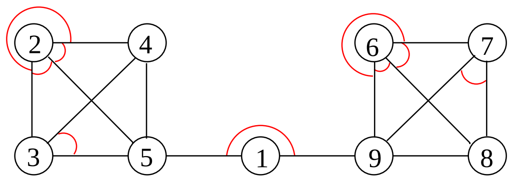


FIG. 8.10 – Angles interdits avec l'algorithme Up/Down glouton. Il y a 9 angles interdits.

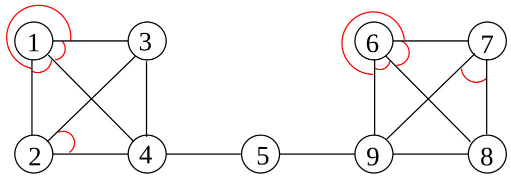


FIG. 8.11 – Angles interdits avec l'algorithme Up/Down glouton avec connexité. Il y a 8 angles interdits, ce qui est le minimum global que l'on peut obtenir pour ce graphe.

Pour cet exemple, l'algorithme qui conserve la connexité donne un meilleur résultat : moins d'angles sont interdits. De plus, le glouton qui ne déconnecte pas obtient le nombre minimal d'angles interdits pour casser tous les cycles. Le lemme suivant appliqué à l'exemple de la figure 8.10 montre que ce n'est pas toujours le cas.

Lemme 8.14 *Soit $G = (V, E)$ un graphe pour lequel l'algorithme Up/Down glouton n'est pas optimal. Alors avec le graphe $G' = (V \cup \{x\}, E \bigcup \cup_{v\in V}(v, x))$, le glouton avec connexité n'est pas optimal.*

Démonstration. Le sommet x est toujours de degré maximal. C'est donc le dernier à être retiré. Le glouton avec connexité a donc le même comportement que le glouton sans contrainte de connexité avec G'. Quel que soit l'ordre des points de V choisi, le nombre d'angles supplémentaires à interdire dans G' est exactement $|E|$. □

Exemple 8.15 La figure 8.13 illustre le lemme précédent. Le nombre d'angles minimal est donné par le glouton avec connexité, sur la figure 8.11 . Donc, le nombre d'angles optimal est donné par la figure 8.12 pour le graphe augmenté d'un sommet relié à tous les autres. Mais pour ce graphe, le glouton avec connexité donne le graphe en figure 8.13.

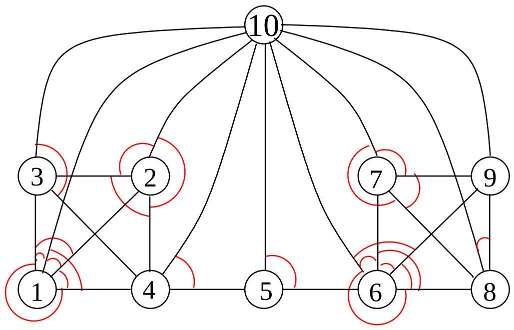


FIG. 8.12 – Up/Down optimal. Il y a 22 angles interdits.

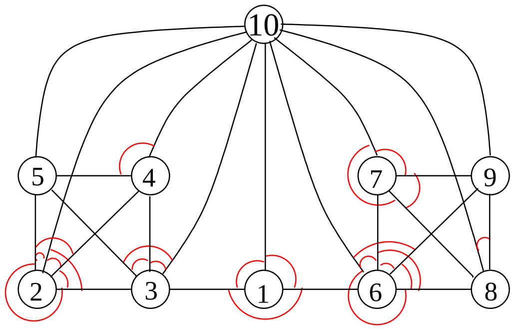


FIG. 8.13 – Up/Down glouton non optimal. Il y a 23 angles interdits.

De même, la figure 8.14 montre que l'algorithme glouton sans contrainte de connexité peut être strictement meilleur que tout algorithme glouton avec containte de connexité. Sur le premier graphe, une numérotation optimale est donnée. Celle-ci correspond à un algorithme glouton sans contrainte de connexité. Les sommets non numérotés sont enlevés en dernier, et l'ordre n'a pas d'importance : ces sommets constituent deux cliques de taille 6, et nous verrons par la suite que pour casser tous les cycles d'une clique de taille n, il faut interdire au moins C_n^3 angles. Le nombre d'angles interdits avec cette numérotation est 55 (un angle interdit pour l'élimination du premier sommet et pour chacune des deux composantes restantes, le nombre d'angles interdits est 27). Le deuxième graphe représente une numérotation optimale quand il y a une contrainte de connexité. Les nombres d'angles interdits pour les sommets de 1 à 10 sont respectivement 6, 10, 6, 3, 3, 1, 0, 0, 0, 0. Ensuite, le nombre minimal d'angles à interdire est 27, comme dans le cas précédent. En tout, il faut donc interdire 29+27=56 angles. Aucune numérotation avec contrainte de connexité n'est donc optimale.

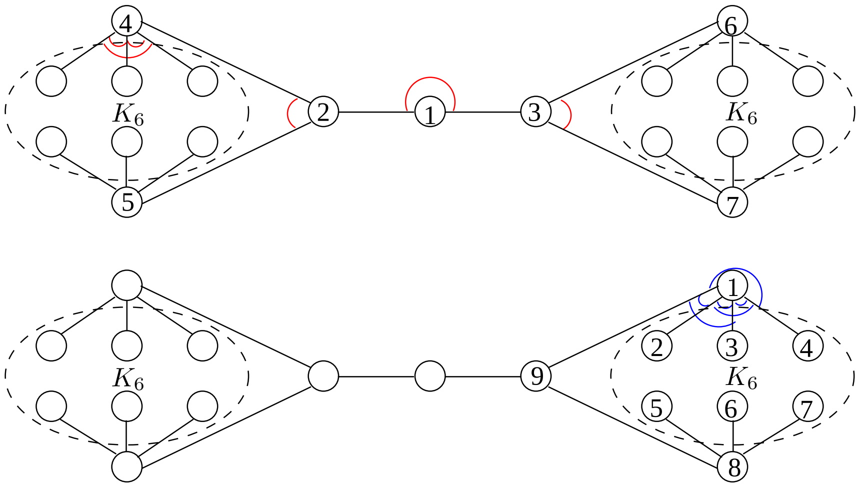


FIG. 8.14 – Up/Down glouton sans déconnecter moins bien qu'en déconnectant.

Le graphe de la figure 8.15 montre que le glouton (et donc le glouton avec connexité d'après le lemme 8.14) peut donner un nombre d'angles arbitrairement éloigné du nombre minimal d'angles interdits pour casser les cycles.

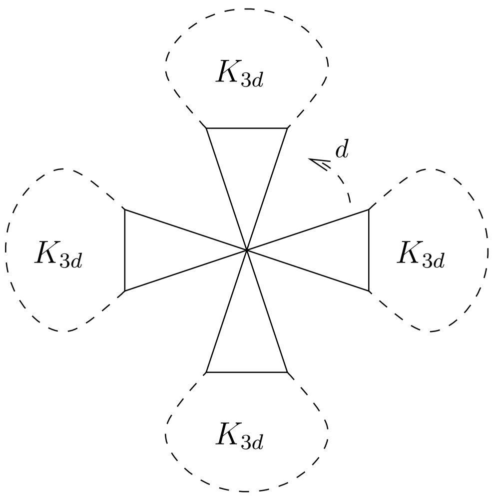


FIG. 8.15 – L'algorithme Up/Down n'est pas optimal. Le graphe est constitué de d cliques de taille $3d$, et deux sommets de chaque clique sont reliés à un sommet commun.

En effet, le nombre d'angle optimal s'obtient en éliminant d'abord les sommets des cliques. Pour chaque clique le coût est donc C_{3d}^3 (voir la proposition 8.17). Ensuite, le nombre d'angles interdits est de d. Au total, $T_{opt} = dC_{3d}^3 + d$. L'algorithme glouton élimine d'abord le sommet du centre, ce qui faut C_{2d}^2 angles interdits. Ensuite, il ne reste que les cliques. Donc, au total, $T_{glou} = dC_{3d}^3 + C_{2d}^2$. Ainsi, $T_{glou} - T_{opt} = C_{2d}^2 - d = 2d.(2d-1)/2 - d = d(2d-2)$, alors que le graphe a $3d+1$ sommets.

8.2.3 Autres algorithmes

D'autres algorithmes ont été développés entres autres par Starobinski et Karpovski pour trouver un ensemble d'angles à interdire minimal, l'algorithme TP dans [32], et l'algorithme BC dans [66] par exemple. Aucun de ces algorithmes n'est exact, et les garanties sont le mêmes que pour le Up/Down glouton : on interdit au plus un tiers des angles du graphe. Ils sont tous les deux basés principalement sur l'algorithme Up/Down : on choisit toujours le sommet de plus petit degré. Si l'enlever ne déconnecte pas le graphe, alors il est éliminé de la manière décrite dans cette partie. Sinon, une construction annexe permet d'éliminer ce point tout en autorisant quelques angles autour de ce sommet pour préserver la connexité.

Nous avons vu dans cette section que les sommets de degré minimal ne devaient pas toujours être enlevés en premier. Dans la section suivante, nous allons montrer que certains sommets peuvent être enlevés en premier pour effectivement minimiser le nombre d'angles interdits : ce sont les sommets simpliciaux. Nous allons aussi montrer que l'on peut trouver un algorithme Up/Down optimal en temps polynomial pour les graphes 3-réguliers.

8.3 Quelques résultats sur les ensembles minimaux d'angles interdits

8.3.1 Graphes triangulés

Graphes complets

Lemme 8.16 *Le nombre d'angles total dans le graphe complet à n sommets, K_n, est $3C_n^3$.*

Démonstration. Chaque triplet de points distincts (a, b, c) définit un triangle, qui a trois angles : (a, b, c), (b, c, a),(c, a, b). À chaque angle, on associe le triangle défini par ses points. Il y a C_n^3 triangles, donc il y a bien $3C_n^3$ angles. □

Proposition 8.17 *Le nombre minimal d'angle à interdire dans K_n est C_n^3.*

Démonstration. Il faut interdire au moins un angle par triangle, donc $|\mathbb{AI}| \geq C_n^3$. L'algorithme Up/Down donne ce nombre : en enlevant un sommet de K_n, on obtient le graphe K_{n-1}. Pour $n = 3$, on a bien $C_3^3 = 1$. Si on enlève un sommet de degré n, le nombre d'angles interdits est $n(n-1)/2 = C_n^2$. Comme $C_n^3 + C_n^2 = C_{n+1}^3$, le résultat est obtenu par récurrence. □

Graphes triangulés

Définition 8.18 *Un graphe est* triangulé *si dans tout cycle de sommets de longueur supérieure ou égale à 4, il existe deux sommets tels qu'il existe une arête entre ces deux sommets non adjacents dans ce cycle.*

Définition 8.19 *Un sommet est* simplicial *si son voisinage est une clique.*

Lemme 8.20 *Un graphe triangulé contient au moins un sommet simplicial.*

Si l'on élimine un sommet simplicial d'un graphe triangulé, on obtient à nouveau un graphe triangulé. Cette remarque nous donne une caractérisation ([93], Theorem 5.3.17, de Dirac) des graphes triangulés : on peut éliminer les sommets dans un ordre tel qu'au moment où ils sont éliminés, ces sommets sont simpliciaux.

Il existe un algorithme de reconnaissance de ces graphes en temps polynomial, par un parcours en largeur (Lex-BFS). Cet algorithme donne un ordre possible pour l'élimination des sommets, qui en outre conserve la connexité.

Proposition 8.21 *Si l'algorithme Up/Down est exécuté avec cet ordre, on interdit un nombre minimal d'angles.*

Démonstration. Un ensemble d'angles qui casse tous les cycles casse au moins tous les triangles. Considérons un sommet v simplicial d'un graphe triangulé. Le voisinage de ce sommet est une clique. En particulier, si t et u sont deux voisins de v, alors tuv est un triangle du graphe. En éliminant tous les angles autour de v, on interdit un et un seul angle de ces triangles, et on autorise tous les autres angles de ces triangles. Ainsi, exécuter Up/Down dans un ordre dans lequel on élimine seulement des sommets simpliciaux permet d'interdire au plus un angle par triangle, ce qui est optimal. $\square$

On peut trouver un sommet simplicial en temps polynomial. Une étude préliminaire des sommets simpliciaux se révèle donc fort utile pour minimiser le nombre d'angles à interdire : ce sont les sommets à traiter en premier quel que soit leur degré.

8.3.2 Grilles et tores

Ces graphes sont les premiers à avoir été étudiés avec le modèle des angles interdits ([54]). Ils ont été étudiés dans le cadre des graphes orientés. Nous en donnons une version non orientée.

Grilles de dimension k

Soit $\ell_1, \ldots, \ell_k \in (\mathbb{N} - \{0,1\})^k$. Une *grille* de dimension k et longueurs $\ell_1, \ldots, \ell_k$ est un graphe dont l'ensemble des sommets est $\{(i_1, \cdots, i_k) \in \mathbb{N}^{*k},\ \forall j \in \mathbb{N}_k, i_j \leq \ell_j\}$. Il y a une arête entre s_1 et s_2 si s_1 et s_2 diffèrent de 1 en exactement une coordonnée. Les circuits élémentaires sont de longueur 4 (les « faces » de la grille). De même, tous les sommets sont de degré $2k$, sauf sur les bords, où ils ont un degré $k_1 + 2k_2$, avec $k_1 + k_2 = k$. La longueur de la grille sur chaque dimension est $\ell_i - 1 > 1$, $\forall i \in \mathbb{N}_k$

Lemme 8.22 *Le nombre minimal d'angles à interdire est donné par l'algorithme glouton Up/Down et est*

$$\frac{1}{2} \sum_{i \in \mathbb{N}_k} \left[(\ell_i - 1) \left(\prod_{j \neq i} \ell_j \right) \left(\sum_{j \neq i} \frac{\ell_j - 1}{\ell_j} \right) \right].$$

Démonstration. Il faut éliminer un angle dans toutes les faces (cycles de longueur 4). En éliminant d'abord un sommet de degré minimal (Up/Down glouton), on élimine exactement un angle dans chaque face contenant cet angle. Le nombre d'angles à interdire est donc exactement le nombre de faces dans la grille et correspond exactement à un quart du nombre total d'angles. La suite de la preuve correspond au dénombrement du nombre de faces de la grille.

Nous commençons par dénombrer le nombre d'arêtes. Les arêtes sont dirigées selon k directions. Sur la direction i, la longueur de la grille est ℓ_i et le nombre d'arêtes selon cette direction est $(\ell_i - 1)\prod_{j\neq i}\ell_j$. Le nombre total d'arêtes est donc

$$\left(\prod_{i\in\mathbb{N}_k}\ell_i\right)\left(\sum_{i\in\mathbb{N}_k}\frac{\ell_i-1}{\ell_i}\right).$$

Comptons maintenant les faces dont deux des arêtes sont dirigées selon la direction i. Pour cela, il suffit de compter le nombre d'arêtes sur la grille de dimension $k-1$, projetée de la grille parallèlement à la direction i : une de ces arêtes définit l'autre direction des arêtes de la face. Comme il y a $\ell_i - 1$ « niveaux » possibles pour une face contenant des arêtes dans la direction i, le nombre de faces cherché est

$$(\ell_i - 1)\left(\prod_{j\neq i}\ell_j\right)\left(\sum_{j\neq i}\frac{\ell_j-1}{\ell_j}\right).$$

Quand on fait la somme de ce nombre de faces pour toutes les directions, on a compté chaque face exactement deux fois. Le nombre de faces de la grille est donc

$$\frac{1}{2}\sum_{i\in\mathbb{N}_k}\left[(\ell_i - 1)\left(\prod_{j\neq i}\ell_j\right)\left(\sum_{j\neq i}\frac{\ell_j-1}{\ell_j}\right)\right].$$

□

Exemple 8.23 La figure 8.16 montre une grille de dimension 3, avec une longueur 2 pour chaque dimension et une grille de dimension 2 avec longueur 5 et 7 en chaque dimension. Le nombre d'angles interdits est 6 dans le premier cas et $4 \times 6 = 24$ dans le deuxième cas.

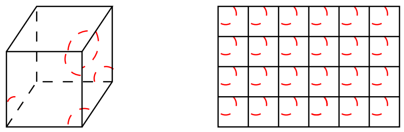


FIG. 8.16 – Grilles de dimension 3 et de dimension 2.

Tores de dimension k

Soit $\ell_1, \ldots, \ell_k \in (\mathbb{N} - \{0,1\})^k$. Un *tore* de dimension k et longueurs $\ell_1, \ldots, \ell_k$ est un graphe dont l'ensemble des sommets est $\{(i_1, \cdots, i_k) \in \mathbb{N}^{*k},\ \forall j \in \mathbb{N}_k, i_j \leq \ell_j\}$. Il y a une arête entre s_1 et s_2 si s_1 et s_2 en exactement une coordonnée i, et cette différence est 1 modulo ℓ_i.

Ici, tous les sommets ont même degré, $2k$.

Lemme 8.24 *Le nombre d'angles minimal à interdire est* $\ell_1 \times \cdots \times \ell_k \times k(2k-1)/4$.

Démonstration. Tout se passe comme dans les grilles, mais le nombre de carrés élémentaires change : tous les sommets ont même degré, et un angle définit un carré élémentaire, puisque le quatrième sommet peut s'en déduire. En tout, il y a donc $\ell_1 \times \cdots \times \ell_k \times (2k(2k-1))/2$ angles, et il faut en interdire $\ell_1 \times \cdots \times \ell_k \times k(2k-1)/4$. □

8.3.3 Graphes réguliers

On s'intéresse maintenant aux graphes de petit degré, les graphes 2-réguliers et les graphes 3-réguliers.

Graphes 2-réguliers

Ce sont les graphes composés d'un seul cycle. Il faut donc interdire un seul angle, quelconque.

Graphes 3-réguliers

Définition 8.25 *Un graphe* (V, E) *est* 2-arêtes connexe *si enlever une arête quelconque de ce graphe ne déconnecte pas la graphe.*

Théorème 8.26 *Trouver une numérotation optimale pour l'algorithme Up/Down peut se faire en temps polynomial. Le nombre d'angles à interdire est* $n/2 + c$ *où* c *est le nombre de feuilles de l'arbre des composantes 2-arêtes connexes et* n *le nombre de sommets du graphe.*

Démonstration.

1. **Une orientation du graphe.** On oriente les arêtes du graphe pour que ce soit un graphe orienté acyclique de la manière suivante : on numérote les sommets, et une arête est orientée vers le sommet de plus grande numérotation. On note m_i le nombre de sommets ayant i flèches sortantes.
2. **Le nombre d'angles interdits ne dépend que de m_0 et m_3.** On a $m_0+m_1+m_2+m_3 = n$. On a aussi $m_1 + 2m_2 + 3m_3 = 3n/2$, le nombre d'arcs. Le nombre d'angles interdits par l'algorithme Up/Down est $3m_3 + m_2 = 3n/2 - m_2 - m_1 = 3n/2 - n + m_3 + m_0 = n/2 + m_3 + m_0$.
3. **Décomposition du graphe en ses composantes 2-arêtes connexes.** Un *isthme* est une arête dont l'élimination déconnecte le graphe. Si on enlève les isthmes du graphe, on obtient une partition des sommets en ses composantes 2-arêtes connexes. Soient U une telle composante et a et z deux sommets de cette composante. On va construire dans U une orientation du graphe telle tout sommet a un arc sortant et un arc entrant au moins, sauf a qui a au moins un arc sortant et z qui a au moins un arc entrant.
4. **Construction d'une orientation pour une composante.** La composante U est 2-arêtes connexe, donc il existe un chemin simple de a à z. On note U' l'ensemble des nœuds de ce chemin. On oriente les arcs de ce chemin selon cette orientation et les arcs entre deux sommets de ce chemin de manière à ne former aucun circuit. On a donc un ordre total sur les nœuds choisis, le plus petit étant a et le plus grand étant z. Si $U' = U$, on n'a plus rien à faire. Sinon on augmente le nombre de sommets choisis :
   (a) S'il existe un sommet $v \in U - U'$ ayant au moins deux voisins dans U', u et w, $u < w$. On oriente les arêtes entre v et les sommets de U' de telle sorte que $u < v < w$. On ajoute v à U', et on a toujours un ordre sur les nœuds de U'.
   (b) S'il existe un chemin $uv_1 \dots v_k w$ sans cycle de u à w, $u, w \in U'$ et $u < w$, dont les nœuds intermédiaires n'appartiennent pas à U', on procède de la même manière, sauf qu'on ordonne plus d'un nœud : on a $u < v_1 < \cdots < v_k < w$. On ajoute ces nœuds à U', et on a toujours un ordre sur les nœuds de U'. Noter que dans ces deux cas, tous les nœuds insérés ont au moins un arc entrant et un arc sortant.

(c) Supposons qu'il n'existe pas de couple $(u, w) \in U'^2$ tel qu'il existe un chemin de u à w dans $U - U'$. Mais alors, si on considère un arc uv avec $u \in U'$ et $v \in U - U'$, si on enlève cet arc, on déconnecte U, ce qui est en contradiction avec le fait que U est une composante 2-arêtes connexe. Ce n'est donc pas possible, et on se trouve forcément dans l'un des deux cas ci-dessus.

On peut donc itérer la construction ci-dessus jusqu'à ce que $U = U'$, et construire pour chaque composante 2-arêtes connexe une orientation sur les arêtes.

5. **Orientation du graphe.** Si l'on réduit les composantes 2-arc connexes à un seul sommet, et que l'on place une arête s'il existe un isthme entre ces deux composantes, on obtient un arbre. On oriente cet arbre de telle sorte que seules les feuilles peuvent avoir un degré entrant ou sortant nul. Ceci nous donne une orientation sur les isthmes. Ensuite, on oriente les composantes 2-arc connexes : on choisit le sommet a parmi les sommets dont un isthme est un arc entrant, et le sommet z parmi les sommets dont un isthme est un arc sortant, sous réserve d'existence de ces isthmes (sinon, le sommet est choisi de manière arbitraire). Pour les composantes qui ne sont pas des feuilles, tous les nœuds ont un arc entrant et un arc sortant au moins. Pour les feuilles, il existe exactement un nœud n'ayant pas d'arc entrant ou pas d'arc sortant. Par construction de cette orientation, $m_0 + m_3$ est alors égale au nombre de feuilles de l'arbre.
6. **Cette orientation est optimale.** En effet, dans les feuilles, il y a nécessairement un nœud choisi en premier ou en dernier, et donc un nœud de degré sortant ou entrant nul.

□

La figure 8.17 illustre la construction de l'orientation du graphe.

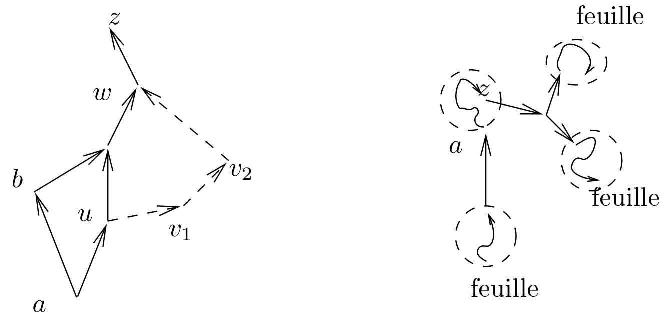


FIG. 8.17 – Construction d'une orientation. À gauche construction pour une composante 2-arêtes connexe, à droite, l'arbre des composantes 2-arêtes connexes.

Exemple 8.27 La figure 8.18 montre un exemple d'un graphe régulier à plusieurs feuilles (3). Le graphe a 16 nœuds, et donc le nombre d'angles interdits est 11.

*
* *

Dans ce chapitre nous avons étudié la question de l'interdiction d'angles dans les graphes. Malgré les problèmes résolus pour certaines classes de graphes, le mystère reste entier pour le cas général, aucune de nos tentatives pour savoir si le problème est NP-complet ou pas n'ayant abouti pour le moment, aussi bien pour le problème avec contrainte de connexité ou sans. Toutefois, nous avons pu cerner l'importance de certains aspects : les sommets simpliciaux sont des cas faciles à traiter, et il faut maintenant se focaliser sur les graphes qui en sont dépourvus.

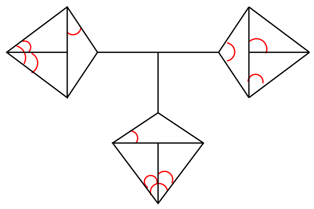


Fig. 8.18 – Exemple de graphe 3-régulier.

Chapitre 9

Garanties de performance sur des réseaux avec angles interdits

Dans le chapitre 8, nous avons introduit le concept d'interdiction d'angles dans les réseaux pour empêcher les blocages et dans le chapitre 7, nous avons étudié les opérations du *network calculus*. Dans ce dernier chapitre, nous faisons le lien entre ces deux notions : comment appliquer le network calculus pour calculer des garanties de performances dans ces réseaux ? Dans une première section, nous donnons des méthodes pour calculer des performances lorsque les chemins empruntés par les flux sont connus. Dans une seconde partie, nous cherchons un chemin optimal pour un flux.

9.1 Calculs de performances d'un ensemble de chemins

Considérons un réseau, dont les liens sont assimilés à des serveurs de capacité constante. Ce réseau est représenté par un graphe $G = (V, E)$ et chaque arête $e = (a, b)$ est assimilée à deux serveurs, l'un de a vers b, de capacité $c_{a,b}$, et l'autre de b vers a, de capacité $c_{b,a}$. On note $\mathbb{AI}$ un ensemble d'angles interdits de G qui casse les cycles et préserve la connexité du graphe.

Soit $F = (f_i)_{i \in \mathbb{N}_k}$ un ensemble de k flux qui traversent le réseau. Le flux f_i suit le chemin $u_1^{(i)}, \ldots, u_{\ell_i}^{(i)} \in V$ dans le réseau et est (σ, ρ) supérieurement contraint. Dans cette section, on cherche à calculer des garanties de performances pour ces flux (nombre de clients dans la file, délai de transmission des paquets).

9.1.1 Une transformation du graphe

On commence par transformer le graphe G en un graphe orienté acyclique noté $G_{\mathbb{AI}} = (N, A)$ en tenant compte des angles interdits. Les nœuds de $G_{\mathbb{AI}}$ sont les arêtes orientées de G :

$$N = \bigcup_{e=(a,b) \in E} \{(a, b), (b, a)\}.$$

Il y a un arc de (a, b) vers (b, c) si (a, b, c) est un angle autorisé :

$$A = \mathbb{A} - \mathbb{AI}.$$

Le graphe $G_{\mathbb{AI}}$ est bien acyclique. Sinon, il existerait un cycle d'angles, ce qui n'est pas possible. La figure 9.1 représente un graphe avec des angles interdits et la transformation en un graphe orienté correspondant à ce graphe. Par construction, à tout chemin admissible dans G correspond un chemin de $G_{\mathbb{AI}}$ et réciproquement.

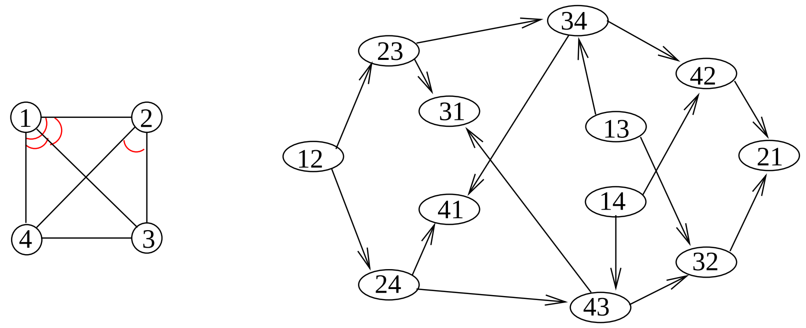


FIG. 9.1 – Transformation d'un graphe avec angles interdits en un graphe orienté acyclique.

9.1.2 Calcul des garanties de bout en bout pour chaque flux

Dans toute la suite du chapitre, on prend en compte uniquement le graphe $G_{\mathbb{AI}}$. On considère donc des graphes orientés acycliques. Les calculs de cette partie sont des applications directes des résultats de [30, 65] et sont donnés ici à titre d'exemples de calculs.

Pour les graphes orientés acycliques, on peut ordonner les nœuds de sorte que tous les prédécesseurs d'un nœud sont inférieurs à ce nœuds. Mais alors, on peut traiter tous les nœuds un par un dans cet ordre : si les flux d'entrée et de sortie pour tous les flux sont connus pour les premiers nœuds, on peut calculer les flux d'entrée et de sortie en le nœud suivant.

Pour chaque serveur, les flux entrants sont les flux sortants des serveurs prédécesseurs de ce serveur. S'ils ont déjà été traités, on peut déduire les garanties sur les flux sortants de ce serveur. Soient A_i, $i \in I$, les flux entrants dans un serveur de capacité c, (σ'_i, ρ'_i) supérieurement contraints. Le flux total dans le serveur est alors $(\sum \sigma'_i, \sum \rho'_i)$ supérieurement contraint. Le serveur est stable si $\sum \rho'_i < c$. Alors, par la proposition 7.9, le nombre de paquets en attente est au plus de

$$\sum \sigma'_i$$

et le délai au plus de

$$\lceil \frac{\sum \sigma'_i}{c - \sum \rho'_i} \rceil.$$

Pour chaque flux, au pire, les paquets du flux f_1 voient un f-serveur avec

$$f(t) = ((c - \sum_{i\in\{2,\dots,k\}} \rho'_i)t - \sum_{i\in\{2,\dots,k\}} \sigma'_i)_+.$$

Soit B_1 le flux de f_1 en sortie du serveur. Le flux B_1 est alors f'_1 supérieurement contraint avec $f'_1(0) = 0$ et $\forall t \geq 1$,

$$\begin{aligned} f'_1(t) &= \sup_{s\geq 0}(\rho'_1(t+s) + \sigma'_1 - f(s)) \\ &= \sup_{s\geq 0}(\rho'_1.(t+s) + \sigma'_1 - (cs - \sum_{i\in\{2,\dots,k\}} \rho'_i s - \sum_{i\in\{2,\dots,k\}} \sigma'_i)_+) \\ &= \rho'_1 t + \sigma'_1 + \sup_{s\geq 0}(\rho'_1 s + (cs - \sum_{i\in\{2,\dots,k\}} \rho'_i s - \sum_{1\in\{2,\dots,k\}} \sigma'_i)_+). \end{aligned}$$

La borne supérieure est obtenue pour $s = \frac{\sum_{1\in\{2,\dots,k\}} \sigma'_i}{c - \sum_{i\in\{1,\dots,k\}} \rho'_i}$, ce qui donne

$$f'_1(t) = \rho'_1 t + \sigma'_1 + \rho'_1 \frac{\sum_{1\in\{2,\dots,k\}} \sigma'_i}{c - \sum_{i\in\{1,\dots,k\}} \rho'_i}.$$

Le flux en sortie du serveur est donc (σ_1'', ρ_1) supérieurement contraint avec $\sigma_1'' = \sigma_1' + \rho_1' \frac{\sum_{1\in\{2,\dots,k\}} \sigma_i'}{c-\sum_{i\in\{1,\dots,k\}} \rho_i'}$. On remarque que ρ_1' n'a pas changé. C'est donc que de serveur en serveur, le flux f_1 reste (x, ρ_1) supérieurement contraint. Par contre, la valeur x de la rafale maximum augmente. On peut déduire les performances en fonction des flux d'entrée calculés pour chaque serveur par application directe de la proposition 7.9.

Pour calculer les garanties, il vaut mieux d'abord calculer la capacité du serveur concaténation des serveurs du chemin, c'est la convolution de toutes les courbes de garanties de service des serveurs (7.8). Ainsi, les rafales sont prises en compte une seule fois par rapport à une méthode plus naïve consistant à calculer les garanties serveur par serveur.

Cette manière de faire n'est pas la meilleure. En effet, elle ne tient aucun compte des enchevêtrements des flux dans le réseau. Donc les rafales des autres flux peuvent être prises en compte plusieurs fois dans ce calcul.

9.1.3 Exemple

On considère la suite de serveurs de la figure 9.2. Le serveur s_1 a une capacité $c_1 = 4$, le serveur s_2 une capacité $c_2 = 5$ et le serveur s_3 une capacité de $c_3 = 4$. Le flux f_1 est $(2, 1)$ supérieurement contraint, f_2 est $(1, 2)$ supérieurement contraint et f_3 est $(3, 1)$ supérieurement contraint.

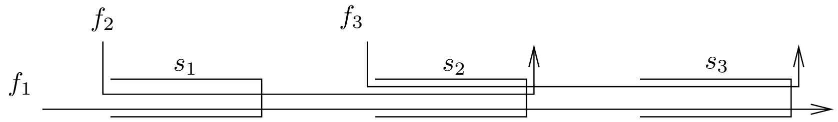


FIG. 9.2 – Exemple de calculs de performances.

Serveur s_1 En sortie de s_1, f_1 est $(3, 1)$ supérieurement contraint et f_2 est $(5, 2)$ supérieurement contraint.

Serveur s_2 En sortie de s_2, f_1 est $(11, 1)$ supérieurement contraint, f_2 est $(17, 2)$ supérieurement contraint et f_3 est $(11, 1)$ supérieurement contraint.

Serveur s_3 En sortie de s_2, f_1 est $(33/2, 1)$ supérieurement contraint et f_3 est $(33/2, 1)$ supérieurement contraint.

Un calcul plus précis

Les flux f_1 et f_2 suivent le même chemin sur les deux premiers serveurs. Pour calculer le flux f_3 après passage dans s_2, on peut considérer que f_1 et f_2 sont un seul flux, $(3, 3)$ supérieurement contraint. En effet, les calculs précédents sont des calculs dans le pire des cas, donc globalement, les performances sont meilleures que si l'on ajoute les deux performances dans le pire des cas. En sortie de s_1, le flux $f_1 + f_2$ est $(3, 3)$ supérieurement contraint ; en sortie de s_2, f_3 est $(6, 3)$ supérieurement contraint. La garantie sur ce flux est donc améliorée en sortie de ce serveur, mais aussi en sortie des autres, puisqu'ils dépendent des flux calculés.

9.2 Optimisation d'un flux dans un réseau

On considère un flux d'un sommet du graphe vers un autre et l'on cherche un chemin pour ce flux qui optimise les garanties de performance. En théorie des graphes, on peut vouloir trouver le chemin le plus court d'un point à un autre. Ceci correspond, en terme de performance à trouver le temps le plus court pour aller d'un point à un autre si le temps de passage dans un nœud est constant. On rappelle tout d'abord l'algorithme de Dijkstra qui permet de trouver le chemin le plus court dans un graphe dirigé, puis on adapte cet algorithme au cas du calcul de garanties dans les réseaux : le temps d'attente dans un nœud n'est plus constant.

9.2.1 Algorithme de Dijkstra pour des flux non-constants

Rappel de l'algorithme de Dijkstra

Soit $G = (N, A)$ un graphe dirigé acyclique pondéré avec des poids positifs sur les arêtes. Soient u et v deux nœuds de G. L'algorithme de Dijkstra permet de trouver le chemin de poids minimal de u à v. Le principe de cet algorithme est de parcourir les nœuds du graphe et de mettre à jour les distances de u aux points parcourus en fonction de la distance de u à leurs prédécesseurs et du poids des arcs les reliant à leurs prédécesseurs. On maintient toujours ces valeurs à la plus petite distance. Parcourir tous les nœuds une fois est suffisant quand on choisit un bon ordre. Cet ordre consiste à prendre le nœud non encore traité qui est à une distance provisoire minimum de u. Pour plus de détails, on pourra consulter des ouvrages d'algorithmiques des graphes tels que [93]. Nous avons choisi l'algorithme de Dijkstra, dont la complexité est linéaire pour des graphes denses ($|A| = O(|N|^2)$). Pour des graphes peu denses, il vaut mieux partir d'un algorithme de calcul des plus courts chemins basé sur un tri topologique car les graphes considérés sont acycliques (complexité en $O(|N| + |A|)$). Dans ce cas, le même type d'adaptation à notre problème de flux s'applique.

L'algorithme 9.1 présente une version légèrement modifiée de l'algorithme de Dijkstra : les poids ne sont plus sur les arcs, mais sur les nœuds. Le poids d'un chemin est alors la somme des poids des nœuds du chemin. L'ensemble des voisins d'un nœud x est $Voi(x)$.

Algorithme 9.1 : Algorithme de Dijkstra modifié

Données : Un graphe orienté $G = (N, A)$ pondéré par $(w_n)_{n\in N}$, $u, v \in N$.
Résultat : Le poids du chemin de poids minimal de u à v.
début
  $d(u) \leftarrow w(u)$;
  pour chaque $x \in Voi(u)$ **faire** $d(x) \leftarrow d(u) + w(x)$;
  pour chaque $x \in N - Voi(u) - \{u\}$ **faire** $d(x) \leftarrow \infty$;
  $S \leftarrow \{u\}$;
  répéter
    $x \leftarrow$ le nœud $N - S$ tel que $d(x) = \min_{y\in N-S} d(y)$;
    $S \leftarrow S \cup \{x\}$;
    pour chaque $y \in Voi(x)$ **faire** $d(y) \leftarrow \min(d(y), d(x) + w(y))$;
  jusqu'à $S = N$ *ou* $\forall x \notin S,\ d(x) = \infty$;
fin

On adapte maintenant l'algorithme pour des flux non constants. Le problème est le suivant : étant donné un flux de données d'un nœud u vers un nœud v (σ, ρ) supérieurement contraint, f_x

des serveurs, représentés par les nœuds du graphe, quel est le chemin qui optimise les garanties pour ce flux. On suppose que les services garantis pour le flux que l'on considère sont indépendants pour chaque serveur, c'est-à-dire que les flux transversaux sont indépendants les uns des autres. Ce n'est pas vrai dans le cas général, puisque les services sont fonctions des autres flux dans le réseau, mais cette hypothèse permet d'obtenir des bornes de garanties de services. Remarquons de plus que si l'on peut trouver un chemin garantissant des bornes finies sur les garanties, le système dans son ensemble reste stable, car le réseau considéré est acyclique. On considère deux types de garanties, le délai et le nombre de paquets dans le système.

Une première hypothèse : les courbes de service sont convexes

Quand aucun flux ne traverse les graphes, les serveurs se comportent comme des liens conservatifs de capacité constante et sont donc bien convexes. Soit f une fonction de $\mathbb{N}$ dans $\mathbb{N}$ convexe croissante. Considérons un f-serveur et un flux passant par ce serveur (σ, ρ) supérieurement contraint. Si un autre flux arrive, ce serveur sera pour lui un f_1-serveur, avec $f_1(t) = (f(t) - \rho t - \sigma)_+$. La fonction f_1 est aussi convexe, puisque f et $t \mapsto -\rho t - \sigma$ le sont, et que la convexité est conservée par les opérations maximum et addition.

De plus, les flux garantis seront toujours (σ, ρ) supérieurement contraints.

Lemme 9.1 *Soit un f-serveur, avec f convexe et croissante, et A un flux passant dans ce f-serveur (σ, ρ) supérieurement contraint. Alors le flux de sortie est (σ', ρ) supérieurement contraint avec $\sigma' = \sigma + \sup_{s \geq 0}[\rho s - f(s)]$.*

Démonstration. Le flux de sortie est supérieurement contraint par h, avec ([30], Lemma 2.3.6)

$$h(t) = \begin{cases} 0 & \text{si } t = 0, \\ \sup_{s \geq 0} \rho(t+s) + \sigma - f(s) & \text{sinon.} \end{cases}$$

Donc, $\forall t > 0$, on a

$$h(t) = \sup_{s \geq 0}[\rho s - f(s)] + \sigma + \rho t.$$

La fonction h est affine, et $\sigma' = \sigma + \sup_{s \geq 0}[\rho s - f(s)]$. □

Il paraît donc justifié de ne considérer que des courbes de service convexes. De plus, puisque les réseaux étudiés sont acycliques, les opérations utilisées pour le calcul des performances n'impliquent pas la fermeture sous-additive. Les fonctions seront donc toujours affines par morceaux.

9.2.2 Nombre de paquets d'un flux dans le système

Le nombre de paquets dans un système est la différence entre le nombre de paquets entrés et le nombre de paquets servis. Le nombre de paquets servis à la date t par un f-serveur est $f(t)$, et le nombre de paquets entrés est au plus $\sigma + \rho t$. Le nombre maximal de paquets dans le système est donc $q = \max_t[\sigma + \rho t - f(t)]$.

Proposition 9.2 *Soit f une fonction convexe croissante affine par morceaux. Considérons un f serveur et un flux entrant (σ, ρ) supérieurement contraint. Soit t_0 le plus petit $t \in \mathbb{N} \cup \{\infty\}$ tel que $f(t+1) - f(t) \geq \rho$. Alors, $q = \sigma + \rho t_0 - f(t_0)$, c'est-à-dire, le nombre maximal de paquets dans le système est donné à l'instant où la pente de la courbe de service devient plus grande que la pente des arrivées.*

Démonstration. Soit $Q(t) = \sigma + \rho t - f(t)$. Comme f est convexe, $t \mapsto f(t+1) - f(t)$ est croissante. Sur $[0, t_0]$, Q est donc décroissante et sur $[t_0, \infty[$, Q est décroissante. Le maximum de Q est donc obtenu en t_0. □

On présente maintenant un algorithme qui permet de trouver le chemin qui garantisse un nombre minimal de paquets dans le système, basé sur l'algorithme 9.1. Il faut prouver que s'il existe deux chemins de u à x, c_1 et c_2 et un chemin c_3 de x à v, alors si c_1 garantie de meilleures performances que c_2, alors c_1c_3 garantie de meilleures performances que c_2c_3. C'est ce que l'on se propose de faire dans le théorème suivant.

Théorème 9.3 *Soient f_1, f_2, f_3 trois fonctions convexes croissantes de $\mathbb{N}$ dans $\mathbb{N}$, et $g : t \mapsto \sigma + \rho t$. On note $q_i = \max_t[\sigma + \rho t - f_i(t)]$ et $q'_i = \max_t[\sigma + \rho t - f_i * f_3(t)]$, $i \in \{1, 2\}$. Alors, si $q_1 \leq q_2$, $q'_1 \leq q'_2$.*

Démonstration. Soit t_i le plus petit entier tel que $f_i(t_i + 1) - f_i(t_i) \geq \rho$. Le plus petit entier tel que $f_1 * f_3(t+1) - f_1 * f_3(t) \geq \rho$ est $t_1 + t_3$. En effet, soit s tel que $f_1 * f_3(t) = f_1(s) + f_3(t-s)$. Si $t < t_1 + t_3$, alors soit $s < t_1$, soit $t - s < t_3$. Supposons, sans perte de généralité, que $s < t_1$. Alors

$$\begin{aligned} f_1 * f_3(t+1) &\leq f_1(s+1) + f_3(t-s) \\ &\leq f_1(s+1) - f_1(s) + f_1(s) + f_3(t-s) \\ &\leq \rho + f_1 * f_3(t). \end{aligned}$$

Si $t \geq t_1 + t_3$, on a nécessairement $s \geq t_1$ et $t - s \geq t_3$: si on avait $s < t_1$ par exemple,

$$\begin{aligned} f_1 * f_3(t) &= f_1(t_1) - \rho_1(t_1 - s) + f_3(t - t_1) + \rho_3(t_1 - s) \\ &\geq f_1(t_1) + f_3(t - t_1), \end{aligned}$$

puisque $\rho_1 \leq \rho \leq \rho_3$. Mais alors, $f_1 * f_3(t+1) - f_1 * f_3(t) \leq \rho$.

Il nous reste maintenant à comparer $q_1 - q_2$ à $q'_1 - q'_2$. D'après la proposition précédente, $q_i = \sigma + \rho t_i - f_i(t_i)$, et $q'_i = \sigma + \rho(t_i + t_3) - f_i(t_i) - f_3(t_3) = q_i + \rho t_3 - f_3(t_3)$. On a donc $q_1 - q_2 = q'_1 - q'_2$, ce qui achève la preuve. □

La preuve du théorème donne aussi une méthode pour calculer le nombre de paquets dans le système dans faire le calcul des convolutions. L'algorithme 9.2 permet de calculer le chemin qui donne la meilleure performance.

Avec la représentation des fonctions du chapitre précédent, il est facile de calculer t_x : si une fonction f est représentée par $[(x_i, y_i, \rho_i), i \in \mathbb{N}_k]$, alors ce point est à chercher dans les $x_i, x_i - 1$.

9.2.3 Délai maximal des paquets

On s'intéresse maintenant à la durée maximale de séjour des paquets dans le système. On suppose toujours un flux (σ, ρ) supérieurement contraint. Si ce flux passe au travers d'un f-serveur, alors le délai maximal, si le serveur est FIFO, pour transmettre un paquet est $d = \inf\{s \geq 0 \mid \sigma + \rho t \leq f(s+t),\ \forall t \in \mathbb{N}\}$.

On cherche maintenant à minimiser le délai de transmission des paquets pour un flux dans un réseau, et donc un chemin dans le réseau minimisant ce délai.

Proposition 9.4 *Soit f une fonction convexe affine par morceaux et croissante. On considère un f-serveur et un flux (σ, ρ)-supérieurement contraint entrant dans ce serveur. Soit t_0 le plus*

Algorithme 9.2 : Dijkstra pour la file d'attente minimale

Données : Un graphe orienté $G = (N, A)$, f_n, $n \in N$ des courbes de services pour les serveurs de chaque nœud, $u, v \in N$, (σ, ρ) la courbe des arrivées.
Résultat : Chemin de u à v minimisant le nombre de paquets dans le système
début
  pour chaque $x \in N$ **faire**
    $t_x \leftarrow \min\{t \mid f_x(t+1) - f_x(t) \leq \rho\}$;
    $w(x) \leftarrow \rho t_x - f_x(t_x)$;
  finprch
  $q(u) \leftarrow \sigma + \rho t_u - f_u(t_u)$;
  pour chaque $x \in Voi(u)$ **faire** $q(x) \leftarrow d(u) + w(x)$;
  pour chaque $x \in N - V(u) - \{u\}$ **faire** $q(x) \leftarrow \infty$;
  $S \leftarrow \{u\}$;
  répéter
    $x \leftarrow$ le nœud $N - S$ tel que $q(x) = \min_{y \in N-S} q(y)$;
    $S \leftarrow S \cup \{x\}$;
    pour chaque $y \in Voi(x)$ **faire** $q(y) \leftarrow \min(q(y), q(x) + w(y))$;
  jusqu'à $S = N$ *ou* $\forall x \notin S,\ q(x) = \infty$;
fin

---

petit $t \in \mathbb{N} \cup \{\infty\}$ *tel que* $f(t+1) - f(t) \geq \rho$. *Alors,*

$$d = \begin{cases} f^{-1}(\sigma) & si\ t_0 \leq \sigma \\ t_0 - \frac{f(t_0)-\sigma}{\rho} & si\ t_0 \geq \sigma. \end{cases}$$

Démonstration. Si $f(t_0) \leq \sigma$, alors pour tout $s \geq f^{-1}(\sigma)$, $f(s+t) - \rho t \geq f(s)$, car f étant croissante, $t_0 \leq f^{-1}(\sigma)$, et l'accroissement moyen pour les $s \geq t_0$ est plus grand que ρ. Mais alors, $f(s+t) - \rho t \geq f(f^{-1}(\sigma)) = \sigma$. Donc $d \leq f^{-1}(\sigma)$. Pour tout $s < f^{-1}(\sigma)$, $f(s+t) - \rho t \leq f(s) < \sigma$. Donc $d \geq f^{-1}(\sigma)$. On a donc montré que dans ce cas, $d = f^{-1}(\sigma)$.

Si $f(t_0) \geq \sigma$, le délai peut être calculé comme le maximum de la différence horizontale entre f et $t \mapsto \rho t + \sigma$. Le maximum de cette distance est nécessairement à $f(t_0)$, ce qui permet d'obtenir le résultat. □

Le délai maximal peut s'observer sur un repère où l'on a tracé f et $t \mapsto \sigma + \rho t$. C'est le maximal de la distance horizontale entre les deux courbes à l'ordonnée σ et $f(t_0)$.

Le fait qu'il y ait deux valeurs possibles pour le délai maximal fait qu'on ne peut pas appliquer l'algorithme de Dijkstra : la propriété qui fait qu'un chemin localement optimal est globalement optimal n'est plus valable. En effet, soit $f_1 : t \mapsto \max(0, 2t - 10)$, $f_2 : t \mapsto \max(t/3, 2t - 20)$ et $f_3 : t \mapsto \max(0, t/3 - 2, 2x - 22)$, et un flux d'arrivée $(2, 1/2)$ supérieurement contraint. Ces fonctions sont représentées figure 9.3. Le délai maximal pour f_1 est $d_1 = 6$, et pour f_2, $d_2 = 8$. Regardons maintenant le délai pour les $f_1 * f_3$ et $f_2 * f_3$- serveurs (d'_1 et d'_2 respectivement). On a $d'_1 = 17$ et $d'_2 = 16$. Il n'y a donc pas de monotonie pour le délai.

Cette non-monotonicité vient du fait qu'il faut comparer deux valeurs sur pour trouver le délai, contrairement au nombre de paquets maximum dans un système.

Lemme 9.5 *Soient f_1, f_2 et f_3 trois fonctions convexes, affines par morceau et croissantes, et $\sigma, \rho \in \mathbb{Q}_+$. Soient t_i les entiers à partir desquels $f_i(t_i+1) - f_i(t_i) > \rho$. Soient $d_i = t_i - \frac{f_i(t_i)-\sigma}{\rho}$ et $d'_i = t_i + t_3 - \frac{f_i * f_3(t_i+t_3)-\sigma}{\rho}$. Alors, $d_1 \leq d_2 \Rightarrow d'_1 \leq d'_2$.*

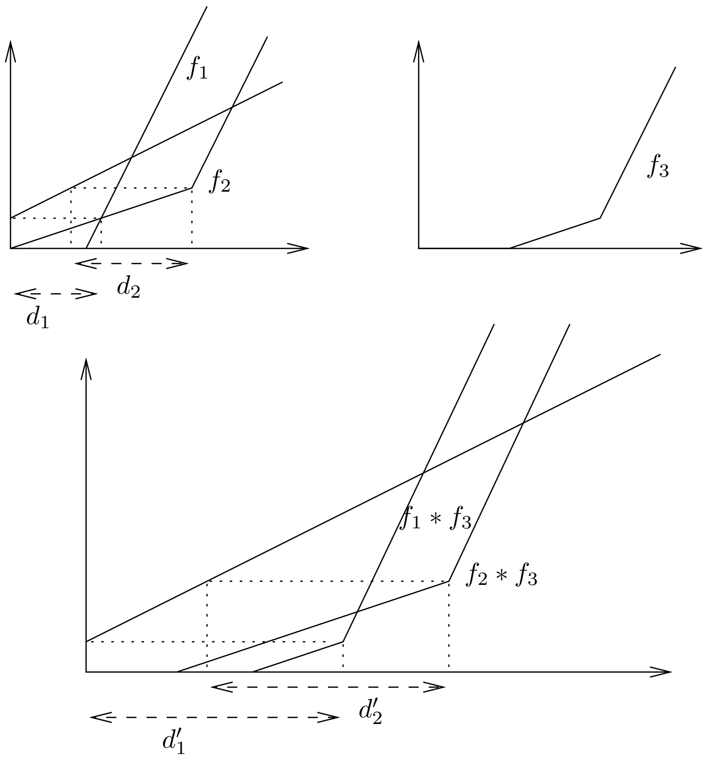


FIG. 9.3 – Le délai n'est pas monotone pour la concaténation de serveurs.

Démonstration. On a déjà montré dans la preuve du théorème 9.3 que $f_i * f_3(t_i + t_3) = f_i(t_i) + f_3(t_3)$. Le résultat se déduit très facilement de cette égalité. □

Une première solution pour adapter l'algorithme de Dijkstra est de faire une majoration du délai d'un f-serveur par $t_0 - (f(t_0) - \sigma)/\rho$. Les calculs sont exacts si $\sigma \geq t_0$ et l'erreur reste bornée sinon.

*
* *

Dans ce chapitre, nous avons appliqué les méthodes du network calculus pour les réseaux avec interdiction d'angles, tout d'abord en calculant les garanties de performances sur les chemins que prennent les flux, puis en optimisant un chemin pour un flux, afin de minimiser le nombre de paquets de ce flux dans le réseau. De nombreux points restent à aborder. En premier lieu, il serait intéressant de chercher un équilibre global du réseau, c'est-à-dire une configuration, telle que pour chacun des flux, le chemin de ce flux est optimal compte tenu des autres flux. Un deuxième point à aborder serait le choix de l'ensemble d'angles à interdire. En effet, dans le chapitre 8, nous avons cherché à interdire un nombre minimal d'angles. Or, ici, on cherche plutôt optimiser des performances qui dépendent très fortement des capacités des liens. Un petit nombre d'angles interdits ne serait donc pas nécessairement optimal pour les garanties de performances. D'autre part, nous avons vu dans ce chapitre comment adapter l'algorithme de Dijkstra pour l'optimisation des garanties. Il existe de nombreux algorithmes de flots sur les graphes orientés. L'étude de ces algorithmes pourrait nous aider à aborder le problème des garanties pour les flux dans les réseaux.

Conclusion et perspectives

Au cours de cette thèse, les systèmes à événements discrets ont été abordés sous trois angles différents.

Tout d'abord, nous avons généralisé certaines études faites pour les monoïdes de traces aux groupes de traces. Cela nous a permis d'exhiber une formule simple pour la série génératrice des groupes de traces. De plus, nous avons montré l'existence d'un taux de croissance asymptotique pour la hauteur des tas de pièces modélisant les traces des groupes de traces. Mais le calcul de ce taux de croissance est difficile et n'a pu être fait que pour les groupes commutatifs libres et les groupes libres. Le calcul de ce taux pour les autres groupes de traces, mêmes les plus simples reste encore un problème ouvert.

Une deuxième partie de cette thèse a été consacrée à l'étude des réseaux de Petri temporisés. Nous avons tout d'abord défini la notion de débit, qui est une mesure de la performance, dans ces réseaux puis étudié ce débit pour plusieurs classes de réseaux : les réseaux avec des temporisations constantes et avec des temporisations aléatoires. Nous avons aussi introduit une méthode d'échantillonnage exact des marquages s'inspirant de l'algorithme de Propp et Wilson.

Une grande partie de notre travail a été l'étude de la dépendance du débit en fonction des routages. Pour les réseaux à choix libres 1-bornés avec des temporisations constantes, nous avons trouvé une caractérisation d'un routage donnant le débit le plus faible. Quand les temporisations sont rationnelles, le débit minimal et le débit maximal sont atteints pour des routages périodiques. Quand les temporisations sont aléatoires, nous avons aussi comparé plusieurs classes de routages.

Concernant ces derniers points, beaucoup de problèmes restent ouverts et dignes d'intérêt : peut-on généraliser les résultats obtenus pour les réseaux à choix libres 1-bornés aux réseaux à choix libres bornés ? Dans le cas des temporisations constantes, les techniques utilisées, l'introduction des TT-invariants et la modélisation par tas de pièces, sont spécifiques aux réseaux 1-bornés. Il semble donc difficile d'adapter notre preuve. D'autre part, dans le cas des temporisations aléatoires, même dans le cas des réseaux 1-bornés, on n'a pas pu établir de hiérarchie entre le meilleur routage périodique et le meilleur routage Bernoulli, alors que sur les exemples étudiés, le meilleur routage périodique semble être le routage équilibré. Pour une première approche simplificatrice de ce problème, on pourrait se restreindre à l'étude des réseaux avec une seule place de choix. Dans ce cas, certaines observations ont pu être faites : le meilleur routage périodique pour une proportion de routage donnée semblent être les routages équilibrés. Des résultats semblables ont été obtenus pour des modèles de tas de pièces pouvant modéliser des réseaux à choix libres 1-bornés avec une place de choix et des temporisations constantes. Il est montré dans [25] que les empilements de deux types de pièces ont un taux de croissance moyen minimal pour des choix équilibrés des pièces. Dans [6, 62], des résultats du même type ont été obtenus pour le routage optimal dans deux files d'attente en boucle ouverte, des temps de service stationnaires et généraux, les calculs ayant été effectués dans le cas des temps de service exponentiels et d'arrivées poissonniennes. Il ne serait donc pas très surprenant d'obtenir le même type de résultats pour les réseaux à choix libres avec une place de choix et des temporisations exponentielles. Pour des réseaux de plus grande taille et plusieurs places de choix, la question de savoir ce que serait la bonne notion de routage équilibré n'est pas claire. De plus, quand il y a plusieurs places de routages, il faut tenir compte des déphasages possibles entre les routages équilibrés des différentes places de choix. Connaître la dépendance du débit en fonction des déphasages des fonctions de routage est un problème qui n'a pas encore été abordé.

La dernière partie de la thèse tente d'appliquer les concepts du *network calculus* à certains types de réseaux. Le premier chapitre de cette partie établit des résultats de stabilité de certaines classes de fonctions pour les opérations du network calculus. Nous avons aussi commencé à développer une algorithmique pour le calcul effectif et efficace de ces opérations pour la classe stable, les fonctions ultimement pseudo-périodiques. Cette classe de fonctions contient les fonctions les plus utilisées (les fonctions affines et affines par morceaux) et c'est une classe de fonctions ayant une représentation finie. Nous visons maintenant à améliorer ces algorithmes, en trouvant des structures de données plus adaptées ou des méthodes de calculs plus efficaces, en vue de l'implémentation de ces algorithmes.

La suite de cette partie est dédiée à l'étude de certains types de routage dans les réseaux : les routages avec interdiction d'angles. Nous nous sommes tout d'abord intéressés à la minimisation du nombre d'angles à interdire afin de casser tous les cycles et empêcher ainsi le blocage dans les réseaux. C'est le problème qui semble le plus naturel au premier abord. Savoir si ce problème est NP-complet est encore une question ouverte. Un autre point, qui semble plus adapté à l'étude des performances de ces réseaux, est l'application des techniques du network calculus. Dans le dernier chapitre, nous n'avons fait qu'aborder ce sujet en proposant d'abord un calcul des performances quand les chemins des flots sont fixés, puis traité une optimisation de ces flots en adaptant des techniques de flot dans les graphes, telles que l'algorithme de Dijkstra. De nombreux points n'ont pas encore été abordés et nous en citons ici quelques uns.

Nous savons optimiser un chemin pour un flots parmi un ensemble de flots. Il semble alors naturel d'essayer de faire une optimisation «globale» des flots : faire en sorte que les flots soient en un certain équilibre. C'est-à-dire, étant donné un ensemble de flots, modifier le chemin d'un flot ne peut pas améliorer les performances de ce flot si les autres flots ne changent pas. Une première piste pour étudier ce problème est d'utiliser des techniques d'équilibres de Nash et donc la recherche de points fixes pour le routage dans ces réseaux.

Un second point serait de trouver un bon ensemble d'angles à interdire. En effet, les heuristiques présentées pour la minimisation du nombre d'angles à interdire ne tiennent pas compte des capacités des serveurs et peuvent mener à une sous-utilisation des ressources les plus importantes. Cette problématique peut être abordée sous deux aspects différents : soit on connaît *a priori* les intensités des flots, leur source et leur destination, soit on ne connaît rien des flots. Dans ces deux cas, une manière d'aborder le problème est d'adapter les algorithmes connus de la théorie des graphes et de la théorie des flots en particulier. Les algorithmes de flot peuvent en effet s'interpréter comme un cas particulier où les capacités des serveurs sont constants et les délais de transmission immédiats. L'étude du network calculus dans les graphes peut donc être vue comme l'étude des flots dans les graphes avec des capacités non constantes.

Cette étude a aussi permis de mettre en évidence les difficultés à appréhender la dépendance des performances d'un flot en fonction d'autres flots dans le réseau. Les techniques utilisées ne sont pas très efficaces et montrent qu'il nous faut encore aller dans le sens d'une étude plus approfondie de l'étude des garanties de performances via le network calculus.

Bibliographie personnelle

[1] A. Bouillard and B. Gaujal. Coupling time of a (max,plus) matrix. In *Workshop on Max-Plus Algebras and Their Applications to Discrete-event Systems, Theoretical Computer Science, and Optimization*, pages 235–240. IFAC, 2001.

[2] A. Bouillard and B. Gaujal. Perfect sampling for fork-join networks. Technical Report RRLIP-2005-12, LIP, 2005.

[3] A. Bouillard, B. Gaujal, and J. Mairesse. Extremal throughputs in free-choice nets. À paraître dans *Journal of Discrete Event Dynamic Systems*.

[4] A. Bouillard, B Gaujal, and J. Mairesse. Throughput in stochastic free-choice nets. Accepté à 44th IEEE Conference on Decision and Control, 2005, Séville.

[5] A. Bouillard, B. Gaujal, and J. Mairesse. Extremal throughputs in free-choice nets. In G. Ciardo and P. Darondeau, editors, *26th International Conference On Application and Theory of Petri Nets and Other Models of Concurrency*, LNCS, pages 108–127. Springer-Verlag, 2005.

[6] A. Bouillard and J. Mairesse. Generating series of the trace group. In Z. Ésik and Z. Fülöp, editors, *Developments in Language Theory*, volume 2710 of *LNCS*, pages 159–170. Springer-Verlag, 2003.

[7] A. Bouillard and J. Mairesse. Möbius inversion formula for the trace group. *C. R. Acad. Sci. Paris, Ser I*, 339 :899–904, 2004.

Bibliographie

[1] R. Agrawal, F. Baccelli, and R. Rajan. An Algebra for Queueing Networks with Time Varying Service and its application to the analysis of integrated service networks. Technical Report 3435, INRIA, 1998.

[2] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan. Performance Bounds for Flow Control Protocols. *IEEE Transactions on Networking*, 7(3) :310–323, 1999.

[3] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* Parallel Computing. Wiley, 1995.

[4] M. Ajmone Marsan, A. Bobbio, and S. Donatelli. Petri nets in performance analysis : An introduction. In *Petri Nets*, pages 211–256, 1996.

[5] E. Altman, B. Gaujal, and A Hordijk. Admission control in stochastic event graphs. Technical Report RR-3179, INRIA, 1999.

[6] E. Altman, B. Gaujal, and A Hordijk. *Discrete-Event Control of Stochastic Networks : Multimodularity and Regularity*, volume 1829 of *Lecture Notes in Mathematics.* Springer, 2003.

[7] M. Andrews. Instability of fifo in session-oriented networks. *J. Algorithms*, 50(2) :232–245, 2004.

[8] F. Baccelli. Ergodic theory of stochastic Petri networks. *Ann. Probab.*, 20(1) :375–396, 1992.

[9] F. Baccelli and P. Brémaud. *Elements of queuing theory.* Springer, 1992.

[10] F. Baccelli, G. Cohen, and B. Gaujal. Recursive equations and basic properties of timed Petri nets. *J. of Discrete Event Dynamic Systems*, 1(4) :415–439, 1992.

[11] F. Baccelli, G. Cohen, Olsder G.J., and Quadrat J.-P. *Synchronization and Linearity.* Wiley, 1992.

[12] F. Baccelli and D. Hong. Analytic expansions of max-plus Lyapunov exponents. *Ann. Appl. Probab.*, 10(3) :779–827, 2000.

[13] F. Baccelli and D. Hong. TCP is max-plus linear : and what tells us on its throughput. In *SIGCOMM*, pages 219–230, 2000.

[14] F. Baccelli and Foss S. On the saturation rule for the stability of queues. *J.Appl.Probab.*, 32(2) :494–507, 1995.

[15] J. Banks and J.G. Dai. Simulation studies of multiclass queueing networks. *IIE Transactions*, 29 :213–219, 1997.

[16] D. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, 1995.

[17] J. Billington, M. Diaz, and G. Rozenberg (Eds.), editors. *Application of Petri Nets to Communication Networks*, volume 1605 of *Lecture Notes in Computer Science.* Springer-Verlag, 1999.

[18] A. Bouillard and B. Gaujal. Coupling time of a (max,plus) matrix. In *Workshop on Max-Plus Algebras and Their Applications to Discrete-event Systems, Theoretical Computer Science, and Optimization*, pages 235–240. IFAC, 2001.

[19] A. Bouillard and B. Gaujal. Perfect sampling for fork-join networks. Technical Report RRLIP-2005-12, LIP, 2005.

[20] A. Bouillard, B. Gaujal, and J. Mairesse. Extremal throughputs in free-choice nets. À paraître dans *Journal of Discrete Event Dynamic Systems*.

[21] A. Bouillard, B. Gaujal, and J. Mairesse. Throughput in stochastic free-choice nets. Accepté à 44th IEEE Conference on Decision and Control, 2005, Séville.

[22] A. Bouillard, B. Gaujal, and J. Mairesse. Extremal throughputs in free-choice nets. In G. Ciardo and P. Darondeau, editors, *26th International Conference On Application and Theory of Petri Nets and Other Models of Concurrency*, LNCS, pages 108–127. Springer-Verlag, 2005.

[23] A. Bouillard and J. Mairesse. Generating series of the trace group. In Z. Ésik and Z. Fülöp, editors, *Developments in Language Theory*, volume 2710 of *LNCS*, pages 159–170. Springer-Verlag, 2003.

[24] A. Bouillard and J. Mairesse. Möbius inversion formula for the trace group. *C. R. Acad. Sci. Paris, Ser I*, 339 :899–904, 2004.

[25] T. Bousch and J. Mairesse. Asymptotic height optimization for topical IFS, Tetris heaps, and the finiteness conjecture. *J. Am. Math. Soc.*, 15(1) :77–111, 2002.

[26] M. Brin and G. Stuck. *Introduction to dynamical systems*. Cambridge University Press, 2002.

[27] J. Carlier and P. Chretienne. Timed Petri net schedules. In *Advances in Petri Nets*, volume 340 of *LNCS*, pages 62–84. Springer, 1988.

[28] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrengements*. Number 85 in Lect. Notes in Mathematics. Springer Verlag, 1969.

[29] F. Cassez and O.(H.) Roux. Traduction structurelle des réseaux de Petri temporels vers les automates temporisés. In *4ieme Colloque Francophone sur la Modélisation des Systèmes Réactifs, (MSR'03)*, Metz, France, October 2003.

[30] C.-S. Chang. *Performances Guarantees in Communication Networks*. Springer Verlag, 2000.

[31] G. Cohen, S. Gaubert, and J.P. Quadrat. Asymptotic throughput of continuous timed Petri nets. In *Proc. of 34-th Conference on Decision and Controlin New-Orleans*, pages 2029–2034. IEEE, 1995.

[32] F. De Pellegrini, D. Starobinski, M. Karpovski, and L. Levitin. Scalable cycle-breaking algorithms for gigabit ethernet backbones. In *IEEE Infocom*, 2004.

[33] E. V. Denardo. Periods of connected networks and power of nonnegative matrices. *Mathematics of Operation Research*, 2(1) :20–24, 1977.

[34] J. Desel and J. Esparza. *Free choice Petri nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1995.

[35] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva, and F.B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman and Hall, 1993.

[36] V. Diekert. *Combinatorics on traces*. Number 454 in Lectures Notes in Computer Sciences. Springer-Verlag, 1990.

[37] V. Diekert and G. Rozenberg, editors. *The book of traces.* World Scientific, 1995.

[38] Volker Diekert and Anca Muscholl. Solvability of equations in free partially commutative groups is decidable. In *ICALP*, pages 543–554, 2001.

[39] C. Droms. Subgraphs of graph groups. *Journal of Algebra*, 110 :519–522, 1987.

[40] J. Esparza. Reachability in live and safe free-choice Petri nets is np-complete. *Theor. Comput. Sci.*, 198(1-2) :211–224, 1998.

[41] J. Esparza and M. Nielsen. Decidability issues for Petri nets - a survey. *Bulletin of the European Association for Theoretical Computer Science*, 52 :245–262, 1994.

[42] A. Di Febbraro, D. Giglio, R. Minciardi, and S. Sacone. Optimization of Manufacturing Systems Modelled by Timed Petri Nets. In *Sixth International Workshop on Discrete Event Systems (WODES'02)*, pages 361–367, 2002.

[43] M. Fidler and G. Einhoff. Routing in turn-prohibition based feed-forward networks. In *NETWORKING*, pages 1168–1179, 2004.

[44] P. Flajolet and R. Sedgewick. The average case analysis : complex asymptotics and generating functions. Technical Report 2026, Rapport de Recherche INRIA, 1993.

[45] É Fleury and P. Fraigniaud. A general theory for deadlock avoidance in wormhole-routed networks. *IEEE Trans. Parallel Distrib. Syst.*, 9(7) :626–638, 1998.

[46] S. Foss and R.L. Tweedie. Perfect simulation and backward coupling. *Stoch. Model*, 14(1-2) :187–204, 1998.

[47] M. Garey and D. Johnson. *Computers and intractability, a guide to the theory of NP-completeness.* Freeman, New York, 1979.

[48] S. Gaubert, J.-J. Loiseau, J. Mairesse, M. Nivat, and J.-É. Pin, editors. *Max-Plus algebras*, volume 293 of *Theoretical Computer Science.* Elsevier, 2003.

[49] S. Gaubert and J. Mairesse. Task resource models and (max,+) automata. In J. Gunawardena, editor, *Idempotency*, pages 133–144. Cambridge University Press, 1997.

[50] S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Autom. Control*, 44(4) :683–697, 1999.

[51] B. Gaujal and A. Giua. Optimal stationary behavior for a class of timed continuous Petri nets. *Automatica*, 40(9) :1505–1516, 2004.

[52] B. Gaujal, S. Haar, and J. Mairesse. Blocking a transition in a free choice net and what it tells about its throughput. *Journal of Computer and System Sciences*, 66 :515–548, 2003.

[53] C. Girault and R. Valk. *Petri nets for Systems Engineering : a guide to Modeling, Verification and applications.* Springer, 2003.

[54] C. Glass and L. Ni. The turn model for adaptive routing. *J. ACM*, 41(5) :874–902, 1994.

[55] G. Grimmett and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, 2001.

[56] T. Gu, P. Bahri, and G. Cai. Timed Petri-net based formulation and an algorithm for the optimal scheduling of batch plants. *J. Appl. Math. Comput. Sci.*, 13(4) :527–536, 2003.

[57] J. Gunawardena, editor. *Idempotency.* Cambridge University Press, 1998.

[58] S. Haddad, P. Moreaux, and G. Chiola. Efficient handling of phase-type distributions in generalized stochastic Petri nets. In *ICATPN*, pages 175–194, 1997.

[59] O. Häggström. *Finite Markov Chains and Algorithmic Applications.* Number 52 in Student texts. Cambridge University Press, 2002.

[60] P.J. Hass. *Stochastic Petri Nets : Modelling, Stability, Simulation.* Springer Series in Operation Research, 2002.

[61] H. P. Hillion and J.-M. Proth. Using timed Petri nets for the scheduling of job-shop systems. *Engineering Costs and Production Economics*, 17(1-4) :149–154, 1989. NewsletterInfo : 35.

[62] E. Hyon. *Contrôle d'admission en boucle ouverte dans les réseaux.* PhD thesis, LORIA, 2001.

[63] F. Kelly. *Reversibility and Stochastic Networks.* Wiley, Chichester, 1979.

[64] D. Krob, J. Mairesse, and I. Michos. Computing the average parallelism in trace monoids. *Discrete Math.*, 273 :131–162, 2003.

[65] J. Y. Le Boudec and P. Thiran. *Network Calculus.* Springer Verlag LNCS 2050, June 2001.

[66] L. Levitin, M. Karpovsky, M. Mustafa, and L. Zakrevski. A new algorithm for finding minimal cycle-breaking sets of turns in a graph. submitted to graph theory and application, 2005.

[67] R.M. Loynes. The stability of a queue with non-independent inter-arrival and service times. *Proceedings of the Cambridge Philosophical Society*, 58 :497–520, 1952.

[68] R. Lyons. Random walks and the growth of groups. *C. R. Acad. Sci., Paris, Sér. I*, 320(11) :1361–1366, 1995.

[69] J. Mairesse. Products of irreducible random matrices in the (max,+) algebra. *Advances in Applied Probability*, 29(2) :444–477, 1997.

[70] J. Mairesse and L. Vuillon. Asymptotic behavior in a heap model with two pieces. *Theoret. Comput. Sci.*, 270(1-2) :525–560, 2002.

[71] C.-M. Marle. *Systèmes dynamiques, une introduction.* Ellipse, 2003.

[72] T. Mulder and A. Storjohann. On lattice reduction for polynomial matrices. *Journal of symbolic computation*, 35 :277–401, 2003.

[73] A. Müller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks.* Willey, 2002.

[74] T. Murata. Petri nets : Properties, analysis and applications. *Proceedings of the IEEE*, April 1989.

[75] L. Ni and P. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26(2) :62–76, 1993.

[76] M.A. Novotny. Introduction to the Propp-Wilson method of exact sampling for the Ising model. *Computer Simulation Studies in Condensed Matter Physics XII*, pages 179–184, 2000.

[77] K. Pandit, C. Kirchner, J. Schmitt, and R. Steinmetz. Optimization of the Min-plus Convolution Computation under Network Calculus Constraints. Technical Report TR-KOM-2004-04, Technische Universität Darmstadt, June 2004.

[78] K. Pandit, J. Schmidt, C. Kirschner, and R. Steinmetz. Optimal allocation of service curves by exploiting properties of the min-plus convolution. Technical Report TR-KOM-2004-08, Technische Universität Darmstadt, december 2004.

[79] J.-É. Pin. Tropical semirings. In J. Gunawardena, editor, *Idempotency*, pages 50–69. Cambridge University Press, 1998.

[80] J. Propp and D. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9 :232–252, 1996.

[81] J. Propp and D. Wilson. Coupling from the past : a user's guide. In D. Aldous and J. Propp, editors, *Microsurveys in Discrete Probability*, volume 41 of *Series in Discrete Mathematics and Theoretical Computer Science*, pages 181–192. American Mathematical Society, 1998.

[82] L. Recalde and M. Silva. Petri net fluidification revisited : Semantics and steady state. *European Journal of Automation APII-JESA*, 35(4) :435–449, 2001.

[83] W. Reisig. *Petri nets : an introduction.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[84] J. Sakarovitch. *Éléments de théorie des automates.* Vuibert, 2003.

[85] J. Sandefur. *Discrete dynamical systems, theory and applications.* Oxford University Press, 1990.

[86] M. Silva and L. Recalde. Continuization of timed Petri nets : From performance evaluation to observation and control. In *ICATPN*, pages 26–47, 2005.

[87] D. Starobinski, M. Karpovsky, and L. Zakrevski. Application of network calculus to general topologies using turn-prohibition. *IEEE/ACM Transactions on Networking*, 11(3) :411–421, June 2003.

[88] L. Thiele. Real-time calculus. Technical report, Computer Engineering and Network Laboratory, ETH Zurich, Suisse, 2003.

[89] Patrick Thiran, Jean-Yves Le Boudec, and Frederic Worm. Network Calculus applied to Optimal Multimedia Smoothing. In *INFOCOM2001*, 2001.

[90] A.M. Vershik, S. Nechaev, and R. Bikbov. Statistical properties of locally free groups with applications to braid groups and growth of random heaps. *Communications in Mathematical Physics*, pages 469–501, 2000.

[91] G.-X. Viennot. Heaps of pieces, i : Basic definitions and combinatorial lemmas. In Labelle and Leroux, editors, *Combinatoire Énumérative*, number 1234 in Lect. Notes in Mathematics, pages 321–350. Springer, 1986.

[92] J.-M. Vincent and C. Marchand. On the exact simulation of functionals of stationary Markov chains. *Linear Algebra and Its Applications*, 386 :285–310, 2004.

[93] D. West. *Introduction to graph theory.* Prentice Hall, 2 edition, 2001.

Résumé

L'objet de cette thèse est l'étude des performances dans les systèmes à événements discrets. Trois modèles différents de ces systèmes sont étudiés.

Dans une première partie, nous nous intéressons aux groupes de traces. Après avoir donné une formule simple, semblable à la formule de Möbius pour les monoïdes de traces, de la série génératrice des groupes de traces, nous montrons l'existence et l'algébricité du taux de croissance asymptotique de la hauteur des traces.

Une deuxième partie de cette thèse est consacrée aux réseaux à choix libres temporisés, une sous-classe importante de réseaux de Petri. Nous définissons une notion de débit dans ces réseaux et étudions la variation du débit en fonction de la politique de résolution de conflits choisie. Tout d'abord, nous montrons comment calculer ce débit. Ensuite, nous nous intéressons aux politiques qui le maximisent ou le minimisent. Enfin, nous donnons une méthode efficace de simulation exacte du processus des marquages pour l'évaluation numérique du débit.

Enfin, nous étudions le calcul des garanties de performances dans les réseaux à l'aide des techniques du *Network Calculus*. Nous montrons la stabilité des fonctions ultimement pseudo-périodiques par les opérations du network calculus et donnons des algorithmes pour le calcul de ces fonctions. Ces techniques sont ensuite appliquées à l'étude des garanties de performances dans les graphes avec angles interdits.

Mots-clés: Systèmes à événements discrets, groupe de traces, réseaux de Petri temporisés, Network Calculus, évaluation de performances

Abstract

This thesis deals with the study of discrete event systems. Three different models are considered.

In the first part, we are interested in the trace groups. After giving a simple Möbius-like formula for the generating series of the trace groups, we show the existence and the algebraicity of the asymptotic growth rate of the height of the traces.

The second part is devoted to timed free-choice nets, an important sub-class of Petri nets. We define the notion of throughput in those nets and study the variation of the throughput in function of the conflict resolution policies. First, we show how to compute the throughput, then we are interested in the policy that maximizes or minimizes the throughput. Finally, we give an efficient method to generate a marking according to its exact distribution in order to numerically evaluate the throughput.

In the last part, we study the computation of performance guarantees in networks thanks to Network Calculus techniques. We show the stability of the ultimately pseudo-periodic functions with the operations of the Network Calculus and give algorithms to compute these functions. These techniques are then applied to the study of performance guarantees in graphs with turn prohibition.

Keywords: Discrete event systems, trace group, timed Petri nets, Network Calculus, performance evaluation