

THÈSE
PRÉSENTÉE PAR

Matthieu BRILMAN

Pour obtenir le titre de docteur
de l'Université Joseph Fourier - Grenoble I
(Arrêtés ministériels du 05/07/84 et du 30/03/92)

Spécialité Informatique

Évaluation de Performances d'une Classe de
Systèmes de Ressources Partagées

soutenue le 30 Septembre 1996 devant un jury composé de

Jacques Mazoyer	Président
François Baccelli	Rapporteur
Philippe Robert	Rapporteur
Marcel Neuts	Examineur
Brigitte Plateau	Examineur
Jean-Marc Vincent	Directeur
Bernard Ycart	Examineur

Je tiens à remercier tout d'abord Jacques Mazoyer pour l'intérêt qu'il a porté à mon travail et pour m'avoir fait l'honneur de présider mon jury.

Je remercie aussi tout particulièrement François Baccelli d'avoir accepté de rapporter sur cette thèse, bien sûr, mais aussi de m'avoir accueilli dans son équipe. Les deux séjours que j'y ai passés ont grandement contribué à l'élaboration de cette thèse.

Je remercie également Philippe Robert d'avoir pris le temps de rapporter sur ma thèse.

Merci à Bernard Ycart pour le soin avec lequel il a relu mon manuscrit. Ses commentaires ont largement profité à la présentation de celui-ci.

Marcel Neuts a accepté de participer à mon jury, je l'en remercie vivement.

Merci à Brigitte Plateau pour sa bonne humeur et ses conseils avisés qui font d'elle une vraie chef!

Merci à Jean-Marc Vincent qui a accepté d'encadrer ma thèse. Ses encouragements et ses critiques m'ont aiguillé au cours de ces trois années.

Cette thèse a grandement profité de collaborations avec l'INRIA-Sophia. Celles-ci n'auraient pas été possibles sans le financement du projet AQSI. Parmi les personnes ayant pris part à ces collaborations, je tiens particulièrement à remercier Jean Mairesse, Bruno Gaujal et Alain Jean-Marie. Leur accueil chaleureux, la disponibilité dont ils ont fait preuve, tant pour confronter nos points de vue que pour relire et commenter mes travaux, ont rendu celles-ci agréables et fructueuses.

Je tiens enfin à remercier les membres des équipes de calcul formel et d'algorithmique parallèle pour l'ambiance agréable du labo et Éric en particulier pour les nombreuses fois où il m'a "dépanné" (Dis, Éric, comment on fait pour...).

Merci aussi à Denis Vandorpe et à toute l'équipe du LIGIM qui m'a accueilli dans ses locaux pendant rédaction du manuscrit.

Enfin, merci à Christophe, Krystof, Jean-Nicolas, Joëlle (bien sûr) et aux autres, qui m'ont soutenu tout au cours de cette thèse, allant même jusqu'à supporter mes discours sur γ !

Table des matières

1	Présentation des systèmes de ressources partagées	9
1.1	Dynamique	13
1.2	Débit asymptotique de systèmes saturés	14
1.3	Plan de cette thèse	14
2	Modélisation	17
2.1	Modélisation et équations d'évolution	17
2.1.1	Un exemple de <i>système de ressources</i>	18
2.1.2	Présentation formelle des systèmes de ressources	19
2.1.3	Définition d'un <i>système de requêtes</i>	21
2.1.4	Équivalence entre ces deux modèles	24
2.1.5	Équivalence avec un modèle de files d'attente	28
2.2	Récriture dans la pseudo-algèbre $(\max, +)$	29
2.2.1	Présentation rapide de $(\max, +)$	30
2.2.2	Récriture des équations d'évolution dans $(\max, +)$	31
2.2.3	Résultats apportés par $(\max, +)$. Cas particulier des systèmes périodiques	32

TABLE DES MATIÈRES

2.3	Processus induits	32
2.3.1	Représentation graphique	33
2.3.2	Lignes et motifs	34
3	Systèmes indépendants	39
3.1	Présentation des systèmes indépendants	39
3.1.1	Définitions	40
3.1.2	Résultats apportés par $(\max, +)$ aux systèmes stochastiques	41
3.1.3	Simulation de systèmes indépendants	46
3.2	Espace projectif	47
3.2.1	Une remarque simple	48
3.2.2	Chaînes de Markov $\mathcal{M}(\text{Res}, \mathbf{S})$ et $\mathcal{M}(\text{Req}, \mathbf{S})$	48
3.2.3	Un théorème central limite	52
3.2.4	Cas d'un système comportant exactement deux ressources	54
3.2.5	Bornes supérieures obtenues par troncature	57
3.3	Notion de ligne asymptotique	60
3.4	Un théorème de composition	65
3.4.1	Présentation du théorème	65
3.4.2	Application à une famille de systèmes particuliers	69
3.5	Propriétés analytiques. Réduction à des systèmes uniformes	71
3.5.1	Propriétés analytiques	71
3.5.2	Restriction à des mesures uniformes	77

TABLE DES MATIÈRES

3.6	Systèmes uniformes	80
3.6.1	Une borne basée sur le rayon spectral de \mathcal{G}	81
3.6.2	Une borne inférieure fonction du degré moyen de \mathcal{G}	85
3.6.3	Relation entre $\gamma(\mathcal{G})$ et $\chi(\mathcal{G})$	88
3.6.4	Quelques exemples	91
4	Études asymptotiques	93
4.1	Rappel des bornes. Cas de graphes de degré constant	93
4.2	Comparaison de deux familles de systèmes de ressources	94
4.3	Graphes étoiles	97
4.3.1	Expression intégrale de $\gamma(E_n)$	97
4.3.2	Comportement asymptotique de $\gamma(E_n)$	98
5	Conclusion	101
A	Applications	103
A.1	Stabilité d'un système de ressources partagées	103
A.2	Application à $(\max, +)$	105
A.3	Table des notations	108

TABLE DES MATIÈRES

Chapitre 1

Présentation des systèmes de ressources partagées

L'étude des systèmes de ressources partagées revêt une importance croissante en informatique. Par "système de ressources partagées" on entend tout système, généralement informatique, composé de *ressources* que doivent se partager des *utilisateurs*. Le partage des ressources peut avoir plusieurs raisons : il peut résulter de contraintes de "coût". Par exemple, dans un réseau de communication, un même lien pourra avoir à servir plusieurs demandes de communication. Il serait possible de rajouter des liens au réseau de façon à ce que chaque communication possible ait ses liens réservés, mais le "coût" d'une telle opération serait prohibitif. Pour cette raison, les liens du réseau seront le plus souvent partagés entre les différents usagers. Le partage de ressources peut aussi avoir pour origine l'impossibilité de répliquer chacune des ressources. Un disque dur, par exemple, est une ressource partagée entre les processus d'un système d'exploitation dont la duplication n'a pas d'intérêt réel : on ne va pas allouer à chacun des processus différents un disque dur différent !

L'étude des systèmes de ressources partagées s'est donc développée en même temps que se sont développés ces systèmes. Dans un premier temps, cette étude a surtout été consacrée à la recherche de politiques de gestion des ressources garantissant que soient préservées certaines propriétés logiques de *cohérence* afin de pouvoir assurer la *fiabilité* des systèmes en question. Par exemple, le problème de l'exclusion mutuelle (on veut garantir que l'accès à une ressource donnée est limité à un seul utilisateur à la fois) a été résolu par l'introduction de *sémaphores* [Dijkstra68b]. D'autres problèmes, tels que la prévention d'*interbloquages* (situation dans laquelle deux utilisateurs s'empêchent mutuellement d'accéder à des ressources) ou de famines (attente perpétuelle des ressources qu'il a demandées par un utilisateur) ont été étudiés. On a donc défini des *politiques d'ordonnement* qui garantissent que ces propriétés sont préservées. La définition générique de ces politiques

a nécessité l'introduction de *modèles*.

Un modèle est une abstraction du système réel, dont on pense qu'elle capture les aspects les plus importants, tout en étant suffisamment simple pour se prêter à un traitement théorique. Son intérêt réside aussi dans le fait qu'il est souvent assez générique : toute une classe de systèmes sera représentée par un même modèle. Il doit donc présenter un compromis entre la complexité du système réel dont l'analyse est souvent difficile et la simplicité d'un modèle trop abstrait d'analyse plus facile. Les problèmes étudiés étant de nature logique (dans le cas de l'interblocage, on cherche à garantir que le système reste dans un ensemble d'états consistants, pour le cas de la famine, on cherche à garantir la prévention de famine pour toute exécution du système), les modèles proposés étaient de nature uniquement logique. On peut par exemple citer les graphes d'exclusion [Chandy et al.88] et les réseaux de Petri [Murata89].

Après l'analyse logique des systèmes de ressources partagées est naturellement apparu le besoin d'une analyse plus "dynamique" : comment ces systèmes se comportent-ils au cours du temps ? C'est le problème de l'évaluation de performances. Cette évaluation *a priori* est fondamentale : elle permet de mesurer le coût de telle ou telle politique d'ordonnement. En effet, la garantie de certaines propriétés comme par exemple l'absence de famine, est généralement coûteuse au niveau des performances du système. Par "performances", on entend tout critère de "qualité". Parmi les principaux critères généralement retenus figurent le débit du système (le nombre d'utilisateurs dont la demande est satisfaite par unité de temps), l'insensibilité du système à des modifications de son environnement, le temps moyen qu'un utilisateur doit attendre les ressources qu'il a demandées lorsque celles-ci ne sont pas libres, ou encore la probabilité qu'un utilisateur ne puisse obtenir les ressources qu'il a demandées en un temps fixé. Ce dernier critère est particulièrement important dans les systèmes dits "temps réel". L'évaluation de performances sur un modèle est donc fondamentale, en ce sens qu'elle permet, à moindre coût, de prédire la dynamique d'un système réel.

Afin de tenir compte de la nature imprévisible du comportement des utilisateurs du système, de pouvoir caractériser des comportements moyens, il a été nécessaire d'introduire des hypothèses probabilistes. Parmi les principaux modèles utilisés en évaluation de performances de systèmes de ressources partagées figurent les réseaux de Petri temporisés [Marsan et al.86] (les temporisations, associées aux hypothèses probabilistes, permettent de décrire le comportement du système au cours du temps) ou les réseaux de files d'attente avec synchronisations [Baccelli et al.89]. Là encore, deux approches sont possibles : une approche qualitative; on va par exemple chercher à montrer la stabilisation d'un système au cours du temps, sa tolérance aux pannes. Les hypothèses correspondantes sont des hypothèses très générales, souvent de type ergodiques. Une autre approche possible est une approche quantitative; on va s'intéresser à évaluer le débit du système, la probabilité de rejet d'un utilisateur... Les hypothèses probabilistes correspondant à ce deuxième type de modèles sont nécessairement plus contraignantes : on doit quantifier les

processus intervenant dans la dynamique des systèmes de réservation. Afin d'avoir une description simple de tels processus on aura souvent besoin d'hypothèses d'indépendance de variables aléatoires. Ces hypothèses font que le modèle étudié est plus abstrait sans pour autant le réduire au point qu'il ne soit plus pertinent.

Cet aspect quantitatif de l'évaluation de performances des systèmes de ressources partagées connaît actuellement un fort développement. En effet, peu de résultats analytiques sont connus autrement que pour des systèmes gérés par des politiques d'ordonnancement ne préservant pas des propriétés telles que l'absence de famine ou l'équité (répartition équitable des ressources entre les utilisateurs). Ceci est dû au fait que les politiques d'ordonnancement préservant de telles propriétés présentent de réelles difficultés intrinsèques. Nous reviendrons sur ce point plus tard.

Le but de cette thèse est donc de présenter un modèle général de systèmes de ressources partagées garantissant l'absence de famine et l'équité, de définir, pour un tel modèle, un paramètre de performance fondamental, et enfin, de proposer des méthodes de faible complexité algorithmique permettant d'encadrer ce paramètre de performance.

Nous allons maintenant présenter quelques classes de systèmes de ressources partagées. Nous dégagerons ensuite des points communs à tous ces systèmes et présenterons de manière informelle leur dynamique commune, que nous avons pris comme point de départ de cette thèse. Nous introduirons ensuite la notion de codébit d'un système de ressources partagées. Ce paramètre de performance permet en effet de retrouver simplement la plupart des autres paramètres de performances, comme par exemple le taux d'utilisation des ressources ou le temps moyen de réponse. Nous présenterons ensuite le plan de cette thèse.

Exemple 1 : Considérons une machine parallèle avec une mémoire partagée. Cette mémoire est partitionnée en plusieurs régions. Un processus peut parfois avoir besoin d'écrire simultanément dans plusieurs régions. Afin d'éviter de possibles conflits, il est nécessaire que l'accès à ces régions soit exclusif. Autrement dit, ce processus va verrouiller les régions dont il a besoin pendant un certain temps, après quoi il les relâchera. Afin d'avoir un service équitable, un processus aura accès aux régions qu'il demande une fois celles-ci libérées par les processus précédemment arrivés dans le système. Ce contrôle peut être mis en place à l'aide d'une file d'attente globale ou distribuée. Dans un tel système, les régions peuvent être vues comme des ressources et les demandes d'accès aux régions par les processus comme des requêtes. Une fois qu'une requête est arrivée dans le système, elle est servie dès que les ressources dont elle a besoin ont toutes été libérées par les requêtes précédemment arrivées dans le système.

Exemple 2 : Considérons plusieurs entités reliées par un réseau de communication. La figure 1.1 en donne un exemple. Si deux entités veulent communiquer entre elles, elles vont devoir réserver un chemin (c'est-à-dire un ensemble de nœuds du réseau) qu'elles seront seules à utiliser. Si tous les nœuds du chemin sont libres au moment où les entités

veulent communiquer, la communication commence, sinon la demande de communication est stockée dans une file d'attente. Dans une machine parallèle à mémoire distribuée, ce type de comportement correspond à un mode de communication par commutation de circuits en faisant l'hypothèse que le temps de la communication est grand devant le *startup* [Kermani et al.79]. Dans un tel système, les demandes de communication peuvent être vues comme des requêtes et les nœuds comme des ressources.

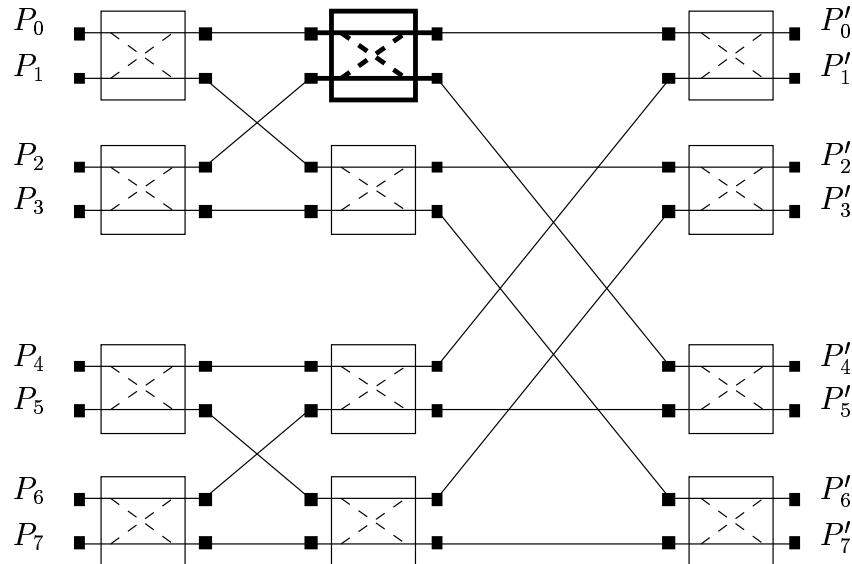


FIG. 1.1 – Un réseau de communication. Sur cet exemple, une communication de P_0 vers P'_0 (première requête) et une communication de P_2 vers P'_4 (deuxième requête) sont concurrentes, puisqu'elles réclament chacune le nœud (ressource) marqué en gras.

Exemple 3 : Une base de données est constituée d'entités. Des transactions demandent chacune l'accès à certaines de ces entités. Afin de préserver la cohérence de la base de données, il est nécessaire d'interdire l'accès concurrent en écriture à une même entité par deux transactions. Afin de garantir le service de chaque transaction (une transaction correspondant à un besoin réel, il n'est guère raisonnable de la rejeter lorsqu'elle ne peut être exécutée immédiatement), une file d'attente est mise en place afin de gérer l'accès aux entités par des transactions concurrentes. Les transactions ayant le plus souvent besoin de lire les valeurs de certaines entités et de modifier les valeurs d'autres entités en fonction des valeurs qu'elles ont lues, la prise des ressources devra se faire de façon synchrone. On retrouve donc les dynamiques présentées précédemment, en identifiant les transactions à des requêtes et les entités à des ressources. Nous reviendrons plus précisément sur le cas des bases de données dans la partie 2.1.1.

1.1 Dynamique

La dynamique des systèmes de ressources partagées que l'on considère peut se résumer de la façon suivante :

Un système est composé d'un nombre fini de ressources auxquelles des requêtes demandent l'accès à diverses dates. Chaque requête va demander l'accès à un sous-ensemble de ressources. L'accès à une ressource est exclusif, ce qui implique que deux requêtes demandant une même ressource ne pourront pas être servies simultanément. Par contre il est possible d'exécuter en parallèle deux requêtes demandant des ensembles de ressources disjoints. Afin d'avoir un service équitable, une requête ne commence à bloquer les ressources dont elle a besoin qu'à partir du moment où celles-ci ont toutes été libérées par l'ensemble des requêtes arrivées précédemment dans le système. La prise des ressources par une requête se fait donc de façon synchrone.

L'équité et l'absence de famine dans un tel système sont garanties par la présence d'une file d'attente. Cette file d'attente apporte au réalisme du modèle. Par contre, elle le rend plus complexe. Les modèles avec rejet, c'est-à-dire sans file d'attente et dans lesquels une requête ne pouvant pas être servie est rejetée ont beaucoup été étudiés [Kelly85] [Mitra85] [Forbes et al.96]. La plupart des résultats sur ces modèles s'appuient sur la réversibilité dans le temps de processus décrivant l'évolution du système. L'apparition d'une file d'attente introduit des synchronisations entre processus et fait disparaître cette réversibilité. C'est ce qui en rend le traitement difficile.

Présentons maintenant deux hypothèses fondamentales que nous ferons sur les systèmes de ressources partagées étudiés dans cette thèse.

Hypothèse de “temps constant” : Nous faisons l'hypothèse que le temps pendant lequel une requête bloque les ressources dont elle a besoin est toujours le même, égal à une unité de temps. Cette hypothèse est assez restrictive. Cependant, on verra que, même avec cette hypothèse, les systèmes de ressources partagées que nous considérons sont très complexes. Par ailleurs, il existe un théorème de comparaison simple entre les systèmes pour lesquels le temps de prise des ressources est variable et ceux pour lesquels il est constant [Brilman et al.95b]. On peut remarquer que le fait qu'une requête ait un temps d'exécution unitaire et que le nombre de ressources soit fini implique qu'il y a un nombre fini de requêtes différentes possibles.

Hypothèse de connexité : Une autre hypothèse que nous faisons est que les systèmes que nous considérons sont connexes, *i.e.* qu'il n'est pas possible de partitionner l'ensemble des requêtes possibles en ensembles de requêtes disjoints demandant l'accès à des ensembles de ressources disjoints. Cette hypothèse est très naturelle : l'étude d'un système non connexe peut se ramener à l'étude de chacune de ses composantes connexes.

1.2 Débit asymptotique de systèmes saturés

Nous avons limité notre étude des systèmes de ressources partagées aux seuls systèmes saturés : un système saturé est un système de ressources partagées tel que toutes les requêtes sont présentes dès le départ dans sa file d'attente (supposée alors infinie). Pourquoi nous être ainsi limités à l'étude des systèmes saturés ? La raison en est la suivante : on peut définir pour un système de ressources partagées plusieurs paramètres de performance. Parmi ceux-ci figurent le taux d'utilisation de chacune de ses ressources, le nombre moyen de requêtes traitées par unité de temps, son débit ou encore la probabilité de perte d'une requête (une requête est perdue si elle arrive au système alors que la file d'attente de celui-ci, supposée alors de capacité finie, est pleine). L'estimation de ces paramètres peut se ramener, au moins dans le modèle que nous avons considéré, à l'étude d'un seul paramètre sur la version saturée du système en question : son débit asymptotique. Il est donc particulièrement intéressant d'étudier le débit asymptotique de systèmes de ressources partagées saturés.

Comme, d'une façon générale, c'est plus souvent l'inverse du débit que le débit lui-même qui a été étudié, nous nous sommes intéressés à l'inverse du débit (que nous appelons codébit) de systèmes de ressources partagées saturés. Étant donnée l'hypothèse que les temps de prise des ressources par les requêtes sont tous unitaires, on peut remarquer que ce codébit est donc compris entre 0 et 1.

D'un point de vue formel, on retrouve dans divers domaines scientifiques des quantités très semblables au codébit. Celles-ci portent alors des noms différents tels que "temps de cycle" dans le cadre de l'étude des réseaux de Petri temporisés, "exposant de Lyapunov" pour les systèmes dynamiques ou bien encore "gain moyen" en économie.

1.3 Plan de cette thèse

Le but de cette thèse est donc, une fois défini de manière plus formelle le modèle sur lequel nous avons travaillé, l'évaluation du codébit de systèmes de ressources partagées.

Le chapitre 2 est consacré à la présentation de deux modèles de systèmes de ressources partagées. Le premier de ces deux modèles est assez naturel et permet une compréhension intuitive de certains phénomènes de synchronisations. Le deuxième de ces modèles, moins naturel, se prête mieux à un traitement théorique. Nous prouvons l'équivalence de ces modèles après quoi nous montrons leur équivalence avec un modèle particulier de files d'attente.

Dans le chapitre 3, nous posons des hypothèses probabilistes sur les systèmes que

1.3 Plan de cette thèse

nous étudions. Nous commençons par donner quelques propriétés du codébit pour de tels systèmes, après quoi nous présentons plusieurs approches permettant d'avoir des calculs exacts du codébit de certains systèmes simples et des bornes dans le cas général.

Le chapitre 4 est consacré à l'étude de "grands" systèmes sur lesquels on peut appliquer des résultats asymptotiques.

Chapitre 2

Modélisation

Nous présentons dans ce chapitre le modèle sur lequel nous avons travaillé. La partie 2.1 est consacrée à une traduction en termes mathématiques de la dynamique présentée dans le chapitre précédent. Deux points de vue ont été adoptés : le premier est centré sur les ressources des systèmes considérés ; le deuxième sur leurs requêtes. Nous montrons comment ces deux approches étaient déjà apparues dans la modélisation de systèmes de ressources partagées, puis les détaillons de façon plus formelle. Nous prouvons ensuite leur équivalence, entre elles d'une part, et avec un modèle de files d'attente particulier d'autre part. La partie 2.2 établit le lien entre notre problématique et la pseudo-algèbre $(\max, +)$. Ce lien nous permettra d'utiliser sans avoir à les démontrer des résultats connus dans $(\max, +)$. Enfin, la partie 2.3 est consacrée à la présentation de notions importantes intervenant dans l'étude des systèmes de ressources partagées.

2.1 Modélisation et équations d'évolution

Les systèmes de ressources partagées que l'on considère sont définis par un ensemble fini de ressources qui peuvent être réclamées par plusieurs types de requêtes à diverses dates. Cependant, comme il est clair que le débit asymptotique d'un tel système dépend de l'ordre d'arrivée des requêtes, il va nous falloir inclure dans la définition d'un *système* cette suite de requêtes. Nous allons voir dans la suite de cette partie deux façons équivalentes de définir un tel système. La première est centrée sur les ressources, et est sans doute la plus naturelle. On appellera un système défini de cette manière un *système de ressources*. La deuxième est centrée sur les requêtes. On appellera donc un système défini de cette seconde manière un *système de requêtes*. Chacune de ces deux approches a ses avantages et ses inconvénients. Elles ont, par contre, toutes deux une formulation équivalente dans la pseudo-algèbre $(\max, +)$ que nous présenterons dans la partie 2.2.

2.1.1 Un exemple de *système de ressources*

Parmi les premiers systèmes de ressources partagées étudiés figurent les bases de données. Nous présentons donc ici un modèle classique de bases de données [Mitra et al.84] [Mitra85]. D'un point de vue structurel, ce modèle est celui des systèmes de ressources que nous présentons dans la partie suivante. Il diffère cependant de ce dernier par l'absence de file d'attente. Nous discuterons donc brièvement de la difficulté qu'induit une telle file sur le traitement mathématique.

Une base de données est composée de N entités. Des transactions, dont l'action consiste à lire et/ou modifier les valeurs de certaines entités, sont émises au cours du temps. Afin de tirer parti du parallélisme inhérent à la structure répartie de la base de données, il est intéressant de chercher à exécuter ces transactions de façon non séquentielle. Cependant, afin de préserver la *cohérence* de la base de données, il faut veiller à ce que deux transactions en conflit sur une même entité ne s'exécutent pas en même temps. C'est dans ce but qu'a été introduite la notion de verrou : une transaction verrouille les entités dont elle a besoin. Une entité ne pouvant être verrouillée par plus d'une transaction, on garantit que deux transactions servies en même temps demandent l'accès à des ensembles d'entités disjoints. Le problème d'une telle politique est qu'elle interdit à plusieurs transactions de lire (sans écrire) en même temps une même entité. Afin de pallier cet inconvénient, il est possible d'introduire la notion de verrou non exclusif : un nombre illimité de verrous non exclusifs peuvent être posés sur une même entité ; par contre il n'est pas possible de poser sur une même entité un verrou exclusif et un verrou non exclusif. Plus formellement si à une date t , l'ensemble des entités verrouillées de façon exclusive dans la base de données est B_w tandis que celui des entités verrouillées de façon non exclusive est B_r , une transaction demandant à poser des verrous exclusifs sur un ensemble G_w et des verrous non exclusifs sur un ensemble G_r pourra être servie à la date t si et seulement si

$$B_w \cap G_w = \emptyset, B_r \cap G_w = \emptyset, B_w \cap G_r = \emptyset.$$

L'introduction de verrous non exclusifs permet un gain de parallélisme dans l'exécution des transactions. Il est cependant possible de constater que, d'un point de vue formel, l'étude de la dynamique d'une base de données \mathbf{B}_1 avec de tels verrous peut se ramener à l'étude d'une base de données \mathbf{B}_2 dans laquelle les verrous sont tous exclusifs : en effet, en définissant G comme l'ensemble des entités de \mathbf{B}_1 et R comme l'ensemble des transactions possibles dans \mathbf{B}_1 , il suffit de définir \mathbf{B}_2 comme étant composée des entités $G \times R$, les transactions de \mathbf{B}_2 étant obtenues à partir des transactions de \mathbf{B}_1 comme suit : une transaction de type $r \in R$ dans \mathbf{B}_1 , demandant un accès exclusif aux entités G_w et un accès non exclusif aux entités G_r , donne lieu dans \mathbf{B}_2 à une transaction demandant un accès exclusif aux entités appartenant à $(G_w \times R) \cup (G_r \times \{r\})$. Deux remarques sont à faire cependant : d'une part, cette équivalence nécessite que toutes les transactions de \mathbf{B}_1 demandent un accès exclusif à au moins une entité. Si tel n'est pas le cas, une transaction

ne demandant que des verrous non exclusifs dans \mathbf{B}_1 et n'étant donc pas en exclusion avec elle-même, devient dans \mathbf{B}_2 une transaction en exclusion avec elle-même. Cette hypothèse est couramment faite [Mitra85]. En effet, elle permet notamment de borner le nombre de transactions servies en même temps dans la base de données. La deuxième remarque est que le modèle de *systèmes de requêtes* que nous introduisons dans la partie 2.1.3 permet de tenir compte des verrous non exclusifs de façon très simple, sans même avoir à passer à un système équivalent. Nous reviendrons sur ce point dans la partie 2.1.3.

Le comportement dynamique de telles bases de données est le plus souvent modélisé comme suit [Kelly85] : le processus d'arrivée des transactions est supposé Poissonnien, les temps de services des transactions indépendants et identiquement distribués. Enfin, et c'est la différence fondamentale avec le modèle que nous avons étudié : les transactions ne pouvant être exécutées sont rejetées et *perdues*, en ce sens qu'elles ne sont pas ressoumises à la base de données. Cette hypothèse est très utile d'un point de vue mathématique : l'absence de file d'attente amène des propriétés de perte de mémoire et fait par là même apparaître des processus markoviens sous-jacents. Cependant, d'un point de vue pratique, elle n'est pas satisfaisante : une transaction ne pouvant être exécutée à une certaine date ne peut simplement être mise de côté. Imaginons par exemple une base de données constituée de N entités et comportant $N + 1$ types de transactions possibles : N transactions requérant chacune une entité différente et une $(N + 1)$ -ème transaction requérant l'ensemble des N entités. On suppose que N est relativement grand. Pour peu que l'intensité des arrivées de transactions soit suffisamment faible, on va pouvoir garantir que la plupart des transactions seront servies. Ceci est vrai pour l'ensemble des transactions. Le problème est que les transactions demandant un accès aux N entités seront par contre la plupart du temps rejetées : il suffit, pour qu'une telle transaction soit rejetée qu'une des entités soit déjà verrouillée, ce qui reste très probable. On voit donc que pour un tel système, l'absence de file d'attente fait apparaître un phénomène de famine pour les requêtes demandant un trop grand nombre de ressources. Cette difficulté posée par les files d'attente a souvent été mise de côté : [Morris et al.84] et [Mitra et al.84] proposent des heuristiques permettant d'essayer de tenir compte d'une telle file d'attente. Cependant, ces heuristiques n'ont pas de réelle justification théorique. Par ailleurs, il n'est pas possible de quantifier l'erreur qu'elles induisent.

2.1.2 Présentation formelle des systèmes de ressources

Nous présentons dans cette partie le modèle de systèmes de ressources que nous avons étudié de façon plus formelle.

Un système de ressources \mathbf{S} est la donnée d'une structure logique :

- un ensemble de K ressources :

$$\text{Res}(\mathbf{S}) = \{r_i(\mathbf{S}), 1 \leq i \leq K\};$$

- un ensemble de N requêtes possibles :

$$\text{Req}(\mathbf{S}) = \{v_i(\mathbf{S}) \in \mathcal{P}(\text{Res}(\mathbf{S})), 1 \leq i \leq N\};$$

et d'une dynamique :

- la suite des requêtes :

$$s(\mathbf{S}) = \{s(\mathbf{S}, n) \in \text{Req}(\mathbf{S}), n \geq 1\}.$$

Remarque 1 : On appellera *requête vide* un élément de $\text{Req}(\mathbf{S})$ égal à \emptyset . Une requête vide correspond donc à une requête ne demandant aucune ressource. Nous verrons par la suite que considérer de telles requêtes peut parfois être pratique.

Remarque 2 : Lorsqu'il n'y a pas d'ambiguïté sur le système considéré, on omettra le paramètre \mathbf{S} dans les notations précédentes.

Remarque 3 : La structure logique d'un système de ressources peut être représentée de manière naturelle par un hypergraphe [Berge87] dont les sommets correspondent aux ressources du système et les arêtes aux requêtes (qui ne sont autres que des *ensembles* de ressources). Pour le cas particulier d'un système de ressources tel que chaque requête demande exactement deux ressources, cet hypergraphe est donc simplement un graphe.

Afin de définir le codébit, c'est-à-dire l'inverse du débit, d'un système de ressources \mathbf{S} , on introduit des dateurs $\{T_i(\mathbf{S}, n)\}_{1 \leq i \leq K, n \geq 0}$ à valeurs entières, $T_i(\mathbf{S}, n)$ représentant la date de libération de la ressource r_i une fois traitées les n premières requêtes de \mathbf{S} . Par convention, l'origine des dates étant 0, $T_i(0) = 0$ pour tout i . Ces dateurs vérifient le système d'équations suivant :

$$T_i(n+1) = \begin{cases} T_i(n) & \text{si } r_i \notin s(n+1), \\ 1 + \max_{j, r_j \in s(n+1)} T_j(n) & \text{si } r_i \in s(n+1). \end{cases} \quad (2.1)$$

Ces équations, appelées équations d'évolution de \mathbf{S} , sont une traduction de la dynamique présentée dans la partie 1.1 : une requête est exécutée dès que les ressources dont elle a besoin ont été libérées par les requêtes arrivées précédemment dans le système. On note

$$\mathbf{T}(\mathbf{S}, n) \stackrel{\text{def}}{=} (T_1(\mathbf{S}, n), \dots, T_K(\mathbf{S}, n))$$

la variable d'état décrivant l'évolution de \mathbf{S} . Il est clair que

$$h(\mathbf{S}, n) \stackrel{\text{def}}{=} |\mathbf{T}(\mathbf{S}, n)|_\infty,$$

où $|x|_\infty = \max_i |x_i|$, représente la date de libération de l'ensemble des K ressources après que les n premières requêtes aient été traitées. En d'autres termes, $h(n)$ représente le temps nécessaire au traitement des n premières requêtes du système. On définit donc le codébit de \mathbf{S} :

$$\gamma(\mathbf{S}) \stackrel{\text{def}}{=} \lim_{n \rightarrow \infty} \frac{h(\mathbf{S}, n)}{n}, \text{ si cette limite existe.}$$

On donnera dans la partie 3.1 des conditions d'existence de γ sous des hypothèses probabilistes.

2.1.3 Définition d'un *système de requêtes*

On définit maintenant ce qu'est un système de requêtes. On montrera par la suite l'équivalence entre *système de ressources* et *système de requêtes*.

A la différence de celle des systèmes de ressources, la structure logique des systèmes de requêtes est définie à partir de la relation d'exclusion entre les requêtes qui composent le système. Nous représentons celle-ci par un *graphe d'exclusion*. Cette approche a tout d'abord été introduite dans [Chandy et al.84] comme une généralisation du paradigme des "dining philosophers" [Dijkstra68a]. Cependant, un tel modèle a surtout été étudié d'un point de vue logique et non dynamique : la plupart des auteurs se sont intéressés à l'étude de propriétés logiques telles que la prévention de famine ou d'interblocage plutôt qu'à des études de performances. Néanmoins, on peut citer [Forbes et al.96] et [Forbes96] où est présenté un tel modèle en vue d'une analyse quantitative. Les hypothèses qui y sont faites sont de type markoviennes. Entre autres, la gestion d'une file d'attente n'y est pas prise en compte. On peut aussi citer [Barbosa et al.89], dans lequel les auteurs s'intéressent à la quantification du codébit obtenu par une classe de politiques d'ordonnancement

particulières. Nous reviendrons sur ce point dans la partie 3.6.3. Avant de passer à une définition plus formelle des systèmes de requêtes, faisons la remarque suivante : on a vu dans la partie 2.1.1 la différence de traitement qui apparaissait entre verrous exclusifs et verrous non exclusifs dans une base de données. Le modèle des systèmes de requêtes ne prenant en compte que les exclusions entre paires de requêtes, peu importe de savoir si cette exclusion est due à des verrous exclusifs ou non exclusifs. Pour peu que l'on considère un graphe non réflexif (c'est à dire un graphe dans lequel un sommet ne fait pas nécessairement partie de l'ensemble de ses voisins), on peut autoriser une requête à ne pas être en exclusion avec une autre requête du même type. Ainsi, on tient compte notamment les requêtes ne demandant que des verrous non exclusifs.

La structure d'un système de requêtes \mathbf{S} est la suivante :

- un ensemble de N types de requêtes :

$$\text{Req}(\mathbf{S}) = \{v_i(\mathbf{S}), 1 \leq i \leq N\};$$

- un graphe non orienté :

$$\mathcal{G}(\mathbf{S}) = (\text{Req}(\mathbf{S}), E(\mathbf{S}))$$

appelé *graphe d'exclusion du système* \mathbf{S} .

On oublie ainsi les ressources du système. La dynamique de \mathbf{S} est donnée par :

- la suite des arrivées successives de requêtes :

$$s(\mathbf{S}) = \{s(\mathbf{S}, n) \in \text{Req}(\mathbf{S}), n \geq 1\}.$$

La sémantique du graphe \mathcal{G} est la suivante : étant données deux requêtes v et v' , $\{v, v'\} \in E$ si et seulement si les requêtes v et v' sont concurrentes, c'est-à-dire si elles ne peuvent être traitées simultanément. On notera $v \perp v'$ si $\{v, v'\} \in E$ et $v \parallel v'$ sinon. On peut remarquer que, si l'on tient à ce que \mathcal{G} ait un sens "physique" (exception faite des bases de données, on considère généralement dans les systèmes de ressources partagées que deux requêtes demandant l'accès aux mêmes ressources sont concurrentes), il faut que la condition suivante soit vérifiée :

Définition 2.1.1 (condition de "réalisme")

$$\forall i, (v_i \parallel v_i) \Rightarrow (\forall j, v_i \parallel v_j). \tag{2.2}$$

En effet, deux requêtes d'un même type vont demander l'accès aux mêmes ressources et sont donc concurrentes. La seule exception à cette règle est une requête vide, qui correspond à une requête ne demandant pas de ressource, et par conséquent ne peut être en concurrence avec aucune autre requête, pas même elle-même.

Remarque 1 : Il n'est pas toujours nécessaire de supposer "réaliste" un système de requêtes afin d'en estimer le codébit. On gagne donc en généralité en considérant des graphes non orientés quelconques. On peut par ailleurs remarquer que pour un graphe \mathcal{G} vérifiant la condition de réalisme, $\mathcal{G} \setminus \{v, V(v) = \emptyset\}$ est réflexif. Les requêtes vides n'influençant pas la dynamique d'un système de requêtes, nous considérerons donc quand ce sera nécessaire des graphes réflexifs.

On peut maintenant définir le codébit d'un système de requêtes de la même façon qu'on l'a fait pour un système de ressources : on introduit cette fois des dateurs $\{C_i(\mathbf{S}, n)\}_{1 \leq i \leq N, n \geq 0}$ tels que $C_i(\mathbf{S}, n)$ représente la date de fin de traitement de l'ensemble des requêtes égales à v_i figurant parmi les n premières requêtes de \mathbf{S} . Par convention, $C_i(0) = 0$ pour tout i . La dynamique présentée dans la partie 1.1 se traduit cette fois de la façon suivante :

$$C_i(n+1) = \begin{cases} 1 + \max_{j, v_j \in V(v_i)} C_j(n) & \text{si } s(n+1) = v_i, \\ C_i(n) & \text{sinon.} \end{cases} \quad (2.3)$$

On pose

$$\mathbf{C}(\mathbf{S}, n) \stackrel{\text{def}}{=} (C_1(\mathbf{S}, n), \dots, C_N(\mathbf{S}, n)).$$

$C_i(\mathbf{S}, n)$ représentant la date de fin de traitement de l'ensemble des requêtes égales à v_i et figurant parmi les n premières requêtes de \mathbf{S} , la date de fin de traitement des n premières requêtes de \mathbf{S} est

$$h(\mathbf{S}, n) \stackrel{\text{def}}{=} |\mathbf{C}(\mathbf{S}, n)|_\infty.$$

Ce qui nous amène à définir le codébit de \mathbf{S} comme étant :

$$\gamma(\mathbf{S}) \stackrel{\text{def}}{=} \lim_{n \rightarrow \infty} \frac{h(\mathbf{S}, n)}{n} \text{ si cette limite existe.}$$

Remarque 2 : On peut facilement vérifier par récurrence que pour tout v_i tel que $V(v_i) = \emptyset$ et tout n , $C_i(n) = 0$. $h(n)$ peut donc se récrire

$$h(n) = \max_{i, v_i \neq \emptyset} C_i(n).$$

Remarque 3 : Si le graphe \mathcal{G} associé à un système de requêtes est une clique, autrement dit si $\mathcal{G} = K_n$, on vérifie facilement par récurrence que $h(n) = n$ et donc que $\gamma = 1$. Ce résultat est tout à fait naturel : les requêtes de \mathbf{S} étant toutes concurrentes, elles sont traitées au rythme de une par unité de temps.

2.1.4 Équivalence entre ces deux modèles

Nous montrons dans cette partie comment il est possible de faire se correspondre un système de ressources et un système de requêtes, de telle façon que leurs codébits soient les mêmes.

Proposition 1 *Soit \mathbf{S} un système de ressources. On définit \mathbf{S}' , le système de requêtes associé à \mathbf{S} , comme suit :*

$$\begin{aligned} \text{Req}(\mathbf{S}') &= \text{Req}(\mathbf{S}), \\ s(\mathbf{S}') &= s(\mathbf{S}), \\ \mathcal{G}(\mathbf{S}') &= (\text{Req}(\mathbf{S}'), E(\mathbf{S}')) \text{ avec } E(\mathbf{S}') = \bigcup_{v_i \cap v_j \neq \emptyset} \{v_i, v_j\}. \end{aligned}$$

On notera $T_{\text{Req}}(\mathbf{S})$ le système \mathbf{S}' . L'égalité suivante est alors vraie pour tout n :

$$h(\mathbf{S}, n) = h(\mathbf{S}', n).$$

Preuve : Il suffit de démontrer que pour tout n et tout i

$$T_i(\mathbf{S}, n) = \max_{j, r_i \in v_j} C_j(\mathbf{S}', n). \quad (2.4)$$

La démonstration de cette égalité se fait par récurrence sur n . Celle-ci est en effet claire pour $n = 0$. Supposons la vraie pour n . On pose $s(n+1) = v_k$. On a donc

$$\begin{aligned} C_j(n+1) &= C_j(n) \text{ si } j \neq k, \\ C_k(n+1) &= 1 + \max_{j, v_j \in V(v_k)} C_j(n). \end{aligned}$$

On va étudier deux cas :

– soit i tel que $r_i \notin v_k$. On a alors

$$\begin{aligned} T_i(n+1) &= T_i(n), \\ &= \max_{j, r_i \in v_j} C_j(n), \\ &= \max_{j, r_i \in v_j} C_j(n+1). \end{aligned}$$

– soit maintenant i tel que $r_i \in v_k$:

$$\begin{aligned} T_i(n+1) &= 1 + \max(T(n+1), \max_{j, r_j \in v_k} T_j(n)), \\ &= 1 + \max(T(n+1), \max_{j, r_j \in v_k} \max_{l, r_l \in v_l} C_l(n)), \\ &= 1 + \max(T(n), \max_{l, v_l \in V(v_k)} C_l(n)), \\ &= C_k(n+1), \\ &= \max_{j, r_i \in v_j} C_j(n). \end{aligned}$$

On peut donc conclure que pour tout n , $h(\mathbf{S}, n) = h(\mathbf{S}', n)$. □

Corollaire 2.1.1 *Soit \mathbf{S} un système de ressources et \mathbf{S}' le système $T_{Req}(\mathbf{S})$. Le codébit de \mathbf{S}' est défini si et seulement si $\gamma(\mathbf{S})$ est lui même défini et dans ce cas*

$$\gamma(\mathbf{S}) = \gamma(\mathbf{S}').$$

Preuve : Immédiate, d'après la proposition précédente. □

Exemple 1 : La figure 2.1 montre un système de ressources \mathbf{S} et le système de requêtes $T_{Req}(\mathbf{S})$ qui lui est associé. Les ressources de \mathbf{S} sont présentées en colonne et ses requêtes en ligne. La case correspondant à une ressource et une requête données est noircie si et seulement si cette requête nécessite la ressource en question.

Remarque 1 : Il peut être intéressant de présenter ici la démarche qui nous a amenés à considérer des systèmes de requêtes. On a vu dans la partie précédente que la structure logique d'un système de ressources peut être représentée par un hypergraphe dont les sommets correspondent aux ressources du système et les arêtes aux requêtes. Sur cet hypergraphe, le comportement dynamique du système de ressources a la propriété que

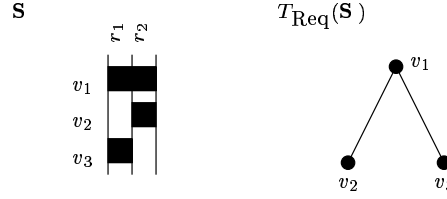


FIG. 2.1 – Un système de ressources \mathbf{S} et le système de requêtes qui lui est associé. Les valeurs de s pour chaque système sont les mêmes et peuvent être quelconques.

le traitement d'une requête ne modifie que les dateurs associés à une même arête (*i.e.* requête). Par ailleurs, ces dateurs prennent alors tous la même valeur. Il est donc naturel de regarder comment la dynamique du système de ressources se traduit sur l'hypergraphe dual. Il ne reste alors plus qu'à constater que sur cet hypergraphe dual, seule la notion d'exclusion est importante : étant donnés deux sommets du dual (*i.e.* deux requêtes), la seule chose qui importe est de savoir si une arête les relie (*i.e.* si elles sont concurrentes); peu importe, en effet, de savoir qu'elle est cette arête (*i.e.* ressource). On peut donc se ramener à l'étude du graphe représentatif ([Berge87], p. 30) de l'hypergraphe associé au système de ressources, c'est-à-dire au graphe d'exclusion du système de requêtes associé à notre système de ressources.

Montrons maintenant comment associer à un système de requêtes \mathbf{S} , un système de ressources \mathbf{S}' qui ait le même codébit. Ceci n'est possible que si \mathbf{S} vérifie la condition de réalisme 2.2. En effet, dans un système de ressources, une requête non vide est nécessairement concurrente avec elle-même.

Proposition 2 Soit \mathbf{S} un système de requêtes, vérifiant la condition de réalisme 2.2, défini par :

$$\begin{aligned} Req(\mathbf{S}) &= \{v_i(\mathbf{S}), 1 \leq i \leq N\}, \\ \mathcal{G}(\mathbf{S}) &= (Req(\mathbf{S}), E(\mathbf{S})), \\ s(\mathbf{S}) &= \{s(\mathbf{S}, n) \in Req(\mathbf{S}), n \geq 1\}. \end{aligned}$$

\mathbf{S}' est construit de la façon suivante :

$$\begin{aligned} Res(\mathbf{S}') &= \{r_{i,j} \text{ t.q. } \{v_i, v_j\} \in E(\mathbf{S}), i < j\}, \\ Req(\mathbf{S}') &= \{v_i(\mathbf{S}'), 1 \leq i \leq N\} \text{ avec } v_i(\mathbf{S}') = Res(\mathbf{S}') \cap \cup_k \{r_{i,k}, r_{k,i}\}, \\ s(\mathbf{S}') &= s(\mathbf{S}). \end{aligned}$$

L'existence de $\gamma(\mathbf{S})$ (resp. $\gamma(\mathbf{S}')$) implique l'existence de $\gamma(\mathbf{S}')$ (resp. $\gamma(\mathbf{S})$). Par ailleurs, quand les codébits de \mathbf{S} et \mathbf{S}' sont définis, ils sont égaux. On notera $T_{Res}(\mathbf{S})$ le système

de ressources \mathbf{S}' .

Preuve : Il suffit de remarquer que $T_{\text{Req}}(T_{\text{Res}}(\mathbf{S})) = \mathbf{S}$. □

Remarque 2 : On peut identifier $v_i(\mathbf{S})$ et $v_i(\mathbf{S}')$.

Exemple 2 : La figure 2.2 montre un système de requêtes \mathbf{S} et le système de ressources $T_{\text{Res}}(\mathbf{S})$ qui lui est associé.

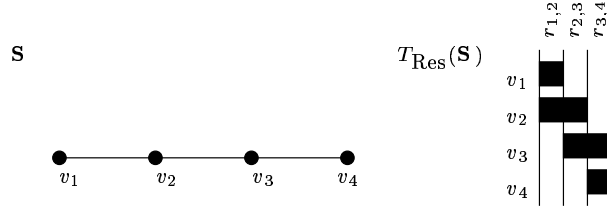


FIG. 2.2 – Un système de requêtes \mathbf{S} et le système de ressources qui lui est associé. Les valeurs de s sont les mêmes pour chaque système et peuvent être quelconques.

Remarque 3 : La définition de $T_{\text{Res}}(\mathbf{S})$ est moins naturelle que celle de $T_{\text{Req}}(\mathbf{S})$. En effet, alors que la définition de $\mathcal{G}(T_{\text{Req}}(\mathbf{S}))$ est relativement canonique, la construction de $\text{Res}(T_{\text{Res}}(\mathbf{S}))$ est beaucoup plus arbitraire. La figure 2.3 montre l'exemple de trois systèmes: deux systèmes de ressources \mathbf{S}_1 et \mathbf{S}_2 et un système de requêtes \mathbf{S} tels que

$$\mathbf{S} = T_{\text{Req}}(\mathbf{S}_1) = T_{\text{Req}}(\mathbf{S}_2),$$

et où $\mathbf{S}_2 = T_{\text{Res}}(\mathbf{S})$ est beaucoup moins simple que \mathbf{S}_1 .

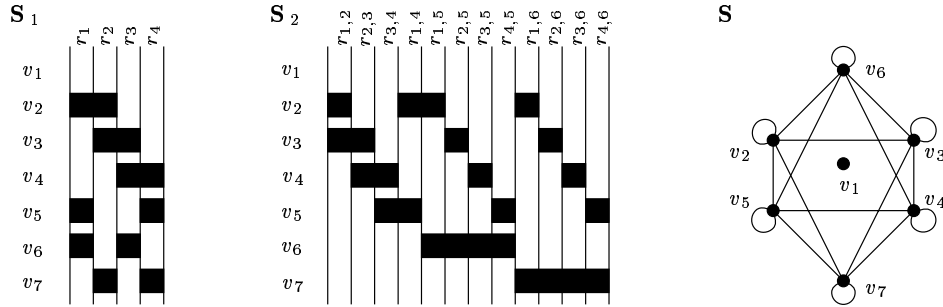


FIG. 2.3 – Deux systèmes de ressources \mathbf{S}_1 et \mathbf{S}_2 et un système de requêtes \mathbf{S} tels que $\mathbf{S} = T_{\text{Req}}(\mathbf{S}_1) = T_{\text{Req}}(\mathbf{S}_2)$ et $\mathbf{S}_2 = T_{\text{Res}}(\mathbf{S})$. Les valeurs de s sont les mêmes pour chaque système et peuvent être quelconques.

Définition 2.1.2 Pour un système de ressources \mathbf{S} , on définit

$$\mathcal{G}(\mathbf{S}) \stackrel{\text{def}}{=} \mathcal{G}(T_{\text{Req}}(\mathbf{S})) \text{ et } C_i(\mathbf{S}, n) \stackrel{\text{def}}{=} C_i(T_{\text{Req}}(\mathbf{S}), n).$$

On ne définira par contre pas de dateurs $T_i(\mathbf{S}, n)$ pour un système de requêtes, étant donné le caractère non canonique du système $T_{\text{Res}}(\mathbf{S})$. Par ailleurs, pour un système quelconque,

Définition 2.1.3

$$\underline{h}(n) \stackrel{\text{def}}{=} \min_i C_i(n),$$

$$C_{s(m)}(n) \stackrel{\text{def}}{=} C_i(n) \text{ si } s(m) = v_i.$$

Ces deux définitions nous seront utiles par la suite.

2.1.5 Équivalence avec un modèle de files d'attente

Nous allons montrer qu'il est possible de voir les systèmes de ressources présentés dans la partie 2.1.2 comme des réseaux simples de files d'attente avec deux types de synchronisations particulières : des synchronisations de type *arrivée-arrivée* et des synchronisations de type *service-service*. On pourra trouver une présentation des modèles de files d'attente avec de telles contraintes de synchronisation dans [Baccelli et al.89]. Il est à noter que ce sont précisément ces synchronisations qui rendent le modèle difficile à traiter.

Étant donné un système de ressources \mathbf{S} , nous allons définir un réseau de files d'attente \mathbf{F} de dynamique équivalente à celle de \mathbf{S} (voir la figure 2.4). Considérons donc K files (K est le nombre de ressources de \mathbf{S}). À chacune de ces files correspond un serveur de capacité 1 et de temps de traitement déterministe égal à une unité de temps. Les arrivées de clients dans \mathbf{F} sont déduites des arrivées de requêtes dans \mathbf{S} comme suit : à chaque requête $v = \{r_{\alpha_1}, \dots, r_{\alpha_p}\}$ arrivant dans \mathbf{S} correspond une arrivée de client de type v dans chacune des files α_i . Ces arrivées de clients se font de façon synchrone. On parle donc de synchronisation de type *arrivée-arrivée*. On peut éventuellement se passer de ce type de synchronisations en ajoutant au réseau une file d'attente, en amont des K files déjà définies, qui soit reliée à un serveur de temps de traitement nul dont le seul but soit la création des clients dans les files *ad hoc*. Afin de garantir que les clients issus d'une même requête seront servis en même temps, il faut encore préciser que, pour commencer son exécution, un client de type $v = \{r_{\alpha_1}, \dots, r_{\alpha_p}\}$ attend que soit prêt à être servi un client du même type dans chacune des files α_i ¹. Ces synchronisations sur les dates de début de service des clients sont de type *service-service*. Il est maintenant possible de vérifier

1. Il n'est pas nécessaire d'imposer que les clients soient issus de la même requête : cette contrainte est garantie par les synchronisations *arrivée-arrivée* et par le fait que chaque file préserve l'ordre FIFO

que les dynamiques de \mathbf{S} et \mathbf{F} sont les mêmes, simplement en vérifiant que deux clients issus d'une même requête sont servis de façon synchrone dans \mathbf{F} , à la date à laquelle est exécutée la requête qui leur correspond dans \mathbf{S} . Nous avons représenté sur la figure 2.4 les exécutions couplées d'un système de files d'attente et d'un système de ressources. Le système de file d'attente est représenté à une date t strictement comprise entre 0 et 1. Il est à remarquer que les deux systèmes sont saturés, mais que l'équivalence que nous avons présentée est encore valable dans le cas de systèmes non saturés. Elle serait de même valable dans le cas de temps de services quelconques.

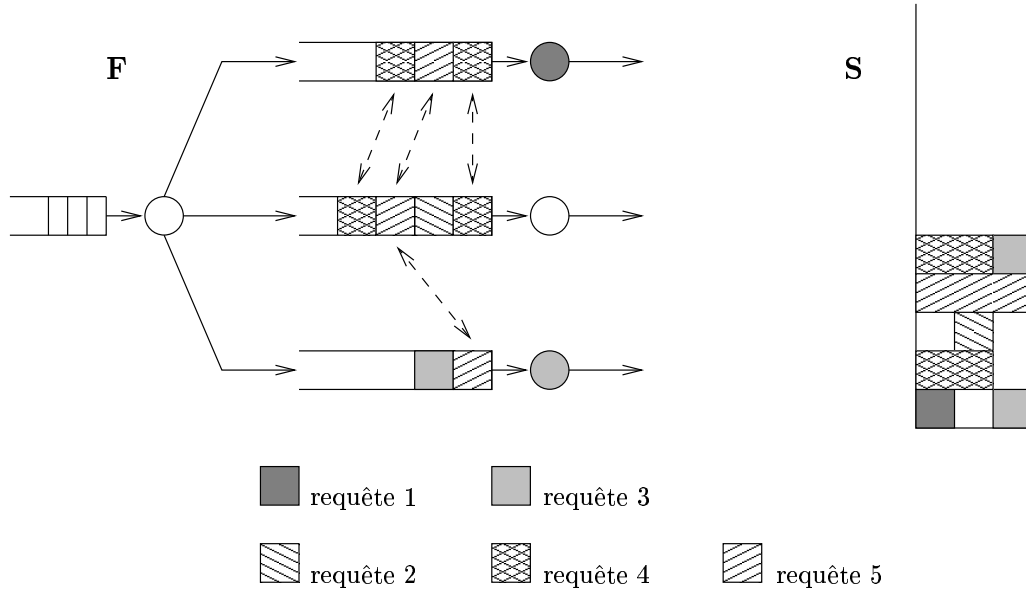


FIG. 2.4 – Modèle de files d'attente équivalent à un système de ressources. Il y a 5 type de requêtes possibles dans \mathbf{S} : $v_1 = \{r_1\}$, $v_2 = \{r_2\}$, $v_3 = \{r_3\}$, $v_4 = \{r_1, r_2\}$ et $v_5 = \{r_1, r_2, r_3\}$. Celles-ci arrivent dans l'ordre $v_1, v_4, v_3, v_2, v_5, v_3, v_4$. Les différentes requêtes sont chacune représentée d'une couleur différente, la même que celle des clients qui leur correspondent. Les flèches en pointillés représentent des synchronisations de type service-service entre clients issus d'une même requête. Le dessin de droite représente les dates de service des requêtes dans \mathbf{S} . On trouvera plus de détails sur ce type de représentation dans la partie 2.3.1.

2.2 Réécriture dans la pseudo-algèbre $(\max, +)$

Nous commençons par présenter rapidement la pseudo-algèbre $(\max, +)$ dans la partie 2.2.1. Nous montrons dans la partie 2.2.2 que les équations d'évolution des systèmes de ressources partagées peuvent s'interpréter comme des systèmes $(\max, +)$ -linéaires. Enfin,

la partie 2.2.3 montre l'application des résultats de $(\max, +)$ que l'on peut faire pour le cas particulier des systèmes de ressources partagées périodiques.

2.2.1 Présentation rapide de $(\max, +)$

L'algèbre $(\max, +)$ est un outil puissant permettant de traiter un grand nombre de modèles de systèmes dynamiques sur un espace de temps discret. Pour la présenter succinctement, nous dirons qu'elle correspond à une adaptation de l'algèbre linéaire classique (sur \mathbb{R}^n) dans laquelle l'opérateur de groupe "+" est remplacé par "max" (noté \oplus) et l'opérateur d'anneau "×" par "+" (noté \otimes). On n'a donc pas à proprement parler de structure d'algèbre. En effet, il manque la structure de groupe, "max" ne permettant pas de définir l'inverse d'un élément. Cependant, les propriétés classiques d'associativité et de distributivité étant préservées, on parle, par abus de langage, d'*algèbre* $(\max, +)$. L'élément neutre de \oplus (absorbant de \otimes) est $-\infty$, l'élément neutre de \otimes est 0.

Un des problèmes les plus célèbres qui se ramène à un système linéaire dans $(\max, +)$ est certainement le problème de la recherche du plus long chemin dans un graphe valué. En effet, si on note $p_{i,j}$ le poids de l'arête allant de i vers j ($p_{i,j} = -\infty$ s'il n'y a pas d'arête) et $P_{i,j}^n$ le poids du plus long chemin de taille n reliant i à j , on s'aperçoit que

$$P_{i,j}^{n+1} = \max_k (P_{i,k}^n + p_{k,j}).$$

On reconnaît la version " $(\max, +)$ " d'un produit matrice-vecteur dans l'algèbre linéaire classique.

De nombreux modèles, entre autres informatiques, admettent une écriture dans $(\max, +)$. Ces modèles correspondent majoritairement à des systèmes dynamiques régis par des équations d'évolution dans lesquelles interviennent des synchronisations (d'où le "max") et des traitements séquentiels (d'où le "+"). Parmi ceux-ci, on peut citer les graphes d'événements temporisés [Baccelli et al.92a], les automates temporisés [Gaubert93], mais aussi certains modèles d'économie [Yakovenko et al.92] ou de mécanique statistique [Griffiths90]. On pourra se rapporter à la thèse de Jean Mairesse [Mairesse95] pour une présentation plus détaillée de tels exemples.

D'un point de vue qualitatif, beaucoup de résultats existent déjà dans $(\max, +)$. Ce sont principalement des résultats assurant la convergence de tel système vers un régime stationnaire. Les résultats quantitatifs sont par contre plus rares.

Nous allons maintenant détailler la façon dont les équations d'évolution des systèmes de ressources partagées que l'on étudie se réécrivent dans $(\max, +)$. Nous montrerons par

la suite les résultats qu'une telle reformulation apporte.

2.2.2 Réécriture des équations d'évolution dans $(\max, +)$

Pour un système de ressources, si l'on pose

$$V_n \stackrel{\text{def}}{=} \begin{pmatrix} T_1(n) \\ \vdots \\ T_K(n) \end{pmatrix},$$

alors l'équation d'évolution 2.1 peut se récrire

$$\begin{aligned} V_{n+1} &= M(n) \otimes V_n, \\ &= M(n) \otimes \cdots \otimes M(0) \otimes V_0, \\ h(n+1) &= |M(n) \otimes \cdots \otimes M(0)|_\infty, \\ \gamma &= \lim_{n \rightarrow \infty} \frac{1}{n+1} |M(n) \otimes \cdots \otimes M(0)|_\infty, \end{aligned}$$

où $M(n) = (m_{i,j}^n)$ avec

$$m_{i,j}^n = \begin{cases} 1 & \text{si } \{r_i, r_j\} \subset s(n+1), \\ 0 & \text{si } i = j \text{ et } r_i \notin s(n+1), \\ -\infty & \text{si } i \neq j \text{ et } \{r_i, r_j\} \not\subset s(n+1). \end{cases}$$

On peut donc voir γ comme l'exposant de Lyapunov de la suite de matrices $(M(n))_{n \geq 0}$, par extension à $(\max, +)$ de la définition d'un exposant de Lyapunov dans les systèmes dynamiques classiques. On peut d'ores et déjà constater que si la suite s est périodique, la suite des matrices $M(n)$ est périodique.

En ce qui concerne les dateurs $C_i(n)$ d'un système quelconque, en posant cette fois-ci

$$V_n \stackrel{\text{def}}{=} \begin{pmatrix} C_1(n) \\ \vdots \\ C_N(n) \end{pmatrix},$$

l'équation d'évolution 2.3 se réécrit $V_{n+1} = M(n) \otimes V_n$ avec $M(n) = (m_{i,j}^n)$ définie de la façon suivante :

$$m_{i,j}^n = \begin{cases} 1 & \text{si } s(n+1) = v_i \text{ et } v_j \in V(v_i), \\ 0 & \text{si } s(n+1) \neq v_i \text{ et } i = j, \\ -\infty & \text{sinon.} \end{cases} \quad (2.5)$$

On a $h(n) = \max_{i,j} (M(n-1) \otimes \cdots \otimes M(0))_{i,j}$ et $\underline{h}(n) = \min_i (M(n-1) \otimes \cdots \otimes M(0) \otimes V_0)_i$. De même que précédemment, γ peut être vu comme l'exposant de Lyapunov de la suite de matrices $(M(n))_{n \geq 0}$. Enfin, on peut vérifier que si la suite s est périodique, alors la suite des matrices $M(n)$ l'est aussi.

2.2.3 Résultats apportés par $(\max, +)$. Cas particulier des systèmes périodiques

Nous présentons dans cette partie certains des résultats que la théorie de l'algèbre $(\max, +)$ apporte à l'étude des systèmes de ressources partagées. Ces résultats portent principalement sur deux types de systèmes fondamentalement différents : les systèmes périodiques (*i.e.* pour lesquels la suite s est périodique) et les systèmes stochastiques (*i.e.* pour lesquels la suite s est stochastique). Les résultats portant sur les systèmes stochastiques seront présentés dans la partie 3.1.2. Nous présentons par contre ici les résultats portant sur les systèmes périodiques, ceux-ci n'étant plus étudiés par la suite.

Définition 2.2.1 *Un système \mathcal{S} est périodique s'il existe $p \in \mathbb{N}^*$ tel que pour tout n ,*

$$s(n) = s(n+p).$$

On définit P , sa période, comme le plus petit p vérifiant l'assertion précédente. L'approche $(\max, +)$ permet de montrer que le codébit d'un système périodique est toujours défini et de calculer ce codébit. En effet, si on définit $A = M(P-1) \otimes \cdots \otimes M(0)$ et λ la plus grande valeur propre de A , on sait [Baccelli et al.92a] qu'il existe un entier M tel que pour tout $m \geq M$, $A^{m+1} = \lambda \otimes A^m$. On en déduit que $\lim_{n \rightarrow \infty} |A^n|_\infty / n = \lambda$, ce qui amène $\gamma = \lambda/P$. Des méthodes numériques sont proposées dans [Baccelli et al.92a] qui permettent de calculer λ . On pourra se rapporter à cet ouvrage pour une étude complète des systèmes périodiques.

2.3 Processus induits

Nous présentons dans cette partie plusieurs notions importantes. La partie 2.3.1 est consacrée à la présentation d’une représentation graphique de l’évolution d’un système de ressources donné. Cette représentation permet une bonne intuition des phénomènes sous-jacents dans les systèmes de ressources partagées. Elle a initialement été introduite dans [Brilman93] et [Brilman et al.95b]. La partie 2.3.2 présente de manière formelle quelques processus dont l’interprétation pourra se faire grâce à la représentation graphique exposée auparavant.

2.3.1 Représentation graphique

La dynamique d’un système de ressources saturé \mathbf{S} peut être représentée sur un intervalle de temps fini par des empilements de pièces de “Tétris” : il suffit d’imaginer un jeu de Tétris, $\text{Tet}(\mathbf{S})$, sur K colonnes, dans lequel la n -ième pièce est une pièce de hauteur 1 occupant les colonnes correspondant aux ressources requises par la n -ième requête de \mathbf{S} . Il est alors clair que la hauteur à laquelle la n -ième pièce de $\text{Tet}(\mathbf{S})$ tombe est égale à la date de début d’exécution de la n -ième requête de \mathbf{S} ². Par ailleurs, la hauteur de la i -ème colonne de $\text{Tet}(\mathbf{S})$ après empilement des n premières pièces est égale à $T_i(\mathbf{S}, n)$. Ces deux résultats proviennent du fait que les équations d’évolution de \mathbf{S} et de $\text{Tet}(\mathbf{S})$ sont les mêmes. L’intérêt de cette remarque est qu’elle permet une représentation graphique intuitive de la dynamique d’un système de ressources \mathbf{S} . Bien que cette représentation présente une certaine lourdeur, elle permet des interprétations “naturelles” de résultats plus abstraits. La figure 2.5 montre un exemple de système \mathbf{S} et le Tétris $\text{Tet}(\mathbf{S})$ qui lui est associé. On montre sur cette figure comment l’empilement des pièces du Tétris permet de retrouver les valeurs des dateurs de \mathbf{S} .

2.3.2 Lignes et motifs

Nous présentons dans cette partie des définitions inspirées par la représentation en “Tétris” d’un système de ressources. Ces définitions s’étendant naturellement à un système de requêtes, nous ne ferons pas de distinction entre systèmes de ressources et systèmes de requêtes.

2. Le “joueur” est passif : il laisse les pièces tomber sans les déplacer

S :

$$\begin{aligned} \text{Res} &= \{r_1, r_2, r_3, r_4\} \\ \text{Req} &= \{v_1 = \{r_1, r_2\}, v_2 = \{r_2, r_3\}, v_3 = \{r_3, r_4\}, v_4 = \{r_4, r_1\}, v_5 = \emptyset\} \\ s &= (v_1, v_3, v_5, v_2, v_3, v_4, v_2, v_1, \dots) \end{aligned}$$

On voit directement sur $\text{Tet}(\mathbf{S})$ que

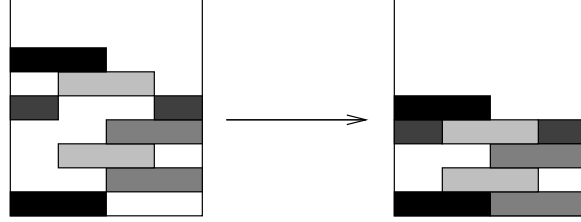
$$\begin{aligned} T_1(8) &= 5 & C_1(8) &= 5 \\ T_2(8) &= 5 & C_2(8) &= 4 \\ T_3(8) &= 4 & C_3(8) &= 3 \\ \text{Tet}(\mathbf{S}) : & & T_4(8) &= 4 & C_4(8) &= 4 \end{aligned}$$


FIG. 2.5 – Représentation d'un système de ressources par un Tétris.

Définition de la n -ième ligne d'un système \mathbf{S}

La n -ième ligne d'un système \mathbf{S} est l'ensemble des requêtes de \mathbf{S} qui sont traitées à la même date que la n -ième requête de \mathbf{S}

Définition 2.3.1 \mathbf{S} étant un système quelconque, on définit pour $n \geq 1$,

$$L(\mathbf{S}, n) \stackrel{\text{def}}{=} \{v_i \text{ t.q. } \exists m \geq 1, C_i(\mathbf{S}, m) = C_{s(n)}(\mathbf{S}, n)\}.$$

On appelle $L(\mathbf{S}, n)$ la n -ième ligne du système \mathbf{S} . On notera $|L(\mathbf{S}, n)|$ son cardinal.

Remarque 1 : Il pourrait sembler plus naturel de définir la n -ième ligne de \mathbf{S} comme l'ensemble des requêtes traitées entre les dates $(n-1)$ et n . Cependant, une telle définition serait moins facile à manipuler que celle que nous avons proposée.

Proposition 3 Pour tout $n \geq 1$,

$$\underline{h}(n) \leq \sum_{i=1}^n \frac{1}{|L(i)|} \leq h(n).$$

Preuve : Commençons par démontrer la première partie de cette proposition.

$$\begin{aligned}
 \underline{h}(n) &= \sum_{i=1}^{\underline{h}(n)} 1, \\
 &= \sum_{i=1}^{\underline{h}(n)} \sum_{p \leq n, C_{s(p)}(p)=i} \frac{1}{\text{card}\{p \leq n, C_{s(p)}(p) = i\}}, \\
 &= \sum_{i=1}^{\underline{h}(n)} \sum_{p \leq n, C_{s(p)}(p)=i} \frac{1}{\text{card}\{p' \leq n, C_{s(p')}(p') = C_{s(p)}(p)\}}.
 \end{aligned}$$

Comme $(p' > n) \Rightarrow (C_{s(p')}(p') > \underline{h}(n))$,

$$\begin{aligned}
 \underline{h}(n) &= \sum_{i=1}^{\underline{h}(n)} \sum_{p \leq n, C_{s(p)}(p)=i} \frac{1}{|L(p)|}, \\
 &\leq \sum_{p \leq n} \frac{1}{|L(p)|}.
 \end{aligned}$$

La deuxième partie de la proposition se montre sur le même principe :

$$\begin{aligned}
 h(n) &= \sum_{i=1}^{h(n)} 1, \\
 &= \sum_{i=1}^{h(n)} \sum_{p, C_{s(p)}(p)=i} \frac{1}{\text{card}\{p, C_{s(p)}(p) = i\}}, \\
 &= \sum_{i=1}^{h(n)} \sum_{p, C_{s(p)}(p)=i} \frac{1}{\text{card}\{p', C_{s(p')}(p') = C_{s(p)}(p)\}}, \\
 &= \sum_{i=1}^{h(n)} \sum_{p, C_{s(p)}(p)=i} \frac{1}{|L(p)|}, \\
 &\geq \sum_{i=1}^{h(n)} \sum_{p \leq n, C_{s(p)}(p)=i} \frac{1}{|L(p)|}.
 \end{aligned}$$

Or, $(p \leq n) \Rightarrow (C_{s(p)}(p) \leq h(n))$, et donc

$$h(n) \geq \sum_{p \leq n} \frac{1}{|L(p)|}.$$

□

Définition de $\text{Ind}(\mathbf{S})$

On note $\text{Ind}(\mathbf{S})$ l'ensemble des valeurs possibles d'une ligne de \mathbf{S} . C'est donc un sous-ensemble de $\mathcal{P}(\text{Req}(\mathbf{S}))$.

Définition 2.3.2 *On définit $\text{Ind}(\mathbf{S})$ par :*

$$\text{Ind}(\mathbf{S}) \stackrel{\text{def}}{=} \{I \in \mathcal{P}(\text{Req}(\mathbf{S})) \setminus \emptyset \text{ t.q. } \forall (v, v') \in I^2, V(v) \neq \emptyset \text{ et } (v \neq v' \Rightarrow v \parallel v')\}.$$

Cette définition implique que $\text{Ind}(\mathbf{S})$ correspond à l'ensemble des stables ([Berge83], p. 260) non vides de $\mathcal{G}(\mathbf{S})$.

Définition d'un motif

C'est dans [Mairesse93b] que l'on trouve pour la première fois la notion de motif. Nous employons ici un abus de langage : par "motif", il faut entendre "motif synchronisant". En effet, un motif de \mathbf{S} est une suite de requêtes qui synchronise l'ensemble des dateurs de \mathbf{S} (que \mathbf{S} soit un système de ressources ou un système de requêtes).

Définition 2.3.3 *Une suite finie $M = (v_{\alpha_1}, \dots, v_{\alpha_{m-1}}, v_{\alpha_m}, v_{\alpha_{m-1}}, \dots, v_{\alpha_1})$ de requêtes de \mathbf{S} est un motif de \mathbf{S} si et seulement si*

$$(\forall i \in [1..m-1], v_{\alpha_i} \perp v_{\alpha_{i+1}}) \text{ et } (\forall v \in \text{Req}(\mathbf{S}) \exists i \in [1..m] \text{ t.q. } v \perp v_{\alpha_i}).$$

On peut remarquer que tout système \mathbf{S} admet un motif. En effet, \mathbf{S} étant connexe, \mathcal{G} est connexe; et donc, un chemin couvrant de \mathcal{G} (aller-retour) est un motif de \mathbf{S} . Nous allons maintenant énoncer une proposition qui justifie le fait de considérer de tels motifs.

Proposition 4 (propriété de perte de mémoire des motifs) *Soit \mathbf{S}_1 et \mathbf{S}_2 deux systèmes tels que $\text{Req}(\mathbf{S}_1) = \text{Req}(\mathbf{S}_2)$ et $\mathcal{G}(\mathbf{S}_1) = \mathcal{G}(\mathbf{S}_2)$. Soit*

$$M = (v_{\alpha_1}, \dots, v_{\alpha_{m-1}}, v_{\alpha_m}, v_{\alpha_{m-1}}, \dots, v_{\alpha_1})$$

un motif de \mathbf{S}_1 et \mathbf{S}_2 . On suppose qu'il existe s_1, s_2 deux suites finies de requêtes et s_∞ une suite infinie de requêtes telles que

$$s(\mathbf{S}_1) = s_1 M s_\infty \text{ et } s(\mathbf{S}_2) = s_2 M s_\infty.$$

On définit $n_1 = |s_1| + m$ et $n_2 = |s_2| + m$. Pour tout $n \geq 1$,

$$L(\mathbf{S}_1, n_1 + n) = L(\mathbf{S}_2, n_2 + n).$$

Ce résultat signifie que la dynamique d'un système après l'arrivée d'un motif ne dépend pas des requêtes arrivées avant ce motif.

Preuve : Il suffit de montrer que pour tout $n \geq 1$,

$$C_{s(\mathbf{S}_1, n_1+n)}(\mathbf{S}_1, n_1 + n) - h(\mathbf{S}_1, n_1) = C_{s(\mathbf{S}_2, n_2+n)}(\mathbf{S}_2, n_2 + n) - h(\mathbf{S}_2, n_2).$$

On peut tout d'abord remarquer que

$$\begin{aligned} h(\mathbf{S}_1, n_1) &= C_{s(\mathbf{S}_1, n_1)}(\mathbf{S}_1, n_1), \\ h(\mathbf{S}_2, n_2) &= C_{s(\mathbf{S}_2, n_2)}(\mathbf{S}_2, n_2), \end{aligned}$$

puisque

$$\begin{aligned} C_{s(\mathbf{S}_1, n_1)}(\mathbf{S}_1, n_1) &\geq \max_{v_j \in \cup_{1 \leq i \leq m} V(v_{\alpha_i})} C_j(\mathbf{S}_1, n_1), \\ &\geq \max_j C_j(\mathbf{S}_1, n_1), \\ C_{s(\mathbf{S}_2, n_2)}(\mathbf{S}_2, n_2) &\geq \max_{v_j \in \cup_{1 \leq i \leq m} V(v_{\alpha_i})} C_j(\mathbf{S}_2, n_2), \\ &\geq \max_j C_j(\mathbf{S}_2, n_2). \end{aligned}$$

Par ailleurs, pour $n \leq m - 1$,

$$C_{s(\mathbf{S}_1, n_1+n)}(\mathbf{S}_1, n_1 + n) = 1 + C_{s(\mathbf{S}_1, n_1+n-1)}(\mathbf{S}_1, n_1 + n - 1),$$

et donc l'égalité est vraie pour $n \leq m - 1$. Pour $n \geq m$, elle se démontre par récurrence à partir des équations d'évolution, en utilisant le fait que pour tout v , il existe i , $1 \leq i \leq m$, tel que $v \perp v_{\alpha_i}$. \square

Chapitre 3

Systemes independants

Nous étudions dans ce chapitre une classe particulière de systèmes de ressources partagées, les systèmes indépendants. De façon informelle, un système est dit indépendant si le processus des requêtes s correspond à une suite de variables aléatoires indépendantes et identiquement distribuées. Nous présentons cette définition de manière plus formelle dans la partie 3.1. Les parties suivantes sont consacrées à diverses approches possibles pour l'étude des systèmes indépendants. La partie 3.2 présente une approche naturelle qui consiste à quotienter l'espace d'états des dateurs associés à un système donné. L'espace quotient est l'équivalent de l'espace projectif dans $(\max, +)$. Une très jolie présentation des espaces projectifs en dimension 2 et 3 dans $(\max, +)$ se trouve dans [Mairesse93a]. Notre approche permet notamment une étude exacte des systèmes ne comportant que deux ressources. La partie 3.3 met en relation γ et ce qu'on appelle la "ligne asymptotique" d'un système donné. La partie 3.4 présente un théorème de composition qui permet de ramener l'étude de certains systèmes à l'étude de systèmes plus petits. Nous donnons en application le calcul exact du codébit pour un ensemble particulier de systèmes. Dans la partie 3.5, nous étudions γ d'un point de vue analytique, ce qui nous permet de restreindre la classe des systèmes intéressants à étudier aux seuls systèmes dans lesquels chaque requête possible a la même probabilité de tirage. De tels systèmes sont dits uniformes. La partie 3.6 leur est consacrée. On montre par ailleurs comment les résultats portant sur les systèmes uniformes se traduisent pour des systèmes non uniformes.

3.1 Présentation des systèmes indépendants

Dans la partie 3.1.1 nous présentons très brièvement les notions de systèmes stochastiques et de systèmes ergodiques. De tels systèmes ont déjà été étudiés d'un point de vue qualitatif mais très peu d'un point de vue quantitatif. En effet, leur définition se prête

mal à cette deuxième approche. Nous introduisons donc par la suite la notion de systèmes indépendants. Ces systèmes, de par leur définition, permettent en effet une approche quantitative de leur étude. La partie 3.1.2 présente quelques résultats apportés par $(\max, +)$ à l'étude des systèmes indépendants. Ces résultats sont de deux types : on commencera par présenter des résultats qualitatifs après quoi nous exposerons des méthodes générales (*i.e.* valable dans $(\max, +)$) de recherche de bornes pour γ . Dans la partie 3.1.3 nous exposons succinctement le problème de la simulation de systèmes indépendants.

3.1.1 Définitions

Définition 3.1.1 *Un système \mathcal{S} est dit stochastique si le processus des requêtes successives de \mathcal{S} est stochastique.*

Définition 3.1.2 *Un système stochastique \mathcal{S} est dit ergodique si le processus des requêtes successives de \mathcal{S} est ergodique.*

Théorème 1 *Si \mathcal{S} est un système ergodique,*

$$\left(\frac{h(\mathcal{S}, n)}{n} \right)_{n \geq 1}$$

converge presque sûrement et dans \mathcal{L}_1 vers une valeur réelle quand n tend vers l'infini.

Ce théorème peut être vu comme un cas particulier de plusieurs théorèmes différents que l'on pourra notamment trouver dans [Baccelli et al.92a] ou [Vincent93]. On définit donc

Définition 3.1.3 *Pour un système stochastique \mathcal{S} , non nécessairement ergodique,*

$$\gamma(\mathcal{S}) \stackrel{\text{def}}{=} \lim_{n \rightarrow \infty} \frac{\mathbb{E}h(\mathcal{S}, n)}{n} \text{ si cette limite existe.}$$

$\gamma(\mathcal{S})$ peut donc être défini pour des systèmes stochastiques non nécessairement ergodiques. Nous verrons par la suite que cette définition nous permettra de calculer le codébit de certains systèmes sans avoir à prouver que ceux-ci sont ergodiques. Nous introduisons maintenant la notion de système indépendant.

Définition 3.1.4 *Un système \mathcal{S} est dit indépendant si la suite des requêtes de \mathcal{S} est une suite de variables aléatoires indépendantes et identiquement distribuées à valeurs dans $\text{Req}(\mathcal{S})$.*

3.1 Présentation des systèmes indépendants

Un système indépendant est donc la donnée d'une structure logique et d'un vecteur de probabilité $\mathbf{p} = (p_1, \dots, p_N)$, N étant le nombre de requêtes différentes dans le système et p_i la probabilité de tirage de la requête v_i . On définit $\gamma(\mathcal{G}, \mathbf{p})$ comme étant le codébit d'un système indépendant dont la structure logique est donnée par \mathcal{G} et la distribution générique d'une requête par \mathbf{p} . On utilisera parfois la notation suivante :

$$P(v) = P(s(\mathbf{S}, 1) = v).$$

Le but de cette thèse est donc l'évaluation de $\gamma(\mathcal{G}, \mathbf{p})$ pour \mathcal{G} et \mathbf{p} donnés. Il y a peu d'approches connues, autres que la simulation, permettant actuellement d'approcher $\gamma(\mathcal{G}, \mathbf{p})$. Des méthodes permettant l'obtention de bornes ont toutefois été dégagées dans le cadre de l'étude des réseaux de Petri stochastiques et plus généralement dans le cadre de $(\max, +)$. Nous présentons ces méthodes dans la partie suivante. La simulation, quant à elle, présente deux inconvénients majeurs : pour un graphe \mathcal{G} trop grand, elle peut être trop longue à converger. Nous revenons sur ce point dans la partie 3.1.3. Par ailleurs, elle ne permet pas de dégager de comportements globaux de la fonction γ : un résultat de simulation pour $(\mathcal{G}_0, \mathbf{p}_0)$ donné ne permettra pas de prédire le résultat d'une simulation pour $(\mathcal{G}_1, \mathbf{p}_1)$. Notre approche se résumera donc en trois points : lorsque nous le pourrons, nous essayerons de trouver une expression exacte de $\gamma(\mathcal{G}, \mathbf{p})$. On verra que ceci est un cas assez peu fréquent. D'un point de vue plus général, nous dégagerons quelques propriétés analytiques de γ . Nous verrons comment ces propriétés peuvent s'interpréter "physiquement" et ainsi permettre de mieux comprendre la dynamique des systèmes de ressources partagées que l'on modélise. Enfin, par le biais d'approches différentes, nous proposerons des bornes pour γ dont nous étudierons la qualité.

Remarque 1 : Pour les systèmes indépendants, l'hypothèse de connexité présentée dans la partie 1.1 se traduit de la façon suivante :

Un système indépendant représenté par un couple $(\mathcal{G}, \mathbf{p})$ est dit connexe si et seulement si le graphe $\mathcal{G} \setminus \{v \text{ t.q. } P(v) = 0\}$ est connexe.

Dans la suite de cette thèse, les systèmes indépendants considérés seront implicitement supposés connexes à moins que le contraire ne soit précisé.

3.1.2 Résultats apportés par $(\max, +)$ aux systèmes stochastiques

On a vu dans la partie 2.2.2 que la suite des requêtes d'un système de requêtes \mathbf{S} définit une suite de matrices $\{M(n)\}_{n \geq 0}$. A chaque type de requête dans \mathbf{S} correspond une valeur possible de $M(n)$. $M(n)$ peut donc prendre N valeurs différentes : M_1, \dots, M_N, M_p étant la matrice associée au type de requêtes v_p . Pour un système de ressources partagées

stochastique, l'étude de γ peut donc être ramenée à l'étude de

$$\gamma(\{M_n\}_{n \geq 0}) = \lim_{n \rightarrow \infty} \frac{1}{n} |M(n) \otimes \cdots \otimes M(0)|_\infty \text{ si cette limite existe.}$$

Dans le cas d'un système indépendant, on se ramène à l'étude de

$$\gamma(M) = \lim_{n \rightarrow \infty} \frac{1}{n} |M(n) \otimes \cdots \otimes M(0)|_\infty$$

où les matrices $M(n)$ sont indépendantes et identiquement distribuées avec $M(0) = M$.

C'est cette traduction dans $(\max, +)$ qui permet, par exemple, de garantir que γ est défini pour tout système de ressources partagées ergodique (th. 1). Plus généralement, les résultats portant sur les produits de matrices stochastiques dans $(\max, +)$ vont amener des résultats sur les systèmes de ressources partagées stochastiques. La réciproque n'est pas vraie : il n'est pas possible d'associer à toute suite de matrices stochastiques un système de ressources partagées stochastique. En effet, les matrices représentant les requêtes d'un système de ressources partagées ont une forme bien particulière.

Nous allons dans un premier temps présenter les résultats qualitatifs apportés par $(\max, +)$ aux systèmes de ressources partagées stochastiques. Ces résultats seront présentés dans le formalisme des systèmes de ressources partagées et dans un cadre beaucoup plus restreint que celui dans lequel ils ont été prouvés. Cependant, ils seront suffisants pour la suite de cette thèse.

Dans un deuxième temps, nous présenterons quelques méthodes permettant l'obtention de bornes pour $\gamma(\{M_n\}_{n \geq 0})$ ou $\gamma(M)$. Ces méthodes s'appliquant à $(\max, +)$, elles sont susceptibles de s'appliquer à l'étude des systèmes de ressources partagées. Nous discuterons donc de leur applicabilité.

Résultats qualitatifs apportés par $(\max, +)$

Proposition 5 *Pour un système de ressources partagées indépendant, il existe une variable aléatoire Δ telle que la suite $\{h(n) - \underline{h}(n)\}_{n \geq 0}$ converge en loi vers Δ .*

Preuve : On trouvera la démonstration de ce résultat dans [Mairesse93b] (th. 5.1), une fois remarqué que si $(v_{\alpha_1}, \dots, v_{\alpha_m})$ est un motif, alors la matrice $M_{\alpha_1} \otimes \cdots \otimes M_{\alpha_m} \otimes M_1 \otimes \cdots \otimes M_N$ est *scs1 - cycl*. \square

Proposition 6 $\mathbb{E}\Delta < \infty$.

Preuve : Ce résultat provient du fait que les coefficients des matrices sont bornés supérieurement. \square

Proposition 7 *Pour un système de ressources partagées indépendant, $\{\underline{h}(n)/n\}_{n \geq 1}$ converge presque sûrement et dans \mathcal{L}_1 vers γ .*

Preuve : Immédiate, d'après les propositions précédentes. \square

Ces résultats ne sont pas les seuls résultats qualitatifs apportés par $(\max, +)$. Cependant, ils nous suffiront dans le cadre de cette thèse.

Quelques méthodes permettant l'obtention de bornes dans $(\max, +)$

Nous exposons maintenant quelques méthodes générales permettant l'obtention de bornes pour $\gamma(\{M_n\}_{n \geq 0})$ ou $\gamma(M)$. Certaines de ces méthodes ont été développées pour l'étude des réseaux de Petri stochastiques [Baccelli et al.92b] et ne s'appliquent qu'à certains type de matrices ne correspondant pas nécessairement aux matrices des systèmes de ressources partagées. Bien que nous présentions ces méthodes indépendamment les unes des autres, il est clair qu'elles sont le plus souvent utilisées conjointement.

Ordres stochastiques : Une première méthode est basée sur la notion d'ordre stochastique. Un ordre stochastique est un ordre (partiel) sur les variables aléatoires à valeurs dans R^n . Deux de ces ordres sont particulièrement intéressants dans $(\max, +)$: l'ordre \leq_{st} et l'ordre \leq_{icx} . Ils sont définis par :

$$\begin{aligned} X \leq_{st} Y & \text{ ssi pour toute fonction } f \text{ croissante, } \mathbb{E}f(X) \leq \mathbb{E}f(Y). \\ X \leq_{icx} Y & \text{ ssi pour toute fonction } f \text{ croissante convexe, } \mathbb{E}f(X) \leq \mathbb{E}f(Y). \end{aligned}$$

Ces ordres présentent les propriétés suivantes :

1. Pour toute fonction f croissante,

$$X \leq_{st} Y \Rightarrow f(X) \leq_{st} f(Y).$$

2. Pour toute fonction f croissante convexe

$$X \leq_{icx} Y \Rightarrow f(X) \leq_{icx} f(Y).$$

3. Si X' et Y' sont indépendantes de X et Y respectivement,

$$\begin{aligned} X \leq_{st} Y, X' \leq_{st} Y' & \Rightarrow X + X' \leq_{st} Y + Y', \\ X \leq_{icx} Y, X' \leq_{icx} Y' & \Rightarrow X + X' \leq_{icx} Y + Y'. \end{aligned}$$

Chapitre 3. Systèmes indépendants

En particulier, les fonctions \max et $+$ étant croissantes et convexes,

$$\begin{aligned}\forall n \ M(n) \leq_{st} M'(n) &\Rightarrow M(n) \otimes \cdots \otimes M(0) \leq_{st} M'(n) \otimes \cdots \otimes M'(0), \\ \forall n \ M(n) \leq_{icx} M'(n) &\Rightarrow M(n) \otimes \cdots \otimes M(0) \leq_{icx} M'(n) \otimes \cdots \otimes M'(0).\end{aligned}$$

On remarque donc que

$$\forall n, \ M(n) \leq_{st} M'(n) \Rightarrow \gamma(\{M(n)\}_{n \geq 0}) \leq \gamma(\{M'(n)\}_{n \geq 0}), \quad (3.1)$$

$$\forall n, \ M(n) \leq_{icx} M'(n) \Rightarrow \gamma(\{M(n)\}_{n \geq 0}) \leq \gamma(\{M'(n)\}_{n \geq 0}), \quad (3.2)$$

Bien que la première de ces implications découle de la seconde, elles amènent chacune des bornes différentes. L'équation 3.2 amène essentiellement des résultats de type “comparaison entre coefficients aléatoires et constants”. Par exemple, en utilisant le fait que $\mathbb{E}X \leq_{icx} X$, elle permet de minorer le codébit d'un système de ressources partagées dans lequel chaque requête v prend les ressources dont elle a besoin pendant un temps aléatoire T_v par le codébit d'un système de ressources partagées de même structure logique dans lequel chaque requête v prend les ressources dont elle a besoin pendant un temps constant égal à $\mathbb{E}T_v$.

L'équation 3.1 est plutôt utilisée en associant, à une suite $\{M(n)\}_{n \geq 0}$, une deuxième suite $\{M'(n)\}_{n \geq 0}$ dont on sait calculer le γ et qui minore (ou majore) la suite $\{M(n)\}_{n \geq 0}$ terme à terme. On dira que $\{M'(n)\}_{n \geq 0}$ minore (ou majore) *trajectoriellement* $\{M(n)\}_{n \geq 0}$. Ceci se fait généralement par construction d'un couplage dont on sait qu'il va induire un ordre \leq_{st} . Un tel couplage, en termes de réseaux de Petri, peut consister à sur-synchroniser le réseau (et donc à “augmenter” certaines matrices). En termes de systèmes de ressources partagées, il peut consister à remplacer certaines requêtes par des requêtes vides (et donc, à “diminuer” certaines matrices).

Association: Une deuxième méthode générale est basée sur la notion d'*association* de variables aléatoires. Des variables aléatoires X_1, \dots, X_n sont dites associées si et seulement si pour toutes fonctions f et g croissantes à valeurs réelles,

$$\mathbb{E}[f(X_1, \dots, X_n)g(X_1, \dots, X_n)] \geq \mathbb{E}f(X_1, \dots, X_n)\mathbb{E}g(X_1, \dots, X_n).$$

En particulier, les variables aléatoires X_1, \dots, X_n sont associées si elles sont fonctions croissantes de variables aléatoires indépendantes. Si X_1, \dots, X_n sont associées, on peut remarquer que

$$P(\max_i X_i \leq a) \geq \prod_i P(X_i < a).$$

3.1 Présentation des systèmes indépendants

Cette inégalité permet de décorréler des variables tout en ayant la possibilité de comparer les maximums de ces variables dans leurs versions dépendantes et indépendantes. Une fois remarqué que

$$h(n) = \max_{j_0, j_1, \dots, j_n \in 1 \dots N} \sum_{i=0}^{n-1} m_{j_i, j_{i+1}}^i,$$

on peut par exemple, sous réserve que les variables $\sum_{i=0}^{n-1} m_{j_i, j_{i+1}}^i$ sont associées, minorer la quantité $P(h(n) \leq n\alpha)$ par un produit de termes plus simples. Cette méthode, utilisée en conjonction avec des grandes déviations, est appliquée dans [Bacelli et al.91] afin de donner une borne supérieure pour le temps de cycle de réseaux de Petri stochastiques. En effet, les matrices correspondant aux réseaux de Petri présentent cette propriété d'association : les coefficients d'une même matrice sont associés. Les matrices représentant les systèmes de requêtes ne présentent par contre pas cette propriété d'association; quant aux matrices des systèmes de ressources, cela n'est généralement pas le cas non plus. Il semble donc difficile d'appliquer cette technique au cas que nous avons étudié.

Grandes déviations : Nous venons de parler de grandes déviations. Ces techniques interviennent aussi dans le calcul de bornes en $(\max, +)$. Le principe des grandes déviations est de majorer les queues de distributions de variables aléatoires. On vient d'en voir une application dans l'exemple ci-dessus. Plus généralement, un domaine d'application des grandes déviations est l'étude du processus des dateurs d'un système indépendant vu comme un processus de branchement. On a alors des résultats sur le comportement asymptotique de ce processus, ou sur le comportement asymptotique d'un processus majorant (on retrouve la notion de bornes stochastiques). Une borne supérieure pour $\gamma(M)$ sous l'hypothèse que $-\infty < \mathbb{E}m_{i,j}^0 < \infty$ pour tout couple (i, j) est présentée dans [Glasserman et al.95]. Dans un cadre proche de celui de cette thèse, Vincent Dumas obtient [Dumas] que pour un graphe \mathcal{G} de degré constant d et de taille N

$$\gamma(\mathcal{G}, (1/N, \dots, 1/N)) \leq \frac{1}{2c} \frac{d+1}{N} \simeq 2.16 \frac{d+1}{N}$$

avec $c = \sup\{x < 1, \ln(2x) < x - 1\}$.

Les bornes que nous présentons dans cette thèse sont essentiellement basées sur des techniques de comparaison stochastique, à l'exception de la borne présentée dans la partie 3.6.1 qui provient de techniques plus proches des grandes déviations.

3.1.3 Simulation de systèmes indépendants

Un moyen simple d'obtenir une valeur approchée de γ pour un système indépendant est la simulation. Celle-ci est de type équationnel : l'état du système à un instant donné est entièrement déterminé par les valeurs de ses dateurs à cet instant. Par ailleurs, son évolution est entièrement déterminée par les équations d'évolution. On se contente donc de faire une suite de tirages de requêtes et d'en déduire la suite des valeurs des dateurs. Une simulation permet donc de produire une réalisation de la suite $\{h(n)/n\}_{n \geq 1}$. La convergence presque sûre de cette suite vers γ implique qu'en "attendant suffisamment" la valeur courante de $h(n)/n$ sera "très proche" de γ .

La difficulté consiste à préciser ce qu'on entend par "très proche" : quelle est l'erreur induite par la simulation ? Peut-on donner un intervalle de confiance pour γ après une simulation ? La réponse théorique à cette question est "oui". En effet, on verra dans la partie 3.2.3 que la suite $\{(h(n) - n\gamma)/\sqrt{n}\}_{n \geq 1}$ converge vers une loi normale d'espérance 0 et de variance σ^2 . Le problème réside dans l'estimation de σ^2 . En effet, l'expression de cette quantité n'est pas simple. Il est possible de la majorer de façon brutale (on reviendra sur ce point dans la partie 3.2.3), mais en pratique, une telle majoration est bien trop excessive. Pour cette raison, on obtient un intervalle de confiance par trop pessimiste. Nous allons donc présenter une autre façon de procéder. C'est cette méthode que nous avons implémentée afin d'obtenir les valeurs simulées de γ qui sont données dans cette thèse.

La méthode de simulation que nous avons employée est basée sur les deux propositions suivantes :

Proposition 8 *Pour tout n ,*

$$\frac{\mathbf{IE}h(n)}{n} \leq \gamma \leq \frac{\mathbf{IE}h(n)}{n}.$$

Proposition 9 *Il existe δ une constante positive telle que*

$$\lim_{n \rightarrow \infty} \mathbf{IE}h(n) - \mathbf{IE}h(n) = \delta.$$

La première proposition est une conséquence directe de la sous-additivité de la suite $\{h(n)/n\}_{n \geq 1}$ et de la sur-additivité de la suite $\{\underline{h}(n)/n\}_{n \geq 1}$ [Vincent93]. La deuxième

proposition est une conséquence des propositions 5 et 6. Afin d'estimer γ pour un système donné on procède donc de la façon suivante :

- On choisit n_0 de telle façon que δ/n_0 soit suffisamment faible. Comme on ne connaît pas la valeur de δ , on procède par essais successifs. En pratique, pour de petits systèmes, si l'on cherche à avoir une précision de 10^{-2} sur γ , prendre n_0 égal à 1000 ou 10000 conviendra parfaitement.
- On estime σ_0 (resp. $\underline{\sigma}_0$) la variance de $h(n_0)/n_0$ (resp. $\underline{h}(n_0)/n_0$). En pratique, on réalise suffisamment de tirages indépendants des variables $h(n_0)/n_0$ et $\underline{h}(n_0)/n_0$ pour que les variances empiriques se stabilisent.
- On réalise n_1 tirages indépendants des variables $h(n_0)/n_0$ et $\underline{h}(n_0)/n_0$ afin d'obtenir un estimateur γ_0 (resp. $\underline{\gamma}_0$) de $h(n_0)/n_0$ (resp. $\underline{h}(n_0)/n_0$). On aura alors par exemple [Ross91]

$$\underline{\gamma}_0 - 2.97 \frac{\underline{\sigma}_0}{\sqrt{n_1}} \leq \gamma \leq \gamma_0 + 2.97 \frac{\sigma_0}{\sqrt{n_1}}$$

avec probabilité 0.997. En pratique, le choix de n_1 se fait donc en fonction de la précision souhaitée pour γ .

L'étape la plus longue de ce procédé de simulation est en fait l'étape de stabilisation des variances empiriques. En pratique, à l'exception des résultats de la partie 4.2, les valeurs simulées de γ que l'on présente dans cette thèse correspondent à de petits systèmes. Il était donc possible d'obtenir une précision sur γ meilleure que 10^{-2} . Ainsi, en l'absence de précision, la notation " $\gamma = x$ " pour un résultat de simulation devra se lire " $|\gamma - x| \leq 5 \cdot 10^{-3}$ avec probabilité 0.997". De même, les valeurs numériques des bornes présentées sont implicitement données avec une précision de $5 \cdot 10^{-3}$.

3.2 Espace projectif

Cette partie propose une première approche du calcul de γ basée sur la notion d'espace projectif.

3.2.1 Une remarque simple

Pour un système stochastique quelconque,

$$\begin{aligned} \frac{\mathbb{E}h(n)}{n} &= \frac{1}{n} \sum_{k=0}^{n-1} \mathbb{E}(h(k+1) - h(k)), \\ &= \frac{1}{n} \sum_{k=0}^{n-1} P(h(k+1) - h(k) = 1). \end{aligned}$$

En définissant $I(\mathbf{S}, n)$ l'incrément moyen de $h(\mathbf{S}, n)$:

Définition 3.2.1 Pour un système stochastique \mathbf{S} ,

$$I(\mathbf{S}, n) \stackrel{\text{def}}{=} P(h(\mathbf{S}, n+1) - h(\mathbf{S}, n) = 1).$$

On peut tirer de cette reformulation la proposition suivante :

Proposition 10 Soit \mathbf{S} un système stochastique tel que $I(\mathbf{S}, n)$ converge quand n tend vers l'infini. Alors, $\gamma(\mathbf{S})$ est défini et est égal à

$$\gamma(\mathbf{S}) = \lim_{n \rightarrow \infty} I(\mathbf{S}, n).$$

Preuve : Immédiate, d'après le théorème de Cesàro. □

3.2.2 Chaînes de Markov $\mathcal{M}(\text{Res}, \mathbf{S})$ et $\mathcal{M}(\text{Req}, \mathbf{S})$

Soit \mathbf{S} un système de ressources. On définit $\mathcal{M}^0(\text{Res}, \mathbf{S})$ la chaîne de Markov suivante :

- les états de $\mathcal{M}^0(\text{Res}, \mathbf{S})$ sont les valeurs possibles du vecteur $\mathbf{T}(\mathbf{S}, n)$ pour tous les n . Étant donné un sommet x de $\mathcal{M}^0(\text{Res}, \mathbf{S})$, on note $T(x)$ le vecteur qui lui est associé.
- On a une transition avec probabilité α de x vers x' si et seulement si

$$P(\mathbf{T}(n+1) = T(x') | \mathbf{T}(n) = T(x)) = \alpha \text{ pour tout } n.$$

Si on note x_0^0 le sommet de $\mathcal{M}^0(\text{Res}, \mathbf{S})$ associé à $\mathbf{T}(0)$ et X_n^0 le sommet atteint après n transitions en partant de x_0^0 , on a

$$I(n) = \sum_{x \in \mathcal{M}^0(\text{Res}, \mathbf{S})} P(X_n^0 = x) I(x),$$

où

$$\begin{aligned} I(x) &= P(h(n+1) - h(n) = 1 | X_n^0 = x), \\ &= \sum_{v \in \text{Req}(\mathbf{S})} P(v) P(h(n+1) - h(n) = 1 | X_n^0 = x \wedge s(n+1) = v). \end{aligned} \quad (3.3)$$

La chaîne $\mathcal{M}^0(\text{Res}, \mathbf{S})$ n'admet pas, par construction, de mesure stationnaire. Nous allons cependant montrer qu'il est possible de construire un processus $(X_n)_{n \geq 0}$ sur une chaîne $\mathcal{M}(\text{Res}, \mathbf{S})$ admettant une mesure stationnaire de telle façon que $\gamma(\mathbf{S})$ s'exprime en fonction de cette mesure stationnaire. Le processus $(X_n)_{n \geq 0}$ est tout simplement la projection de $(X_n^0)_{n \geq 0}$ sur l'espace projectif IPN^K [Mairesse93a]. Soit \mathcal{R} la relation d'équivalence suivante sur les sommets de $\mathcal{M}^0(\text{Res}, \mathbf{S})$:

$$x \mathcal{R} x' \text{ si et seulement si } \exists \lambda \text{ t.q. } T(x) = T(x') + (\lambda, \dots, \lambda).$$

Deux choses sont claires à propos de \mathcal{R} . D'une part, l'expression 3.3 donne

$$x \mathcal{R} x' \Rightarrow I(x) = I(x').$$

D'autre part,

$$(x \mathcal{R} x', y \mathcal{R} y', x \rightarrow_\alpha y) \Rightarrow (x' \rightarrow_\alpha y').$$

On peut donc définir $\mathcal{M}(\text{Res}, \mathbf{S})$ le quotient de $\mathcal{M}^0(\text{Res}, \mathbf{S})$ par \mathcal{R} :

$$\mathcal{M}(\text{Res}, \mathbf{S}) = \mathcal{M}^0(\text{Res}, \mathbf{S}) / \mathcal{R}.$$

Si on note maintenant x_0 le sommet de $\mathcal{M}(\text{Res}, \mathbf{S})$ associé à $\mathbf{T}(0)$ et X_n le sommet atteint après n transitions en partant de x_0 dans $\mathcal{M}(\text{Res}, \mathbf{S})$ (X_n est donc le projeté par \mathcal{R} de X_n^0), on a

$$I(n) = \sum_{x \in \mathcal{M}(\text{Res}, \mathbf{S})} P(X_n = x) I(x).$$

Proposition 11 *Si \mathbf{S} est tel que pour tout $r \in \text{Res}(\mathbf{S})$ il existe $v \in \text{Req}(\mathbf{S})$ tel que $r \in v$, alors la chaîne de Markov $\mathcal{M}(\text{Res}, \mathbf{S})$ admet une unique mesure stationnaire.*

Preuve : $\mathcal{M}(\text{Res}, \mathbf{S})$ est homogène par construction. Il suffit de montrer qu'il existe un sommet x_R de $\mathcal{M}(\text{Res}, \mathbf{S})$, un réel α strictement positif et un entier t tels que

$$\begin{aligned} P(X_t = x_R | X_0 = x) &\geq \alpha \text{ pour tout } x, \\ P(X_{t+1} = x_R | X_0 = x) &\geq \alpha \text{ pour tout } x. \end{aligned}$$

Ces conditions sont en effet suffisantes pour garantir l'existence d'une unique mesure stationnaire pour $\mathcal{M}(\text{Res}, \mathbf{S})$ (voir par exemple [Asmussen87], Chapitre 6). Les vérifier est immédiat une fois constaté que si l'on considère $M = (v_{\alpha_1}, \dots, v_{\alpha_m})$ un motif de \mathbf{S} , la propriété de perte de mémoire des motifs implique que la fonction

$$x \rightarrow x_R(x) = (x_{n+m+N} | (x_n = x) \wedge s(n+1), \dots, s(n+m+N) = M, v_1, \dots, v_N)$$

est constante égale à $x_R(x_0)$. Il suffit donc de prendre

$$\begin{aligned} x_R &= x_R(x_0), \\ \alpha &= \prod_{i=1}^m P(v_{\alpha_i}) \prod_{i=1}^N P(v_i), \\ t &= m + N. \end{aligned}$$

□

Remarque 1 : La condition que toute ressource soit requise par au moins un type de requête est naturelle. Si jamais elle n'est pas vérifiée, il suffit de ne plus considérer les ressources inutilisées (qui n'ont aucune influence sur la dynamique du système) pour qu'elle le soit.

Si on note $\mu_{\mathcal{M}(\text{Res}, \mathbf{S})}$ la mesure stationnaire de $\mathcal{M}(\text{Res}, \mathbf{S})$, on obtient

$$\lim_{n \rightarrow \infty} I(n) = \sum_{x \in \mathcal{M}(\text{Res}, \mathbf{S})} \mu_{\mathcal{M}(\text{Res}, \mathbf{S})}(x) I(x),$$

et donc d'après la Proposition 10

$$\gamma(\mathbf{S}) = \sum_{x \in \mathcal{M}(\text{Res}, \mathbf{S})} \mu_{\mathcal{M}(\text{Res}, \mathbf{S})}(x) I(x). \quad (3.4)$$

Remarque 2 : Les termes dont l'évaluation pose problème dans cette formule sont les $\mu_{\mathcal{M}(\text{Res}, \mathbf{S})}(x)$. En effet, pour tout x , $I(x)$ se calcule aisément à partir de

$$\begin{aligned} I(x) &= P(h(n+1) - h(n) = 1 | x_n = x), \\ &= \sum_{v \in \text{Req}(\mathbf{S})} P(v) P(h(n+1) - h(n) = 1 | x_n = x \wedge s(n+1) = v). \end{aligned}$$

Remarque 3 : Pour un système indépendant \mathbf{S} , on peut définir $\mathcal{M}(\text{Req}, \mathbf{S})$ sur le même principe. Sans redétailler toutes les étapes de la construction de cette chaîne, donnons-en tout de même les principaux points :

- On construit $\mathcal{M}^0(\text{Req}, \mathbf{S})$. Les sommets de $\mathcal{M}^0(\text{Req}, \mathbf{S})$ correspondent aux valeurs possibles du vecteur $\mathbf{C}(\mathbf{S}, n)$ pour toutes les valeurs de n et les transitions de $\mathcal{M}^0(\text{Req}, \mathbf{S})$ représentent les probabilités d'évolution de ces vecteurs;
- On définit \mathcal{R} comme précédemment et $\mathcal{M}(\text{Req}, \mathbf{S})$ le quotient de $\mathcal{M}^0(\text{Req}, \mathbf{S})$ par \mathcal{R} ;
- On pose $I(x) = P(h(n+1) - h(n) = 1 | x_n = x)$.

On obtient alors l'expression suivante pour $\gamma(\mathbf{S})$:

$$\gamma(\mathbf{S}) = \sum_{x \in \mathcal{M}(\text{Req}, \mathbf{S})} \mu_{\mathcal{M}(\text{Req}, \mathbf{S})}(x) I(x).$$

Nous ne détaillons pas plus cette deuxième reformulation de γ . Elle se prouve sur le même principe que l'équation 3.4. Les deux expressions que nous venons de donner, sauf cas particulier, se contentent de déplacer la difficulté intrinsèque du calcul de γ . On verra cependant dans la partie 3.2.4 une application de la première formule permettant un calcul exact de γ pour tout système basé sur deux ressources. Dans la partie 3.2.5, nous verrons comment "tronquer" la chaîne $\mathcal{M}(\text{Res}, \mathbf{S})$ afin de majorer $\gamma(\mathbf{S})$.

3.2.3 Un théorème central limite

Nous montrons dans cette partie comment la chaîne $\mathcal{M}(\text{Req}, \mathbf{S})$ permet d'énoncer un théorème central limite pour les suites $\{h(n)/n\}_{n \geq 1}$ et $\{\underline{h}(n)/n\}_{n \geq 1}$.

Théorème 2 *Il existe une constante positive ou nulle σ^2 telle que*

$$\frac{h(n) - n\gamma}{\sqrt{n}} \rightarrow \mathcal{N}(0, \sigma^2) \quad \text{et} \quad \frac{\underline{h}(n) - n\gamma}{\sqrt{n}} \rightarrow \mathcal{N}(0, \sigma^2).$$

Par convention, on note $\mathcal{N}(0, 0)$ une loi ayant toute sa masse en 0.

Ce théorème a déjà été énoncé dans un cadre $(\max, +)$ pour le cas des matrices de taille 2 à coefficients positifs [Resing et al.90]. Les auteurs précisent que leur démonstration se généralise sans difficulté autre que celle des notations à des matrices de taille quelconque et que, par ailleurs, l'hypothèse de "positivité" des matrices pourrait être levée en y ajoutant une hypothèse de connexité (cette hypothèse correspond en fait à l'hypothèse de connexité que nous avons faite sur les systèmes que nous considérons). Leur démonstration s'appuie sur un théorème central limite pour les processus $J-X$ présenté dans [Obrien74]. Nous allons donc montrer comment ce théorème s'applique au cas qui nous intéresse. Nous commençons donc par le rappeler. Les notations ont été adaptées afin de correspondre à celles de cette thèse.

Théorème 3 *Soit $\{X_n\}_{n \geq 0}$ une chaîne de Markov irréductible, apériodique et positive récurrente sur un espace d'états au plus dénombrable. Soit $\{H_n\}_{n \geq 1}$ une suite de variables aléatoires réelles telles que*

$$\begin{aligned} & P(X_n = j, H_n \leq x | X_0, H_1, X_1, \dots, H_{n-1}, X_{n-1} = i) \\ &= P(X_n = j, H_n \leq x | X_{n-1} = i). \end{aligned}$$

On dit que le processus $\{X_n, H_n\}$ est un processus $J-X$. Soit $\{Y_n, K_n\}$ la version stationnaire de $\{X_n, H_n\}$. On suppose qu'il existe $\delta > 0$ et g une fonction définie sur \mathbb{N} telle que

1. $\lim_k g(k) = 0,$
2.
$$\left. \begin{array}{l} A \in \mathcal{B}(J_1, \dots, J_n) \\ B \in \mathcal{B}(J_{n+k+1}, J_{n+k+2}, \dots) \end{array} \right\} \Rightarrow |P(A \cap B) - P(A)P(B)| \leq g(k),$$

$$3. \sum_k g(k)^{\frac{\delta}{2+\delta}} < \infty.$$

Enfin, on suppose que $\mathbb{E}|K_1|^{2+\delta} < \infty$. Sous ces hypothèses, la quantité

$$\sigma^2 = \text{Cov}(K_1, K_1) + 2 \sum_{i \geq 2} \text{Cov}(K_1, K_i)$$

est définie et positive et si $\sigma^2 \neq 0$ alors $0 < \sigma^2 < \infty$ et

$$\frac{H_1 + \dots + H_n - n\mathbb{E}K_1}{\sigma\sqrt{n}} \rightarrow \mathcal{N}(0, 1).$$

Ce théorème s'applique maintenant de la façon suivante : on prend comme processus X_n le processus défini sur la chaîne $\mathcal{M}(\text{Req}, \mathbf{S})$ et $H_n = h(n) - h(n-1)$. La donnée de X_n et X_{n-1} permet de déterminer $s(n)$ et donc H_n . Le processus $\{X_n, H_n\}$ est donc un processus $J-X$, la chaîne $\mathcal{M}(\text{Req}, \mathbf{S})$ étant apériodique, irréductible et positive récurrente. On considère $\{Y_n, K_n\}$ la version stationnaire de $\{X_n, H_n\}$. Ce processus étant stationnaire, K_n suit une loi de Bernoulli de paramètre γ . Nous allons maintenant construire $g(k)$. Pour cela, on note comme précédemment M un motif, m la taille de ce motif. On introduit

$$p_M = P((s(1), \dots, s(m)) = M)$$

la probabilité qu'un tirage de m requêtes successives constitue un motif M . Pour $A \in \mathcal{B}(J_1, \dots, J_n)$ et $B \in \mathcal{B}(J_{n+k+1}, J_{n+k+2}, \dots)$, la propriété de perte de mémoire des motifs implique que les événements A et B sont indépendants pour peu que soit apparu un motif parmi les requêtes $(n+1)$ à $(n+k)$. On note E_k un tel événement et $p(k)$ sa probabilité, on a

$$1 - p(k) \leq (1 - p_M)^{\lfloor k/m \rfloor}.$$

On a donc

$$\begin{aligned} P(A \cap B) &= P(A \cap B | E_k)p(k) + P(A \cap B | \neg E_k)(1 - p(k)) \\ &= P(A)p(B)p(k) + P(A \cap B | \neg E_k)(1 - p(k)) \end{aligned}$$

$$\begin{aligned} P(A)P(B)p(k) &\leq P(A \cup B) && \leq P(A)p(B)p(k) + (1 - p(k)) \\ P(A)P(B)(p(k) - 1) &\leq P(A \cap B) - P(A)P(B) && \leq (1 - P(A)P(B))(1 - p(k)) \\ &|P(A \cup B) - P(A)P(B)| && \leq 1 - p(k) \end{aligned}$$

On peut donc prendre $g(k) = (1 - p_M)^{\lfloor k/m \rfloor}$. En choisissant maintenant par exemple $\delta = 1$, on a $\sum g(k)^{\frac{\delta}{2+\delta}} < \infty$ et $\mathbb{E}|K_1|^3 < \infty$, ce qui implique que σ^2 est défini. Pour peu que σ^2 soit non nul on peut donc appliquer le théorème. Le cas $\sigma^2 = 0$ ne pose pas de problème. En effet, si on définit $\sigma_n^2 = \text{Var}(K_0 + \dots + K_{n-1})$, il est prouvé dans [Ibragimov62] que $\sigma_n^2 = \sigma^2 n(1 + o(1))$. En particulier $\sigma^2 = 0$ implique que $\text{Var}(K_0 + \dots + K_{n-1}) = 0$ pour tout n et donc $\text{Var}(h(n)) = 0$ pour tout n . On retrouve le théorème.

Maintenant que nous avons prouvé celui-ci pour la suite $\{h(n)/n\}_{n \geq 1}$, le prouver pour $\{\underline{h}(n)/n\}_{n \geq 1}$ est immédiat une fois rappelé que $\{h(n) - \underline{h}(n)\}_{n \geq 0}$ converge en loi vers une loi de probabilité d'espérance finie (voir partie 3.1.2).

Remarque 1 : On peut majorer σ^2 en utilisant le fait que les variables K_1 et K_i sont indépendantes pour peu que soit apparu un motif parmi les requêtes 2 à $i - 1$, ce qui implique que $\mathbb{E}(K_1 K_i | E_{i-2}) = 0$. On obtient alors

$$\begin{aligned}
 \sigma^2 &= \text{Cov}(K_1, K_1) + 2 \sum_{i \geq 2} \text{Cov}(K_1, K_i) \\
 &= \gamma(1 - \gamma) + 2 \sum_{i \geq 2} \mathbb{E}K_1 K_i \\
 &= \gamma(1 - \gamma) + 2 \sum_{i \geq 2} [\mathbb{E}(K_1 K_i | E_{i-2}) p(i-2) + \mathbb{E}(K_1 K_i | \neg E_{i-2}) (1 - p(i-2))] \\
 &\leq \gamma(1 - \gamma) + 2 \sum_{i \geq 2} (1 - p(i-2)) \\
 &\leq 1/4 + 2 \sum_{i \geq 0} (1 - p_M)^{\lfloor \frac{i}{m} \rfloor}
 \end{aligned}$$

Malheureusement, cette majoration est très grossière : Pour un système donné, et M un motif de ce système, p_M est généralement très faible, ce qui fait que la borne supérieure que nous venons d'obtenir est très grande. Cependant, pour de petits systèmes, elle offre l'avantage de pouvoir donner un intervalle de confiance rigoureux pour une valeur simulée de γ .

3.2.4 Cas d'un système comportant exactement deux ressources

Nous présentons dans cette partie le calcul du codébit de tout système indépendant basé sur deux ressources. Nous adopterons une vision "ressources", car celle-ci est plus adaptée à ce cas particulier. Si l'on note r_1 et r_2 les deux ressources, il y a donc 4 types de requêtes possibles :

$$v_1 = \{r_1\}, \quad v_2 = \{r_2\}, \quad v_3 = \{r_1, r_2\}, \quad v_4 = \emptyset.$$

On a $\sum_{i=1}^4 p_i = 1$. On peut supposer que p_4 est nul, étant donné que

$$\gamma(p_1, p_2, p_3, p_4) = (p_1 + p_2 + p_3) \gamma\left(\frac{p_1}{p_1 + p_2 + p_3}, \frac{p_2}{p_1 + p_2 + p_3}, \frac{p_3}{p_1 + p_2 + p_3}, 0\right).$$

On s'intéresse donc à des systèmes tels que $\sum_{i=1}^3 p_i = 1$. On note un tel système $\mathbf{S}(p_1, p_2, p_3)$. On peut remarquer que si $p_3 = 0$, $\mathbf{S}(p_1, p_2, p_3)$ est en fait l'union de deux systèmes, chacun basé sur une ressource, une requête demandant cette ressource avec probabilité p_1 dans le premier système et p_2 dans le deuxième système. On a donc $\gamma = \max(p_1, p_2)$. Nous allons maintenant nous intéresser au cas où p_3 est strictement positif.

Proposition 12 *Si $p_3 > 0$,*

$$\gamma(\mathbf{S}(p_1, p_2, p_3)) = \frac{1}{2} \frac{p_1 + p_2 - 4p_1p_2 + (2 - p_1 - p_2)\sqrt{1 - 4p_1p_2}}{\sqrt{1 - 4p_1p_2}}.$$

Preuve : L'expression 3.4 va nous permettre de calculer γ . Les figures 3.1 et 3.2 montrent respectivement une partie des chaînes de Markov $\mathcal{M}^0(\text{Res}, \mathbf{S})$ et $\mathcal{M}(\text{Res}, \mathbf{S})$.

Il est facile de vérifier que pour $i > 0$,

$$\begin{aligned} I((i, 0)) &= p_1 + p_3, \\ I((0, 0)) &= 1, \\ I((0, i)) &= p_2 + p_3. \end{aligned}$$

En posant $\mu_i = \mu((i, 0))$, l'équation 3.4 se réécrit :

$$\gamma(\mathbf{S}) = \sum_{i=-\infty}^{-1} \mu_i(p_2 + p_3) + \mu_0 + \sum_{i=1}^{\infty} \mu_i(p_1 + p_3). \quad (3.5)$$

Les μ_i se calculent facilement. En effet, les équations de balance sur $\mathcal{M}(\text{Res}, \mathbf{S})$ donnent :

$$\begin{aligned} \mu_0 &= p_1\mu_{-1} + p_2\mu_1 + p_3, \\ \mu_i &= p_1\mu_{i-1} + p_2\mu_{i+1} \text{ si } i \neq 0. \end{aligned}$$

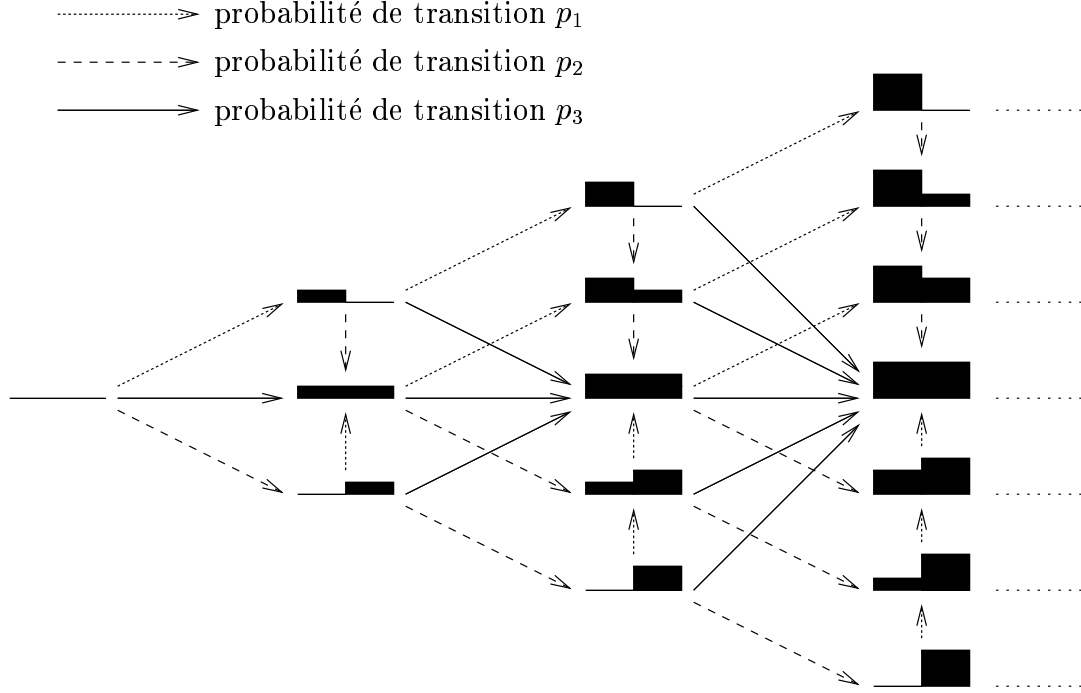


FIG. 3.1 – Une partie de $\mathcal{M}^0(\text{Res}, \mathbf{S})$. Les valeurs des composantes d'un vecteur sont représentées par les hauteurs de deux colonnes.

Pour $i > 0$ on a donc

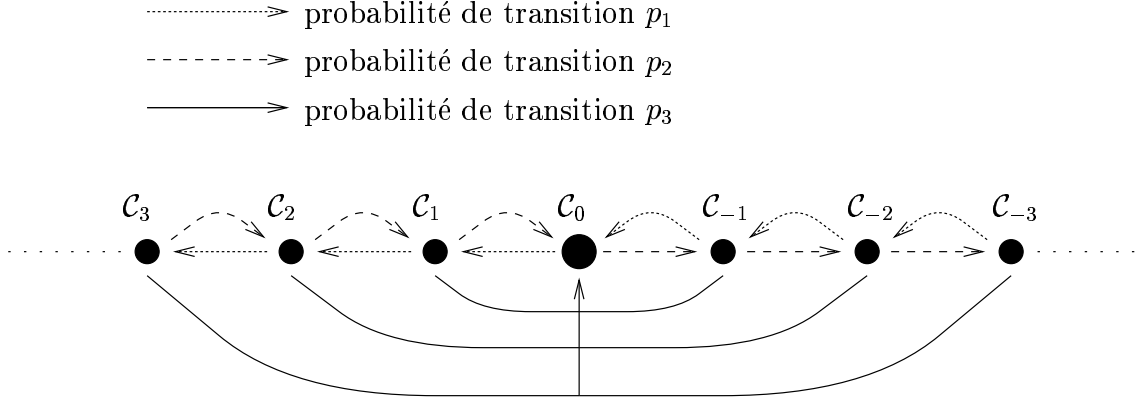
$$\begin{aligned}\mu_i &= \mu_0 \lambda_+^i, \\ \mu_{-i} &= \mu_0 \lambda_-^i,\end{aligned}$$

avec $\lambda_+ = \frac{1 - \sqrt{1 - 4p_1 p_2}}{2p_2}$ et $\lambda_- = \frac{1 - \sqrt{1 - 4p_1 p_2}}{2p_1}$. Comme $\sum_i \mu_i = 1$, on a

$$\mu_0 = \left(1 + \frac{\lambda_+}{1 - \lambda_+} + \frac{\lambda_-}{1 - \lambda_-} \right)^{-1}.$$

L'équation 3.5 se récrit donc

$$\gamma = \frac{1}{1 + \frac{\lambda_+}{1 - \lambda_+} + \frac{\lambda_-}{1 - \lambda_-}} \left(1 + \frac{(p_1 + p_3)\lambda_+}{1 - \lambda_+} + \frac{(p_2 + p_3)\lambda_-}{1 - \lambda_-} \right).$$


 FIG. 3.2 – Une partie de $\mathcal{M}(\text{Res}, \mathbf{S})$.

Après simplification on obtient

$$\gamma = \frac{1}{2} \frac{p_1 + p_2 - 4p_1p_2 + (2 - p_1 - p_2)\sqrt{1 - 4p_1p_2}}{\sqrt{1 - 4p_1p_2}}.$$

□

On peut remarquer que, même pour ce cas simple, l'expression de γ est assez compliquée.

Remarque 1 : Le système que l'on a étudié correspond à un système de requêtes défini par le graphe étoile à deux branches, le sommet v_3 étant le centre de l'étoile. Si l'on note E_2 l'étoile à deux branches, on a donc

$$\gamma(E_2, \mathbf{p}) = \frac{1}{2} \frac{p_1 + p_2 - 4p_1p_2 + (2 - p_1 - p_2)\sqrt{1 - 4p_1p_2}}{\sqrt{1 - 4p_1p_2}}. \quad (3.6)$$

3.2.5 Bornes supérieures obtenues par troncature

Soit \mathbf{S} un système indépendant de ressources tel que toute ressource soit requise par au moins un type de requête (on a déjà vu que cette hypothèse n'est pas contraignante). Nous présentons dans cette partie une méthode qui permet de calculer une suite de bornes

supérieures du codébit de \mathbf{S} . Cette suite sera notée $\{u_m(\mathbf{S})\}_{m \geq 1}$. Le calcul de u_m se fait en “tronquant” (cette troncature étant paramétrée par m) la chaîne $\mathcal{M}(\text{Res}, \mathbf{S})$. En effet, pour un système \mathbf{S} quelconque, la mesure stationnaire de $\mathcal{M}(\text{Res}, \mathbf{S})$ ne peut pas être calculée analytiquement. En la tronquant, c’est-à-dire en la majorant (en un certain sens) par des chaînes sur des espaces d’états finis, on va se ramener à l’évaluation de mesures stationnaires de chaînes finies.

Étant donné un système de ressources indépendant \mathbf{S} et un entier m strictement positif, on note v_s une requête demandant toutes les ressources de \mathbf{S} et on considère le système \mathbf{S}_m obtenu par couplage avec \mathbf{S} de la façon suivante :

- si $(\max_i T_i(\mathbf{S}_m, n+1) - \min_i T_i(\mathbf{S}_m, n+1)) > m | s(\mathbf{S}_m, n+1) = s(\mathbf{S}, n+1)$
alors $s(\mathbf{S}_m, n+1) = v_s$;
- si $(\max_i T_i(\mathbf{S}_m, n+1) - \min_i T_i(\mathbf{S}_m, n+1)) \leq m | s(\mathbf{S}_m, n+1) = s(\mathbf{S}, n+1)$
alors $s(\mathbf{S}_m, n+1) = s(\mathbf{S}, n+1)$.

On construit donc \mathbf{S}_m en reprenant les requêtes de \mathbf{S} telles qu’elles arrivent, à ceci près que l’on remplace par v_s toute requête qui ferait passer la différence entre le plus grand dateur et le plus petit dateur à plus que m . Le système \mathbf{S}_m majore donc trajectoriellement \mathbf{S} .

Cette dynamique se traduit de la façon suivante sur la chaîne $\mathcal{M}(\text{Res}, \mathbf{S}_m)$: si on note S^m l’ensemble des sommets x de $\mathcal{M}(\text{Res}, \mathbf{S})$ tels que

$$\max_i T_i(x) - \min_i T_i(x) > m,$$

$\mathcal{M}(\text{Res}, \mathbf{S}_m)$ est obtenue en remplaçant dans $\mathcal{M}(\text{Res}, \mathbf{S})$ chaque transition $x \rightarrow_\alpha x'$ avec $x \notin S^m$ et $x' \in S^m$ par une transition $x \rightarrow_\alpha (0, \dots, 0)$. Les sommets dans S^m ne pouvant être atteints dans $\mathcal{M}(\text{Res}, \mathbf{S}_m)$ en partant du point $(0, \dots, 0)$, on peut les supprimer. Il est donc facile de voir que la chaîne $\mathcal{M}(\text{Res}, \mathbf{S}_m)$ est finie, ce qui rend possible le calcul exact de sa mesure stationnaire. Une fois cette mesure stationnaire calculée, il est possible d’utiliser l’équation 3.4 afin de calculer

$$u_m(\mathbf{S}) = \gamma(\mathbf{S}_m).$$

Comme \mathbf{S}_m majore trajectoriellement \mathbf{S} , on aura de plus

$$\gamma(\mathbf{S}) \leq u_m(\mathbf{S}).$$

Nous allons maintenant montrer que la suite $\{u_m(\mathbf{S})\}_{m \geq 1}$ converge vers $\gamma(\mathbf{S})$ quand m tend vers l'infini.

Théorème 4 $\lim_{m \rightarrow \infty} u_m(\mathbf{S}) = \gamma(\mathbf{S})$.

Preuve : Soit τ_m le rang de la première requête v_s dans \mathbf{S}_m générée par une transition (dans $\mathcal{M}(\text{Res}, \mathbf{S})$) entrant dans S^m . Un argument de type “découpage en blocs”, identique à celui présenté dans la preuve du théorème 7, permet d'écrire :

$$\gamma(\mathbf{S}_m) = \frac{1 + \mathbb{E}h(\mathbf{S}, \tau_m - 1)}{E\tau_m}.$$

Pour tout m , τ_m est presque sûrement fini, et donc $(\tau_m)_{m \geq 1}$ est presque sûrement définie. Par ailleurs, $\tau_m \geq m$. Soit $z_N = \sup_{n \geq N} \frac{h(\mathbf{S}, n)}{n}$. z_N tend presque sûrement vers $\gamma(\mathbf{S})$. Soit c tel que $\gamma(\mathbf{S}) < c < 1$ (le cas $\gamma(\mathbf{S}) = 1$ est trivial et sans intérêt). On a donc

$$\begin{aligned} \forall \varepsilon > 0, \exists M \geq 0, \forall N \geq M & \quad P(z_N > c) < \varepsilon, \\ \forall \varepsilon > 0, \exists M \geq 0, \forall N \geq M & \quad P\left(\exists n \geq N, \frac{h(\mathbf{S}, n)}{n} > c\right) < \varepsilon, \\ \forall \varepsilon > 0, \exists M \geq 0 & \quad P\left(\exists n \geq M, \frac{h(\mathbf{S}, n)}{n} > c\right) < \varepsilon, \\ \forall \varepsilon > 0, \exists M \geq 0 & \quad P\left(\forall n \geq M, \frac{h(\mathbf{S}, n)}{n} \leq c\right) \geq 1 - \varepsilon, \\ \forall \varepsilon > 0, \exists M \geq 0 & \quad P\left(\forall n \geq M, \frac{h(\mathbf{S}, n)}{n} \leq c\right) \wedge ((\tau_m)_{m \geq 1} \text{ définie}) \geq 1 - \varepsilon, \\ \forall \varepsilon > 0, \exists M \geq 0 & \quad P\left(\forall m \geq M, \frac{h(\mathbf{S}, \tau_m)}{\tau_m} \leq c\right) \geq 1 - \varepsilon, \\ \forall \varepsilon > 0, \exists M \geq 0, \forall m \geq M & \quad P(h(\mathbf{S}, \tau_m) \leq c\tau_m) \geq 1 - \varepsilon. \end{aligned}$$

Comme

$$\begin{aligned} \mathbb{E}h(\mathbf{S}, \tau_m) &= P(h(\mathbf{S}, \tau_m) \leq c\tau_m) \mathbb{E}(h(\mathbf{S}, \tau_m) | h(\mathbf{S}, \tau_m) \leq c\tau_m) \\ &+ P(h(\mathbf{S}, \tau_m) > c\tau_m) \mathbb{E}(h(\mathbf{S}, \tau_m) | h(\mathbf{S}, \tau_m) > c\tau_m), \end{aligned}$$

on obtient

$$\begin{aligned} \forall \varepsilon > 0, \exists M \geq 0, \forall m \geq M & \quad \mathbb{E}h(\mathbf{S}, \tau_m) \leq (1 - \varepsilon)(c\mathbb{E}\tau_m) + \varepsilon\mathbb{E}\tau_m, \\ \forall \varepsilon > 0, \exists M \geq 0, \forall m \geq M & \quad \mathbb{E}h(\mathbf{S}, \tau_m) \leq c\mathbb{E}\tau_m(1 - \varepsilon + \frac{\varepsilon}{c}). \end{aligned}$$

Ceci étant vrai pour tout $c > \gamma(\mathbf{S})$, on en déduit

$$\forall c > \gamma(\mathbf{S}), \exists M \geq 0, \forall m \geq M \quad \gamma(\mathbf{S}_m) \leq c.$$

Comme $\gamma(\mathbf{S}_m) \geq \gamma(\mathbf{S})$, on a bien

$$\lim_{m \rightarrow \infty} \gamma(\mathbf{S}_m) = \gamma(\mathbf{S}).$$

□

Nous ne donnerons pas d'application de cette borne. En effet, bien que u_m converge rapidement vers γ , les tailles des chaînes tronquées croissent elles aussi très rapidement, ce qui fait que ces bornes sont très longues à calculer, même pour de petits systèmes et de petites valeurs de m . Par ailleurs, pour de petites valeurs de m , elles sont généralement moins bonnes (et plus coûteuses en calculs) que d'autres bornes que nous présenterons plus tard. Pour ces raisons, on peut les qualifier de "mauvaises" bornes.

3.3 Notion de ligne asymptotique

Nous présentons dans cette partie une étude du comportement asymptotique de $L(\mathbf{S}, n)$ pour les systèmes uniformes. Cette étude avait permis dans un premier temps l'obtention d'une borne inférieure pour γ . Cependant, cette borne étant rendue obsolète par la borne de la partie 3.6.3, nous ne la présenterons pas dans cette thèse.

La définition de la n -ième ligne d'un système \mathbf{S} est présentée dans la partie 2.3. Pour un système \mathbf{S} déterministe, il est clair que $L(\mathbf{S}, n)$ est un élément de $\text{Ind}(\mathbf{S})$. Par contre, si \mathbf{S} est un système stochastique (ce qui correspond à l'hypothèse que nous faisons ici), $L(\mathbf{S}, n)$ est une variable aléatoire à valeurs dans $\text{Ind}(\mathbf{S})$.

Théorème 5 *Étant donné un système indépendant \mathbf{S} , il existe $L_\infty(\mathbf{S})$, variable aléatoire à valeurs dans $\text{Ind}(\mathbf{S})$, telle que $L(\mathbf{S}, n)$ converge en loi vers $L_\infty(\mathbf{S})$. $L_\infty(\mathbf{S})$ est appelée la ligne asymptotique de \mathbf{S} .*

Preuve : La preuve de ce théorème est un peu technique. Il semble en effet difficile de faire appel à des résultats généraux de convergence de processus, le processus $L(\mathbf{S}, n)$

n'étant en effet ni stationnaire ni markovien. Soit donc M un motif de taille m de \mathbf{S} . Pour $k \geq 1$, on définit

$$m(k) = \begin{cases} 1 & \text{si } (s(k), \dots, s(k+m-1)) = M, \\ 0 & \text{sinon.} \end{cases}$$

Pour $n > m$ et $k, 1 \leq k \leq n-m$, soit $E(k, n)$ l'événement :

$$E(k, n) = \{m(k) = 1\} \wedge \bigwedge_{k', k < k' \leq n-m} \{m(k') = 0\}.$$

On définit

$$E(0, n) = \left\{ \bigvee_{k, 1 \leq k \leq n-m} E(k, n) \right\}^c.$$

Enfin, on définit $E'(k)$ l'événement :

$$E'(k) = \{m(k) = 1\} \wedge \bigwedge_{k', 1 \leq k' < k} \{m(k') = 0\}.$$

L'indépendance des requêtes successives de \mathbf{S} , d'une part, et le fait que le motif M est symétrique (par définition d'un motif), d'autre part, impliquent

$$P(E(n-k+1, n)) = P(E'(k-m)) \text{ pour } m+1 \leq k \leq n.$$

Afin d'alléger les calculs qui vont suivre, on pose

$$P_0(n) = P(E(0, n))P(L(\mathbf{S}, n) = l | E(0, n)).$$

On peut maintenant écrire

$$\begin{aligned}
 P(L(\mathbf{S}, n) = l) &= P_0(n) + \sum_{k=1}^{n-m} P(E(k, n))P(L(\mathbf{S}, n) = l|E(k, n)), \\
 &= P_0(n) + \sum_{k=1}^{n-m} P(E(k, n))P(L(\mathbf{S}, n - k + 1) = l|E(1, n - k + 1)), \\
 &\quad \text{par la propriété de perte de mémoire des motifs,} \\
 &= P_0(n) + \sum_{k=m+1}^n P(E(n - k + 1, n))P(L(\mathbf{S}, k) = l|E(1, k)), \\
 &\quad \text{en réarrangeant la somme,} \\
 &= P_0(n) + \sum_{k=m+1}^n P(E'(k - m))P(L(\mathbf{S}, k) = l|E(1, k)).
 \end{aligned}$$

Comme $\lim_{n \rightarrow \infty} P(E(0, n)) = 0$, on obtient

$$\lim_{n \rightarrow \infty} P(L(\mathbf{S}, n) = l) = \sum_{k=m+1}^{\infty} P(E'(k - m))P(L(\mathbf{S}, k) = l|E(1, k)).$$

On pose donc

$$\begin{aligned}
 \alpha(l) &= \sum_{k=m+1}^{\infty} P(E'(k - m))P(L(\mathbf{S}, k) = l|E(1, k)), \\
 P(L_{\infty} = l) &= \alpha(l).
 \end{aligned}$$

□

On peut maintenant donner l'expression suivante pour γ :

Théorème 6 $\quad \gamma = \mathbb{E} \left(\frac{1}{|L_{\infty}|} \right).$

Preuve : On rappelle la proposition 3 :

$$\underline{h}(n) \leq \sum_{k=1}^n \frac{1}{|L(k)|} \leq h(n).$$

On a donc, d'une part,

$$\begin{aligned}
 \mathbb{E}h(n) &\geq \sum_{k=1}^n \mathbb{E} \frac{1}{|L(k)|}, \\
 &\geq \sum_{k=1}^n \sum_{l \in \text{Ind}(\mathbf{s})} P(L(k) = l) \frac{1}{|l|}, \\
 &\geq \sum_{l \in \text{Ind}(\mathbf{s})} \sum_{k=1}^n P(L(k) = l) \frac{1}{|l|}, \\
 &\geq \sum_{l \in \text{Ind}(\mathbf{s})} \frac{1}{|l|} \sum_{k=1}^n P(L(k) = l).
 \end{aligned}$$

En divisant les deux membres de l'inégalité par n et en faisant tendre n vers l'infini, on obtient

$$\gamma \geq \mathbb{E} \frac{1}{|L_\infty|}.$$

En reprenant le même raisonnement avec $\underline{h}(n)$, on obtient $\gamma \leq \mathbb{E} \frac{1}{|L_\infty|}$. \square

Énonçons maintenant un résultat portant sur les quantités $\alpha(l)$.

Proposition 13 *Pour tout $v \in V(\mathcal{G})$,*

$$\sum_{l, v \in l} \frac{\alpha(l)}{|l|} = P(v).$$

Preuve : Soit

$$\begin{aligned}
 D_n &= \{m, C_{s(m)}(m) \leq n\}, \\
 F_n &= \{m, C_{s(m)}(m) > n\}, \\
 d_n &= \max_{C_{s(m)}(m) \leq n} m, \\
 f_n &= \min_{C_{s(m)}(m) > n} m.
 \end{aligned}$$

Soit v fixé. Pour tout $l \in \text{Ind}(\mathbf{S})$ contenant v et tout $n \geq 1$,

$$\sum_{m \in D_n} \mathbb{I}_{L(m)=l} = |l| \sum_{m \in D_n} \mathbb{I}_{(L(m)=l) \wedge (s(m)=v)}$$

Chapitre 3. Systèmes indépendants

On a donc pour tout $n \geq 1$,

$$\begin{aligned} \sum_{l,v \in l} \frac{1}{|l|} \sum_{m \in D_n} \mathbb{1}_{L(m)=l} &= \sum_{l,v \in l} \sum_{m \in D_n} \mathbb{1}_{L(m)=l \wedge s(m)=v}, \\ \sum_{m \in D_n} \sum_{l,v \in l} \frac{1}{|l|} \mathbb{1}_{L(m)=l} &= \sum_{m \in D_n} \mathbb{1}_{s(m)=v}, \\ \sum_{m=1}^{d_n} \sum_{l,v \in l} \frac{1}{|l|} \mathbb{1}_{L(m)=l} &\geq \sum_{m=1}^{f_n-1} \mathbb{1}_{s(m)=v}. \end{aligned}$$

Soit $\underline{\delta} < 1/\gamma$ et $\bar{\delta} > 1/\gamma$. Comme

$$\liminf \frac{h(n)}{n} = \gamma \text{ et } \limsup \frac{h(n)}{n} = \gamma,$$

pour tout $\varepsilon > 0$, il existe N tel que pour tout $n \geq N$,

$$\begin{aligned} P((f_n - 1 \geq \underline{\delta}n) \wedge (d_n \leq \bar{\delta}n)) &\geq 1 - \varepsilon, \\ P\left(\sum_{m=1}^{\bar{\delta}n} \sum_{l,v \in l} \frac{1}{|l|} \mathbb{1}_{L(m)=l} \geq \sum_{m=1}^{\underline{\delta}n} \mathbb{1}_{s(m)=v}\right) &\geq 1 - \varepsilon, \\ \mathbb{E}\left(\sum_{m=1}^{\underline{\delta}n} \mathbb{1}_{s(m)=v} - \sum_{m=1}^{\bar{\delta}n} \sum_{l,v \in l} \frac{1}{|l|} \mathbb{1}_{L(m)=l}\right) &\leq \varepsilon \underline{\delta}n. \end{aligned}$$

On a donc montré que pour tout $\varepsilon > 0$, il existe N tel que pour tout $n \geq N$,

$$\sum_{m=1}^{\underline{\delta}n} P(s(m) = v) - \sum_{m=1}^{\bar{\delta}n} \sum_{l,v \in l} \frac{1}{|l|} P(L(m) = l) \leq \varepsilon \underline{\delta}n.$$

En divisant chaque membre de cette inégalité par n et en passant à la limite, on obtient

$$\underline{\delta}P(v) - \bar{\delta} \sum_{l,v \in l} \frac{\alpha(l)}{|l|} \leq \varepsilon \underline{\delta}.$$

Ceci étant vrai pour tout triplet $(\varepsilon, \underline{\delta}, \bar{\delta})$, on en déduit

$$P(v) \leq \sum_{l,v \in l} \frac{\alpha(l)}{|l|}.$$

De la même façon on prouve l'inégalité inverse, et donc

$$P(v) = \sum_{l, v \in l} \frac{\alpha(l)}{|l|}.$$

□

3.4 Un théorème de composition

Cette partie présente un théorème de composition permettant de ramener le calcul du codébit de certains systèmes à des calculs de codébits pour des systèmes plus petits. La partie 3.4.1 est une présentation du théorème de composition. La partie 3.4.2 en montre une application.

3.4.1 Présentation du théorème

Soit $\mathcal{G}_1, \dots, \mathcal{G}_m$ des graphes de tailles N_1, \dots, N_m . Pour $1 \leq i \leq m$,

$$V(\mathcal{G}_i) = \{v_i^1, \dots, v_i^{N_i}\}.$$

Par ailleurs, pour $1 \leq i \leq m$, soit $\mathbf{p}_i = (p_i^1, \dots, p_i^{N_i})$. On note $p_i = \|\mathbf{p}_i\|$ et on suppose que $\sum_{i=1}^m p_i = 1$. Soit $K[\mathcal{G}_1, \dots, \mathcal{G}_m]$ le graphe défini de la façon suivante :

$$V(K[\mathcal{G}_1, \dots, \mathcal{G}_m]) = \bigcup_{i=1}^m V(\mathcal{G}_i),$$

$$E(K[\mathcal{G}_1, \dots, \mathcal{G}_m]) = \left(\bigcup_{i \neq j, v \in V(\mathcal{G}_i), v' \in V(\mathcal{G}_j)} \{v, v'\} \right) \cup \left(\bigcup_{i=1}^m E(\mathcal{G}_i) \right).$$

Moins formellement, $K[\mathcal{G}_1, \dots, \mathcal{G}_m]$ est obtenu en faisant l'union des \mathcal{G}_i et en ajoutant une arête entre deux sommets appartenant à deux graphes \mathcal{G}_i différents. La figure 3.3 montre un exemple d'une telle construction.

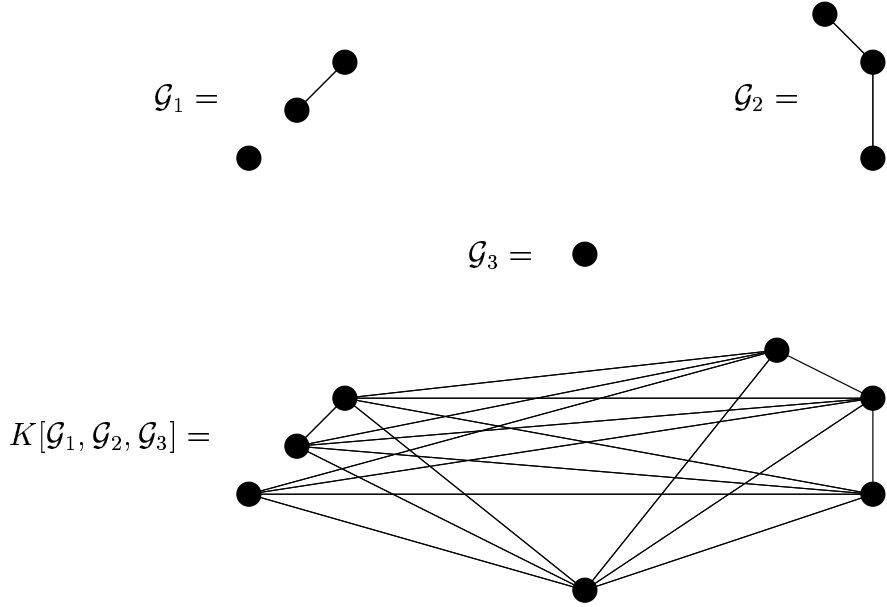


FIG. 3.3 – Trois graphes, \mathcal{G}_1 , \mathcal{G}_2 et \mathcal{G}_3 et le graphe $K[\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3]$.

Soit $\overline{\mathcal{G}}_i$:

$$\begin{aligned} V(\overline{\mathcal{G}}_i) &= v_i^0 \cup V(\mathcal{G}_i), \\ E(\overline{\mathcal{G}}_i) &= \left(\cup_{v \in V(\mathcal{G}_i)} \{v_i^0, v\} \right) \cup E(\mathcal{G}_i). \end{aligned}$$

Le graphe $\overline{\mathcal{G}}_i$ est obtenu à partir de \mathcal{G}_i en lui ajoutant un sommet que l'on relie à tous les autres. Soit $\mathbf{p} = (p_1, \dots, p_m)$ et $\overline{\mathbf{p}}_i = (1 - p_i, p_i)$. On va montrer le théorème suivant :

Théorème 7 Avec les notations précédentes,

$$\gamma(K[\mathcal{G}_1, \dots, \mathcal{G}_m], \mathbf{p}) = 1 - m + \sum_{i=1}^m \gamma(\overline{\mathcal{G}}_i, \overline{\mathbf{p}}_i).$$

Preuve : Considérons \mathbf{S} un système défini par $(K[\mathcal{G}_1, \dots, \mathcal{G}_m], \mathbf{p})$. On définit

$$\begin{aligned} \alpha_0 &= 0, \\ \alpha_{k+1} &= \max\{\text{nt.q.} \exists \mathcal{G}_j, \forall i, \alpha_k < i \leq n, s(\mathbf{S}, i) \in V(\mathcal{G}_j)\}. \end{aligned}$$

3.4 Un théorème de composition

On définit $B_i = (s(\mathbf{S}, \alpha_{i-1} + 1), \dots, s(\mathbf{S}, \alpha_i))$. On appelle B_i le i -ème bloc du système \mathbf{S} . On note

$$\begin{aligned} |B_i| &= \alpha_i - \alpha_{i-1}, \\ |B_i|_c &= h(\mathbf{S}_i, |B_i|), \end{aligned}$$

où \mathbf{S}_i désigne un système basé sur le graphe \mathcal{G}_j auquel appartiennent les requêtes de B_i et dont les $|B_i|$ premières requêtes correspondent au bloc B_i . On peut tout d'abord remarquer que

$$h(\mathbf{S}, \alpha_n) = \sum_{i=1}^n |B_i|_c.$$

Ceci vient du fait que les \mathcal{G}_i sont totalement interconnectés entre eux. On en déduit

$$\gamma(\mathcal{G}, \mathbf{p}) = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n |B_i|_c}{\sum_{i=1}^n |B_i|},$$

ce qui peut se récrire

$$\gamma(\mathcal{G}, \mathbf{p}) = \frac{E|B|_c}{E|B|}. \quad (3.7)$$

où B est la variable aléatoire correspondant à un bloc. Nous allons maintenant calculer $E|B|_c$ et $E|B|$. Pour cela on peut remarquer qu'un bloc donné peut être de m types différents: construit sur des requêtes de \mathcal{G}_1 , de $\mathcal{G}_2 \dots$ ou de \mathcal{G}_m . On note

$$B^i = (B|B \text{ est construit sur des requêtes de } \mathcal{G}_i).$$

Soit $\mu_i = P(B = B^i)$. L'équation 3.7 se récrit

$$\gamma(\mathcal{G}, \mathbf{p}) = \frac{\sum_i \mu_i E|B^i|_c}{\sum_i \mu_i E|B^i|}. \quad (3.8)$$

Les μ_i sont faciles à calculer. En effet, si l'on note

$$I(B) = i \text{ si et seulement si } B = B^i,$$

Chapitre 3. Systèmes indépendants

le processus $\{I(B_n)\}_{n \geq 1}$ est markovien, une transition de i vers j se faisant avec probabilité $\frac{p_j}{1-p_i}$. On peut donc facilement en déduire

$$\mu_i = \frac{p_i - p_i^2}{\sum_j p_j - p_j^2}.$$

Cherchons maintenant à calculer $E|B^i|$ et $E|B^i|_c$. $E|B^i|$ est donné par

$$P(|B^i| \geq k) = p_i^{k-1},$$

et donc

$$E(|B^i|) = \frac{1}{1-p_i}. \quad (3.9)$$

Calculer $E|B^i|_c$ est un peu plus difficile. On peut cependant faire la remarque suivante : l'équation 3.8 appliquée à $(\overline{\mathcal{G}}_i, \overline{\mathbf{p}}_i)$ donne

$$\begin{aligned} \gamma(\overline{\mathcal{G}}_i, \overline{\mathbf{p}}_i) &= \frac{\frac{1}{p_i} + E|B^i|_c}{\frac{1}{p_i} + \frac{1}{1-p_i}}, \\ &= (1-p_i) + p_i(1-p_i)E|B^i|_c. \end{aligned}$$

On a donc

$$E|B^i|_c = \frac{\gamma(\overline{\mathcal{G}}_i, \overline{\mathbf{p}}_i) - 1 + p_i}{p_i - p_i^2}. \quad (3.10)$$

En réinjectant les équations 3.9 et 3.10 dans 3.8, on obtient

$$\gamma(K[\mathcal{G}_1, \dots, \mathcal{G}_m], \mathbf{p}) = 1 - m + \sum_{i=1}^m \gamma(\overline{\mathcal{G}}_i, \overline{\mathbf{p}}_i).$$

□

Remarque 1 : Soit \mathcal{F}_1 la famille des graphes multipartis complets¹ et \mathcal{F}_2 la famille des graphes étoiles. La connaissance de $\gamma(\mathcal{G}, \mathbf{p})$ pour tout graphe \mathcal{G} dans \mathcal{F}_1 amène trivialement la connaissance de $\gamma(\mathcal{G}, \mathbf{p})$ pour tout graphe \mathcal{G} dans \mathcal{F}_2 puisque $\mathcal{F}_2 \subset \mathcal{F}_1$. La réciproque est aussi vraie : la connaissance de $\gamma(\mathcal{G}, \mathbf{p})$ pour tout graphe \mathcal{G} dans \mathcal{F}_2 amène la connaissance de $\gamma(\mathcal{G}, \mathbf{p})$ pour tout graphe \mathcal{G} dans \mathcal{F}_1 puisque, par le théorème de composition précédent, le calcul de γ sur un graphe multiparti complet se ramène au calcul de γ sur des étoiles.

3.4.2 Application à une famille de systèmes particuliers

Nous présentons dans cette partie une application du Théorème 7. Celui-ci va en effet nous permettre un calcul exact du codébit de certains systèmes de requêtes. Nous présenterons ensuite quelques cas particuliers de tels systèmes.

Théorème 8 Soit $\mathcal{G}[n, m]$ le graphe suivant :

$$\begin{aligned} V(\mathcal{G}[n, m]) &= \{v_{i,j}, 1 \leq i \leq n, 1 \leq j \leq 2\} \cup \{v_i, n+1 \leq i \leq n+m\}, \\ E(C(\mathcal{G}[n, m])) &= \cup_{1 \leq i \leq n} \{v_{i,1}, v_{i,2}\}, \end{aligned}$$

où $C(\mathcal{G})$ désigne le complémentaire de \mathcal{G} . Soit $p_{i,j}$ la probabilité de tirer le sommet $v_{i,j}$ et p_i la probabilité de tirer v_i . On suppose les sommets de $\mathcal{G}[n, m]$ ordonnés, et on considère \mathbf{p} le vecteur des poids. Le codébit de $(\mathcal{G}[n, m], \mathbf{p})$ est égal à :

$$\gamma(\mathcal{G}[n, m], \mathbf{p}) = 1 - n + \frac{1}{2} \sum_{i=1}^n \frac{p_{i,1} + p_{i,2} - 4p_{i,1}p_{i,2} + (2 - p_{i,1} - p_{i,2})\sqrt{1 - 4p_{i,1}p_{i,2}}}{\sqrt{1 - 4p_{i,1}p_{i,2}}}.$$

Preuve : Il suffit de remarquer que

$$\mathcal{G}[n, m] = K[\mathcal{G}_1, \dots, \mathcal{G}_{n+m}],$$

où pour $1 \leq i \leq n$ \mathcal{G}_i est le graphe union de deux sommets disjoints valués $p_{i,1}$ et $p_{i,2}$ et où pour $n+1 \leq i \leq n+m$, \mathcal{G}_i est le graphe réduit à un sommet valué p_i . Si on note K_2

1. Un graphe est dit multiparti complet si il est possible de partitionner l'ensemble de ses sommets de telle sorte que deux sommets dans une même classe ne soient jamais voisins tandis que deux sommets dans deux classes différentes sont toujours voisins.

le graphe complet à deux sommets, le Théorème 7 donne

$$\begin{aligned}\gamma(\mathcal{G}[n, \mathbf{p}], \mathbf{p}) &= 1 - n - m + \sum_{i=1}^n \gamma(E_2, (p_{i,1}, p_{i,2}, 1 - p_{i,1} - p_{i,2})) \\ &\quad + \sum_{i=n+1}^{n+m} \gamma(K_2, (p_i, 1 - p_i)), \\ \gamma(\mathcal{G}[n, \mathbf{p}], \mathbf{p}) &= 1 - n + \sum_{i=1}^n \gamma(E_2, (p_{i,1}, p_{i,2}, 1 - p_{i,1} - p_{i,2})).\end{aligned}$$

En utilisant l'équation 3.6, on obtient le résultat cherché. \square

On va maintenant présenter un cas particulier de ce théorème :

Proposition 14 Soit $\mathcal{G}[n] = \mathcal{G}[n, 0]$. On note $\gamma(\mathcal{G}[n])$ pour $\gamma(\mathcal{G}[n], (\frac{1}{2n}, \dots, \frac{1}{2n}))$. Cette notation sera justifiée par la suite.

$$\gamma(\mathcal{G}[n]) = \frac{1}{2} \left(1 + \sqrt{\frac{n-1}{n+1}} \right).$$

Preuve : Le calcul se fait simplement à partir du Théorème 8 en remplaçant les $p_{i,j}$ par leur valeur, c'est-à-dire $\frac{1}{2n}$. \square

Remarque 1 : Ce résultat était déjà connu pour les graphes $\mathcal{G}[1]$, $\mathcal{G}[2]$ et $\mathcal{G}[3]$ [JM]. Ces graphes sont représentés sur la Figure 3.4. Cependant, le calcul direct sur $\mathcal{G}[2]$ ou $\mathcal{G}[3]$ était relativement complexe, l'idée étant d'étudier la chaîne de Markov $\mathcal{M}(\text{Res}, \mathbf{S})$, d'agrèger finement celle-ci en utilisant quelques symétries non triviales du système et de se ramener à des calculs analytiques relativement longs.

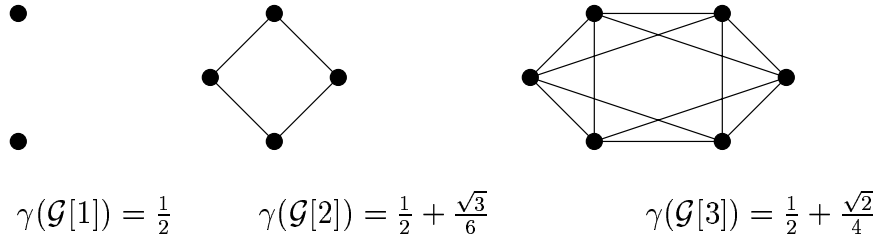


FIG. 3.4 – Les graphes $\mathcal{G}[1]$, $\mathcal{G}[2]$ et $\mathcal{G}[3]$. Les arêtes bouclant sur chaque sommet ne sont pas représentées.

Remarque 2 : Il est possible de tirer du Théorème 8 un calcul de $\gamma(\mathcal{G}[n], \mathbf{p}_n)$ où \mathbf{p}_n serait un vecteur quelconque.

3.5 Propriétés analytiques. Réduction à des systèmes uniformes

Cette partie reprend les résultats présentés dans [Brilman et al.95a] qui vont nous permettre de restreindre notre étude à une classe particulière de systèmes de ressources partagées : les systèmes uniformes. On peut noter qu'aucune hypothèse de connexité n'est utilisée dans cette partie. Les résultats énoncés restent donc vrais pour des systèmes non connexes.

3.5.1 Propriétés analytiques

Nous présentons dans cette partie quelques propriétés simples de γ qui nous seront utiles par la suite. Cependant, la première chose que nous allons faire est d'étendre la définition de $\gamma(\mathcal{G}, \mathbf{p})$ à des vecteurs \mathbf{p} quelconques, pourvu que leurs composantes soient toutes positives. En effet, de par la définition de \mathbf{p} , on a fait l'hypothèse implicite que $\|\mathbf{p}\| = 1$. Soit \mathcal{G} un graphe de taille N . On note v_1, \dots, v_N les sommets de \mathcal{G} . Soit $\mathbf{p} = (p_1, \dots, p_N) \in \mathbb{R}_+^N$ tel que $\|\mathbf{p}\| < 1$. Soit maintenant \mathcal{G}' le graphe obtenu en ajoutant à \mathcal{G} un sommet v_{N+1} correspondant à une requête vide. v_{N+1} est donc de degré 0. Soit $\mathbf{p}' = (\mathbf{p}, 1 - \|\mathbf{p}\|)$. Il est facile de vérifier, en couplant les réalisations de systèmes basés sur $(\mathcal{G}, \mathbf{p})$ et sur $(\mathcal{G}', \mathbf{p}')$, que

$$\gamma(\mathcal{G}', \mathbf{p}') = \left(\sum_{i=1}^N p_i \right) \gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|}).$$

Comme pour un couple $(\mathcal{G}, \mathbf{p})$ tel que $\|\mathbf{p}\| < 1$ il est naturel d'affecter le défaut de probabilité à une requête vide, on définit donc

$$\gamma(\mathcal{G}, \mathbf{p}) = \|\mathbf{p}\| \gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|}).$$

Bien qu'il n'y ait pas d'interprétation physique satisfaisante pour $\|\mathbf{p}\| > 1$, on utilisera la même extension. En résumé, on étend la définition de $\gamma(\mathcal{G}, \mathbf{p})$ pour $\mathbf{p} \in \mathbb{R}_+^N$ de la façon suivante :

Définition 3.5.1 $\gamma(\mathcal{G}, \mathbf{p}) = \|\mathbf{p}\| \gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|})$.

Remarque 1 : On a donc $\gamma(\mathcal{G}, \mathbf{p}) \leq \|\mathbf{p}\|$ puisque pour $\|\mathbf{p}\| = 1$, $\gamma(\mathcal{G}, \mathbf{p}) \leq 1$.

Nous allons maintenant énoncer quelques propriétés de γ vu comme fonction de \mathcal{G} et de \mathbf{p} .

Proposition 15 *On définit l'ordre partiel suivant : $\mathcal{G} \preceq \mathcal{G}'$ si et seulement si il existe un homomorphisme bijectif de \mathcal{G} dans \mathcal{G}' . Autrement dit, $\mathcal{G} \preceq \mathcal{G}'$ si et seulement si \mathcal{G}' peut être construit en ajoutant des arêtes à \mathcal{G} . Par rapport à cet ordre, $\gamma(\cdot, \mathbf{p})$ est croissante.*

Preuve : Ce résultat est une conséquence directe de la monotonie des équations d'évolution 2.3. \square

Proposition 16 *Soit \mathcal{G} un graphe tel que $V(v_1) \subset V(v_2)$. Alors, pour $x > 0$, on a l'inégalité suivante :*

$$\gamma(\mathcal{G}, (p_1 + x, p_2, p_3, \dots, p_N)) \leq \gamma(\mathcal{G}, (p_1, p_2 + x, p_3, \dots, p_N)).$$

Preuve : On suppose $x + \sum_{i=1}^n p_i = 1$. Cette hypothèse n'est pas restrictive étant donnée la linéarité en \mathbf{p} de $\gamma(\mathcal{G}, \cdot)$. Soit \mathbf{S} un système basé sur $(\mathcal{G}, \mathbf{p})$ et \mathbf{S}' un système basé sur $(\mathcal{G}, \mathbf{p}')$. On va montrer comment coupler les réalisations de \mathbf{S} et \mathbf{S}' de telle sorte que pour toute réalisation et pour tout n ,

$$\max(C_1(\mathbf{S}, n), C_2(\mathbf{S}, n)) \leq \max(C_1(\mathbf{S}', n), C_2(\mathbf{S}', n)), \quad (3.11)$$

$$\forall i \geq 3 \quad C_i(\mathbf{S}, n) \leq C_i(\mathbf{S}', n). \quad (3.12)$$

Soit $(U_n)_{n \geq 1}$ une suite de variables aléatoires indépendantes uniformes sur $[0, 1[$. D'une réalisation de U_n nous allons déduire les valeurs de $s(\mathbf{S}, n)$ et de $s(\mathbf{S}', n)$.

- Si $U_n < \sum_{i=1}^N p_i$, soit k tel que $\sum_{i=1}^{k-1} p_i \leq U_n < \sum_{i=1}^k p_i$. On pose alors $s(\mathbf{S}, n) = s(\mathbf{S}', n) = v_k$.
- Si $U_n \geq \sum_{i=1}^N p_i$ alors on pose $s(\mathbf{S}, n) = v_1$ et $s(\mathbf{S}', n) = v_2$.

La monotonie des équations d'évolution permet maintenant de vérifier par récurrence que pour toutes réalisations couplées des systèmes \mathbf{S} et \mathbf{S}' , les équations 3.11 et 3.12 sont vérifiées. \square

Remarque 2 : On peut donner une interprétation intuitive de ce résultat sur les systèmes de ressources: si dans la suite des pièces d'un Tétris on remplace certaines pièces par des pièces plus petites (*i.e.* incluses dans les précédentes), alors la hauteur finale de l'empilement sera plus petite (*i.e.* γ sera plus petit).

Corollaire 3.5.1 *La fonction $\gamma(\mathcal{G}, \cdot)$ est croissante par rapport à l'ordre composante par composante \prec sur les vecteurs de \mathbb{R}_+^N .*

Preuve : Si l'on note $v_1 \cdots v_N$ les sommets de \mathcal{G} , soit $\mathbf{p} = (p_1, \cdots, p_N)$ et $\mathbf{p}' = (p'_1, \cdots, p'_N)$ tels que $\mathbf{p} \prec \mathbf{p}'$. Soit v_{N+1} un sommet correspondant à une requête vide. On pose $\mathcal{G}' = \mathcal{G} \cup v_{N+1}$. La Proposition 16 implique

$$\begin{aligned} \gamma(\mathcal{G}', (p_1, \cdots, p_N, \sum_{i=1}^N p'_i - p_i)) &\leq \gamma(\mathcal{G}', (p'_1, \cdots, p'_N, 0)), \\ \gamma(\mathcal{G}, \mathbf{p}) &\leq \gamma(\mathcal{G}, \mathbf{p}'). \end{aligned}$$

□

Ce résultat peut s'interpréter de la façon suivante: si dans la suite des requêtes arrivant à un système, on remplace certaines requêtes vides par des requêtes non vides, on fait baisser le débit.

Proposition 17 *Soit \mathcal{G} un graphe sur les sommets v_1, \cdots, v_N tel que $V(v_1) = V(\mathcal{G})$. Soit α un réel strictement positif. Soit $\mathbf{p} = (p_1, \cdots, p_N)$. Soit \mathcal{S} un système basé sur $(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|})$. Soit X_β une variable aléatoire de distribution géométrique de paramètre $\beta = \frac{\alpha}{\|\mathbf{p}\| + \alpha}$: $P(X_\beta = k) = \beta(1 - \beta)^k$. On a*

$$\gamma(\mathcal{G}, (p_1 + \alpha, p_2, \cdots, p_N)) = \alpha(1 + \mathbb{E}h(\mathcal{S}, X_\beta)).$$

Preuve : Supposons dans un premier temps que $\alpha + \|\mathbf{p}\| = 1$. On considère le graphe \mathcal{G}' construit en ajoutant à \mathcal{G} un sommet v_{N+1} que l'on connecte à tous les sommets. v_1 et v_{N+1} ayant les mêmes voisinages dans \mathcal{G}' , la Proposition 16 implique que

$$\gamma(\mathcal{G}, (p_1 + \alpha, p_2, \cdots, p_N)) = \gamma(\mathcal{G}', (p_1, \cdots, p_N, \alpha)).$$

Chapitre 3. Systèmes indépendants

Soit \mathbf{S}_α un système basé sur $(\mathcal{G}', (p_1, \dots, p_N, \alpha))$. Soit $t_0 = 0$ et t_k le rang de la k -ième apparition du sommet v_{N+1} dans le système \mathbf{S}_α . On peut écrire

$$\begin{aligned} \gamma(\mathbf{S}_\alpha) &= \lim_{n \rightarrow \infty} \frac{h(\mathbf{S}_\alpha, t_n)}{t_n}, \\ &= \lim_{n \rightarrow \infty} \frac{\sum_{k=1}^n (h(\mathbf{S}_\alpha, t_k) - h(\mathbf{S}_\alpha, t_{k-1}))}{\sum_{k=1}^n (t_k - t_{k-1})}, \\ &= \frac{\mathbb{E}h(\mathbf{S}_\alpha, t_1)}{Et_1}. \end{aligned}$$

Sur l'intervalle $[0, t_1[$, le système \mathbf{S}_α étant conditionné par le fait qu'aucune requête v_{N+1} n'apparaît, il évolue comme un système basé sur $(\mathcal{G}', (p_1/(1-\alpha), \dots, p_N/(1-\alpha), 0))$, c'est-à-dire comme un système basé sur $(\mathcal{G}, (p_1/\|\mathbf{p}\|, \dots, p_N/\|\mathbf{p}\|))$, c'est-à-dire comme \mathbf{S} . Le sommet v_{N+1} étant connecté à tous les sommets de \mathcal{G}' , on peut écrire

$$\gamma(\mathbf{S}_\alpha) = \frac{1 + \mathbb{E}h(\mathbf{S}, t_1 - 1)}{1/\alpha}.$$

Soit X_α une variable aléatoire de distribution géométrique de paramètre α . $(t_1 - 1)$ a la même distribution que X_α . On peut donc écrire

$$\gamma(\mathcal{G}, (p_1 + \alpha, p_2, \dots, p_N)) = \alpha(1 + \mathbb{E}h(\mathbf{S}, X_\alpha)). \quad (3.13)$$

On vient donc de démontrer la proposition dans le cas où $\alpha + \|\mathbf{p}\| = 1$. Supposons maintenant que $\alpha + \|\mathbf{p}\| \neq 1$.

$$\gamma(\mathcal{G}, (p_1 + \alpha, p_2, \dots, p_N)) = (\alpha + \|\mathbf{p}\|) \gamma(\mathcal{G}, \frac{1}{\alpha + \|\mathbf{p}\|} (p_1 + \alpha, p_2, \dots, p_N)).$$

D'après 3.13, on a donc

$$\begin{aligned} \gamma(\mathcal{G}, (p_1 + \alpha, p_2, \dots, p_N)) &= (\alpha + \|\mathbf{p}\|) \frac{\alpha}{\|\mathbf{p}\| + \alpha} (1 + \mathbb{E}h(\mathbf{S}', X_{\frac{\alpha}{\|\mathbf{p}\| + \alpha}}^0)), \\ &= \alpha(1 + \mathbb{E}h(\mathbf{S}', X_\beta)), \end{aligned}$$

où \mathbf{S}' est un système basé sur $(\mathcal{G}, (\frac{p_1/(\|\mathbf{p}\| + \alpha)}{\|\mathbf{p}\|/(\|\mathbf{p}\| + \alpha)}, \dots, \frac{p_N/(\|\mathbf{p}\| + \alpha)}{\|\mathbf{p}\|/(\|\mathbf{p}\| + \alpha)}))$, c'est-à-dire $(\mathcal{G}, (\frac{p_1}{\|\mathbf{p}\|}, \dots, \frac{p_N}{\|\mathbf{p}\|}))$. \mathbf{S}' est donc égal à \mathbf{S} . \square

Théorème 9 (Continuité) *La fonction $\gamma(\mathcal{G}, \cdot)$ est continue.*

Preuve : On peut supposer qu'un des sommets de \mathcal{G} est connecté à l'ensemble des sommets de \mathcal{G} . En effet, si jamais cette hypothèse n'est pas vérifiée, il suffit de rajouter à \mathcal{G} un tel sommet auquel on affecte le poids 0. Sans perte de généralité, nous supposons de plus que ce sommet est v_1 . Soit donc $\mathbf{p} = (p_1, \dots, p_N)$. Nous allons montrer que la fonction $\gamma(\mathcal{G}, \cdot)$ est continue en ce point. Soit

$$\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N),$$

tel que $p_i + \varepsilon_i \geq 0$ pour tout i . Soit

$$\begin{aligned} \boldsymbol{\varepsilon}_+ &= \boldsymbol{\varepsilon} \oplus (0, \dots, 0) \text{ la partie positive de } \boldsymbol{\varepsilon}, \\ \boldsymbol{\varepsilon}_- &= \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_+ \text{ la partie négative de } \boldsymbol{\varepsilon}. \end{aligned}$$

On note $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq N} |x_i|$.

$$\begin{aligned} \gamma(\mathcal{G}, \mathbf{p} + \boldsymbol{\varepsilon}) &\geq \gamma(\mathcal{G}, \mathbf{p} + \boldsymbol{\varepsilon}_-), \\ &\geq \gamma(\mathcal{G}, \mathbf{p} - \|\boldsymbol{\varepsilon}_-\|_\infty \mathbf{1}), \\ &\geq \min_{p_i \neq 0} \frac{p_i - \|\boldsymbol{\varepsilon}_-\|_\infty}{p_i} \gamma(\mathcal{G}, \mathbf{p}). \end{aligned} \tag{3.14}$$

Par ailleurs,

$$\begin{aligned} \gamma(\mathcal{G}, \mathbf{p} + \boldsymbol{\varepsilon}) &\leq \gamma(\mathcal{G}, \mathbf{p} + \boldsymbol{\varepsilon}_+), \\ &\leq \gamma(\mathcal{G}, (p_1 + N\|\boldsymbol{\varepsilon}_+\|_\infty, p_2, \dots, p_N)). \end{aligned}$$

Rappelons maintenant la proposition 17 :

$$\gamma(\mathcal{G}, (p_1 + \alpha, p_2, \dots, p_N)) = \alpha(1 + \mathbb{E}h(\mathbf{S}, X_\beta)),$$

où \mathbf{S} est un système basé sur $(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|})$ et X_β une variable aléatoire de distribution géométrique de paramètre $\beta = \frac{\alpha}{\|\mathbf{p}\| + \alpha}$.

Soit e_1 l'événement $\{X_\beta \leq \alpha^{-\frac{1}{3}}\}$ et e_2 l'événement $\{X_\beta > \alpha^{-\frac{1}{3}}\}$. Il est clair que

$$P(e_1) \leq \lceil \beta^{-\frac{1}{3}} \rceil \beta = \mathcal{O}(\alpha^{\frac{2}{3}})$$

au voisinage de 0. Ceci implique que $P(e_2) = 1 + o(1)$. Comme $\mathbb{E}h(\mathbf{S}, n) \leq n$, il est clair que

$$\mathbb{E}h(\mathbf{S}, X_\beta | e_1) = \mathcal{O}(\alpha^{-\frac{1}{3}}).$$

Par ailleurs, comme $\lim_{n \rightarrow \infty} \mathbb{E}h(\mathbf{S}, n)/n = \gamma(\mathcal{G}, \mathbf{p}/\|\mathbf{p}\|)$, on a

$$\begin{aligned} \mathbb{E}h(\mathbf{S}, X_\beta | e_2) &= (\gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|}) + o(1))E(X_\beta | e_2), \\ &= (\gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|}) + o(1))\frac{\|\mathbf{p}\|}{\alpha}, \\ &= \frac{\gamma(\mathcal{G}, \mathbf{p}) + o(1)}{\alpha}. \end{aligned}$$

De tout ceci on déduit

$$\begin{aligned} \frac{1}{\alpha}\gamma(\mathcal{G}, (p_1 + \alpha, \dots, p_N)) &= 1 + P(e_1)\mathbb{E}(h(\mathbf{S}, X_\beta) | e_1) + P(e_2)\mathbb{E}(h(\mathbf{S}, X_\beta) | e_2), \\ &= 1 + \mathcal{O}(\alpha^{\frac{2}{3}})\mathcal{O}(\alpha^{-\frac{1}{3}}) + (1 + o(1))\frac{\gamma(\mathcal{G}, \mathbf{p}) + o(1)}{\alpha}, \\ &= 1 + o(1) + \frac{\gamma(\mathcal{G}, \mathbf{p}) + o(1)}{\alpha} \end{aligned}$$

au voisinage de 0. En remplaçant α par $N|\boldsymbol{\varepsilon}_+|_\infty$ dans l'équation précédente, on obtient

$$\gamma(\mathcal{G}, \mathbf{p} + \boldsymbol{\varepsilon}) \leq \gamma(\mathcal{G}, \mathbf{p}) + o(1). \quad (3.15)$$

Les inégalités 3.14 et 3.15 impliquent

$$\lim_{|\boldsymbol{\varepsilon}|_\infty \rightarrow 0} \gamma(\mathcal{G}, \mathbf{p} + \boldsymbol{\varepsilon}) = \gamma(\mathcal{G}, \mathbf{p}),$$

et donc $\gamma(\mathcal{G}, \cdot)$ est continue. □

Remarque 3 : Ce théorème justifie l'utilisation de techniques de simulation pour estimer $\gamma(\mathcal{G}, \mathbf{p})$ pour un \mathbf{p} donné. En effet, dans la plupart des cas, un programme de simulation censé estimer $\gamma(\mathcal{G}, \mathbf{p})$ va "tourner" sur \mathbf{p}' proche de \mathbf{p} . La continuité de la fonction $\gamma(\mathcal{G}, \cdot)$ implique que la simulation va tendre vers une valeur proche de $\gamma(\mathcal{G}, \mathbf{p})$.

3.5.2 Restriction à des mesures uniformes

Nous allons montrer dans cette partie qu'il est possible d'encadrer le codébit de tout système par les codébits de deux systèmes dans lesquels chaque type de requête a la même probabilité d'apparition. En d'autres termes, si \mathcal{G} est un graphe de taille N et $\mathbf{p} = (p_1, \dots, p_N)$ un vecteur de \mathbb{R}_+^N , nous allons montrer comment construire deux graphes, \mathcal{G}_ε et \mathcal{G}^ε de tailles respectivement \underline{n} et \bar{n} tels que

$$\gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_{\underline{n}}) \leq \gamma(\mathcal{G}, \mathbf{p}) \leq \gamma(\mathcal{G}^\varepsilon, \varepsilon \mathbf{1}_{\bar{n}}), \quad (3.16)$$

où $\mathbf{1}_{\underline{n}}$ est le vecteur de $\mathbb{R}_+^{\underline{n}}$ dont toutes les coordonnées sont égales à 1. Les graphes \mathcal{G}_ε et \mathcal{G}^ε sont donc fonction de \mathcal{G} et d'un réel arbitraire ε . Nous montrerons par la suite que la continuité de la fonction $\gamma(\mathcal{G}, \cdot)$ implique que pour ε suffisamment petit, ces deux bornes peuvent être aussi fines que souhaité. Définissons tout d'abord pour $1 \leq i \leq N$ et $\varepsilon > 0$

$$\underline{c}_i = \lfloor \frac{p_i}{\varepsilon} \rfloor \text{ et } \bar{c}_i = \lceil \frac{p_i}{\varepsilon} \rceil.$$

Les \underline{c}_i et \bar{c}_i sont fonctions de ε . Cependant, par soucis de simplicité, nous avons décidé de ne pas le montrer dans leurs notations. \mathcal{G}_ε est défini de la façon suivante :

$$\mathcal{G}_\varepsilon = (V_\varepsilon, E_\varepsilon),$$

où

$$\begin{aligned} V_\varepsilon &= \{(i, j) \text{ t.q. } 1 \leq i \leq N, 1 \leq j \leq \underline{c}_i\}, \\ E_\varepsilon &= \cup_{\{v_i, v_{i'}\} \in E(\mathcal{G})} \{(i, j), (i', j')\}. \end{aligned}$$

Pour construire \mathcal{G}_ε , on réplique \underline{c}_i fois le sommet v_i de \mathcal{G} . Grossièrement, la complexité d'une telle transformation en nombre de sommets et d'arêtes créés est en $\Theta(\varepsilon^{-2})$. On définit $\underline{n} = \text{card}(V_\varepsilon)$. Il est facile de vérifier que

$$\gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_{\underline{n}}) = \gamma(\mathcal{G}, (\underline{c}_1 \varepsilon, \dots, \underline{c}_N \varepsilon)),$$

ce qui implique d'une part (croissance de $\gamma(\mathcal{G}, \cdot)$) que

$$\gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_{\underline{n}}) \leq \gamma(\mathcal{G}, \mathbf{p}),$$

et d'autre part que

$$\lim_{\|\underline{\varepsilon}\| \rightarrow 0} \gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_n) = \gamma(\mathcal{G}, \mathbf{p}),$$

en utilisant la continuité de $\gamma(\mathcal{G}, \cdot)$ et le fait que

$$\lim_{\|\underline{\varepsilon}\| \rightarrow 0} (\underline{c}_1 \varepsilon, \dots, \underline{c}_N \varepsilon) = (p_1, p_2, \dots, p_N).$$

Le graphe \mathcal{G}^ε est construit sur le même principe en reproduisant cette fois-ci \bar{c}_i fois le sommet v_i de \mathcal{G} . Si \bar{n} est la taille de \mathcal{G}^ε , on aura

$$\gamma(\mathcal{G}^\varepsilon, \varepsilon \mathbf{1}_{\bar{n}}) \geq \gamma(\mathcal{G}, (p_1, \dots, p_N)).$$

Les figures 3.5 et 3.6 montrent les graphes \mathcal{G}_ε et \mathcal{G}^ε associés à deux systèmes donnés. On peut remarquer que la Figure 3.6 représente un cas particulier pour lequel les p_i/ε sont tous entiers. Dans ce cas, $\mathcal{G}_\varepsilon = \mathcal{G}^\varepsilon$.

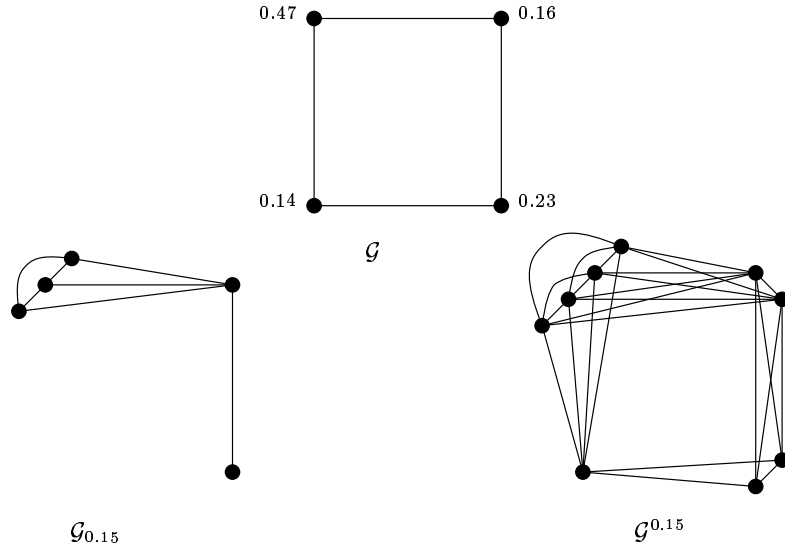


FIG. 3.5 – Un couple $(\mathcal{G}, \mathbf{p})$ et les graphes \mathcal{G}_ε et \mathcal{G}^ε associés pour $\varepsilon = 0.15$.

Remarque 1 : La somme des poids des sommets de \mathcal{G}_ε ou \mathcal{G}^ε n'est pas nécessairement égale à 1. Cependant, l'équation 3.16 peut être réécrite de la façon suivante :

$$\underline{n} \varepsilon \gamma(\mathcal{G}_\varepsilon, \frac{1}{\underline{n}} \mathbf{1}_{\underline{n}}) \leq \gamma(\mathcal{G}, \mathbf{p}) \leq \bar{n} \varepsilon \gamma(\mathcal{G}^\varepsilon, \frac{1}{\bar{n}} \mathbf{1}_{\bar{n}}).$$

3.5 Propriétés analytiques. Réduction à des systèmes uniformes

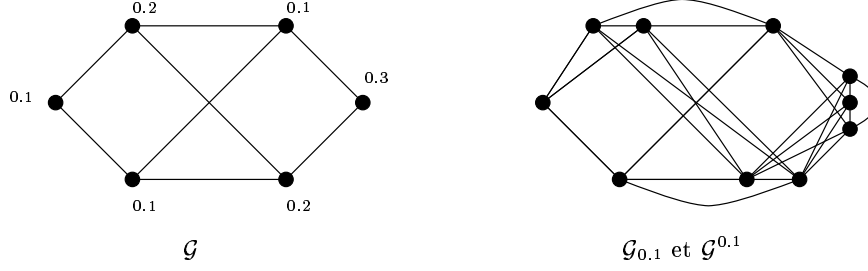


FIG. 3.6 – Un couple $(\mathcal{G}, \mathbf{p})$ et les graphes \mathcal{G}_ε et \mathcal{G}^ε associés pour $\varepsilon = 0.1$.

Ceci implique que, d'un point de vue numérique, on peut se restreindre à des mesures uniformes sur les sommets des graphes qu'on considère. La figure 3.7 présente un exemple de système de requêtes, la valeur de γ pour ce système, ainsi que les valeurs de $\underline{n}\varepsilon\gamma(\mathcal{G}_\varepsilon, \frac{1}{\underline{n}}\mathbf{1}_{\underline{n}})$ et $\bar{n}\varepsilon\gamma(\mathcal{G}^\varepsilon, \frac{1}{\bar{n}}\mathbf{1}_{\bar{n}})$ pour différentes valeurs de ε .

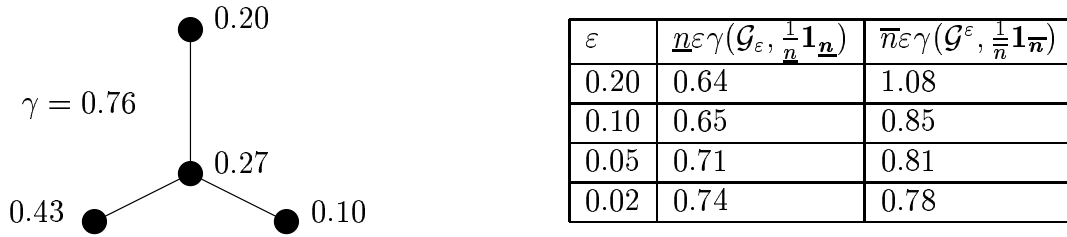


FIG. 3.7 – Approximations d'un système non uniforme par des systèmes uniformes.

Nous posons maintenant les définitions suivantes :

Définition 3.5.2 Un système associé à un graphe \mathcal{G} et à un vecteur de probabilités uniforme sur \mathcal{G} sera dit uniforme.

Définition 3.5.3

$$\gamma(\mathcal{G}) = \gamma(\mathcal{G}, \frac{1}{N}\mathbf{1}_N), \quad (3.17)$$

où N est la taille de \mathcal{G} .

Remarque 2 : Les systèmes uniformes ont deux intérêts principaux : d'une part, certaines des bornes que nous présenterons par la suite ne sont valables que pour des systèmes uniformes. Cependant, si l'on se donne un système non uniforme particulier, on pourra les utiliser en commençant par encadrer notre système par deux systèmes uniformes.

D'autre part, certaines des bornes pour les systèmes uniformes peuvent être étendues analytiquement à des systèmes non uniformes. Cependant, il est souvent plus pratique de commencer à prouver la borne sur un système uniforme puis de l'étendre à un système quelconque, plutôt que de chercher à la prouver directement sur un système quelconque.

Remarque 3 : Soit ε tel que

$$\varepsilon \leq \min_{i, p_i \neq 0} p_i.$$

Le corollaire 3.5.1 implique que

$$\begin{aligned} \gamma(\mathcal{G}, (\overline{c_1}\varepsilon, \dots, \overline{c_N}\varepsilon)) &\leq \left(\max_{i, p_i \neq 0} \frac{\overline{c_i}}{c_i} \right) \gamma(\mathcal{G}, (c_1\varepsilon, \dots, c_N\varepsilon)), \\ \gamma(\mathcal{G}^\varepsilon, \varepsilon \mathbf{1}_{\overline{\mathbf{n}}}) &\leq \left(\max_{i, p_i \neq 0} \frac{\overline{c_i}}{c_i} \right) \gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_{\underline{\mathbf{n}}}). \end{aligned}$$

Et donc, en utilisant le fait que $\gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_{\underline{\mathbf{n}}}) \leq \varepsilon \underline{\mathbf{n}} \leq 1$,

$$|\gamma(\mathcal{G}^\varepsilon, \varepsilon \mathbf{1}_{\overline{\mathbf{n}}}) - \gamma(\mathcal{G}_\varepsilon, \varepsilon \mathbf{1}_{\underline{\mathbf{n}}})| \leq \left(\max_{i, p_i \neq 0} \frac{\overline{c_i}}{c_i} \right) - 1 \leq \max_{i, p_i \neq 0} \frac{1}{c_i}.$$

Ce résultat implique que, étant donné $\alpha > 0$, il est possible de choisir ε de telle sorte que la différence entre les deux bornes de l'équation 3.16 soit moindre que α . En effet, il suffit de prendre

$$\varepsilon \leq \alpha \min_{i, p_i \neq 0} p_i \text{ tel que } \min_{i, p_i \neq 0} \frac{p_i}{\varepsilon} \text{ est entier.}$$

3.6 Systèmes uniformes

Cette partie présente des bornes spécifiques aux systèmes uniformes. Cependant nous verrons que les bornes présentées dans les parties 3.6.1 et 3.6.2 peuvent être étendues aux systèmes non uniformes.

3.6.1 Une borne basée sur le rayon spectral de \mathcal{G}

Théorème 10 *Soit \mathcal{G} un graphe de taille N . Soit $A(\mathcal{G})$ la matrice d'adjacence de \mathcal{G} et ρ son rayon spectral. On a la borne suivante sur $\gamma(\mathcal{G})$:*

$$\gamma(\mathcal{G}) \leq f^{-1}\left(\frac{\rho}{N}\right),$$

avec

$$f(x) = \exp\left(\frac{x \ln(x) + (1-x) \ln(1-x)}{x}\right)$$

la fonction dont le graphe est représenté sur la figure 3.8.

On peut remarquer que le théorème de Perron-Frobenius [Gantmacher60] implique que ρ est, dans ce cas particulier, la plus grande valeur propre réelle de $A(\mathcal{G})$.

Preuve : Rappelons tout d'abord un lemme utile [Bacelli et al.91] :

Lemme 3.6.1 (Bacelli, Konstantopoulos) *Soit \mathcal{S} un système indépendant quelconque. Soit α un réel tel que*

$$\lim_{n \rightarrow \infty} P(h(n) \geq n\alpha) = 0. \tag{3.18}$$

On a alors $\gamma \leq \alpha$.

Ce lemme est en fait une conséquence directe de la convergence vers γ de la suite $\{h(n)/n\}_{n \geq 1}$. Notre but est donc maintenant de trouver un réel α qui vérifie 3.18. Pour cela, nous allons établir un résultat de la forme

$$P(h(n) \geq n\alpha) \leq ce^{\theta(\alpha)n}.$$

où $\theta(\alpha)$ est une fonction décroissante de α . Nous en déduirons donc comme borne supérieure pour γ la valeur de α en laquelle θ s'annule. Pour cela, évaluons tout d'abord la probabilité que $h(n)$ soit supérieur ou égal à m , avec n et $m \leq n$ fixés. Soit $c(n, m) = \binom{n}{m}$.

Chapitre 3. Systèmes indépendants

On note $s_1, s_2 \dots s_{c(n,m)}$ les différentes sous-suites de taille m extraites de la suite des n premières requêtes du système. Les deux évènements suivants sont équivalents :

$$\{h(n) \geq m\} \Leftrightarrow \{\exists i \in 1 \dots c(n, m) \text{ tel que } s_i \text{ est un chemin de } \mathcal{G}\}.$$

De cette équivalence, de la sous-additivité d'une mesure de probabilité et de l'équidistribution des s_i , on déduit :

$$P(h(n) \geq m) \leq c(n, m)P(s_1 \text{ est un chemin de } \mathcal{G}).$$

Évaluons maintenant la quantité $P(s_1 \text{ est un chemin dans } \mathcal{G})$. Pour ce faire, on peut remarquer que $A(\mathcal{G})_{i,j}^{(m)}$ compte le nombre de chemins de taille m dans \mathcal{G} qui partent du sommet v_i et arrivent au sommet v_j . On a donc

$$P(s_1 \text{ est un chemin de } \mathcal{G}) = \frac{\sum_{i,j} A(\mathcal{G})_{i,j}^{(m)}}{N^m} = O\left(\left(\frac{\rho}{N}\right)^m\right).$$

En remplaçant m par $n\alpha$, avec $0 < \alpha < 1$, on obtient :

$$P(h(n) \geq n\alpha) = O\left(\frac{n!}{(n - n\alpha)!(n\alpha)!} \left(\frac{\rho}{N}\right)^{n\alpha}\right).$$

En utilisant la formule de Stirling, et après simplification, on obtient :

$$P(h(n) \geq n\alpha) = O\left(\frac{1}{(\alpha\beta)^{\alpha n} (1 - \alpha)^{n(1-\alpha)} \sqrt{2\pi n\alpha(1 - \alpha)}}\right),$$

avec $\beta = N/\rho$. Pour tout α tel que $(\alpha\beta)^\alpha (1 - \alpha)^{1-\alpha} > 1$ on a donc

$$\lim_{n \rightarrow \infty} P(h(n) \geq n\alpha) = 0.$$

et, d'après le lemme 3.6.1 , $\gamma \leq \alpha$. On en déduit

$$\gamma \leq \inf\{\alpha \text{ t.q. } \alpha \ln(\alpha\beta) + (1 - \alpha) \ln(1 - \alpha) > 0\},$$

ce qui peut se récrire

$$\gamma \leq \inf\{\alpha \text{ t.q. } f(\alpha) > \frac{\rho}{N}\},$$

avec

$$f(x) = \exp\left(\frac{x \ln(x) + (1-x) \ln(1-x)}{x}\right).$$

La fonction f étant croissante et continue, on a

$$\gamma \leq f^{-1}\left(\frac{\rho}{N}\right).$$

□

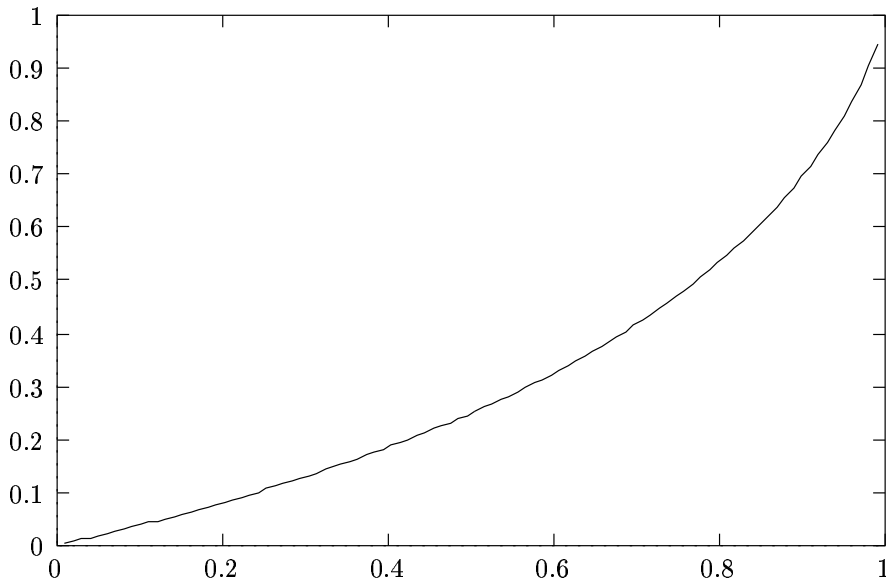


FIG. 3.8 – La fonction f .

Remarque 1 : Étant donné un graphe \mathcal{G} , il est généralement difficile d'obtenir une expression exacte de $\rho(\mathcal{G})$. Cependant ρ peut être estimé, soit en calculant le polynôme caractéristique de $A(\mathcal{G})$, dans le cas de petits graphes, soit en utilisant une méthode des puissances [Golub et al.87]. Ainsi, le coût du calcul de cette borne supérieure est

relativement faible et dépend de la précision souhaitée pour ρ .

Corollaire 3.6.1 *Pour un graphe \mathcal{G} quelconque,*

$$\gamma(\mathcal{G}) \leq \frac{ed_{\max}(\mathcal{G})}{N},$$

où $e = \exp(1)$ et $d_{\max}(\mathcal{G})$ représente le plus grand degré d'un sommet dans \mathcal{G} .

Preuve : la fonction f^{-1} est croissante et $\rho(\mathcal{G}) \leq d_{\max}(\mathcal{G})$. On a donc

$$\gamma(\mathcal{G}) \leq f^{-1} \left(\frac{d_{\max}(\mathcal{G})}{N} \right).$$

Comme f^{-1} est concave et $(f^{-1})'(0) = \exp(1)$,

$$\gamma(\mathcal{G}) \leq \frac{ed_{\max}(\mathcal{G})}{N}.$$

□

Corollaire 3.6.2 *Pour $(\mathcal{G}, \mathbf{p})$ quelconque,*

$$\gamma(\mathcal{G}, \mathbf{p}) \leq e \max_i \left(\sum_{j, v_j \in V(v_i)} p_j \right).$$

Preuve : Supposons tout d'abord que $\|\mathbf{p}\| = 1$. Le degré maximum des graphes \mathcal{G}_ε et \mathcal{G}^ε , définis dans la partie 3.5.2, est équivalent à $\max_i \sum_{j, v_j \in V(v_i)} p_j / \varepsilon$. Leurs tailles sont équivalentes à $1/\varepsilon$. On en déduit, pour $\|\mathbf{p}\| = 1$,

$$\gamma(\mathcal{G}, \mathbf{p}) \leq e \max_i \sum_{j, v_j \in V(v_i)} p_j.$$

Supposons maintenant $\|\mathbf{p}\| \neq 1$. En utilisant le fait que

$$\gamma(\mathcal{G}, \mathbf{p}) = \|\mathbf{p}\| \gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|}),$$

on obtient

$$\gamma(\mathcal{G}, \mathbf{p}) \leq e \max_i \sum_{j, v_j \in V(v_i)} p_i.$$

□

3.6.2 Une borne inférieure fonction du degré moyen de \mathcal{G}

Théorème 11 *Soit \mathcal{G} un graphe quelconque de taille N . Soit $d_{\text{moy}}(\mathcal{G})$ le degré moyen de \mathcal{G} . On a la borne inférieure suivante pour $\gamma(\mathcal{G})$:*

$$\frac{d_{\text{moy}}(\mathcal{G})}{N} \leq \gamma(\mathcal{G}).$$

Preuve : Soit \mathbf{S} le système canoniquement associé à \mathcal{G} . Nous allons définir un système \mathbf{S}' , basé sur \mathcal{G} mais non indépendant, qui minore trajectoriellement \mathbf{S} . Nous calculerons ensuite le codébit de ce système \mathbf{S}' .

Soit $(U_n)_{n \geq 1}$ la suite des arrivées de requêtes dans le système \mathbf{S} . On définit deux suites, $(U'_n)_{n \geq 1}$ et $(M_n)_{n \geq 1}$, de la façon suivante :

$$U'_1 = U_1 \text{ et } M_1 = U_1.$$

Si

- $U_{n+1} \in V(M_n)$ alors $U'_{n+1} = U_{n+1}$ et $M_{n+1} = U_{n+1}$.
- $U_{n+1} \notin V(M_n)$ alors $U'_{n+1} = \emptyset$ et $M_{n+1} = M_n$.

Soit \mathbf{S}' le système basé sur \mathcal{G} dont les arrivées de requêtes sont données par la suite $(U'_n)_{n \geq 1}$. Le système \mathbf{S}' n'est pas indépendant. Nous montrerons cependant que son codébit est défini. Par ailleurs, \mathbf{S}' étant majoré trajectoriellement par \mathbf{S} , $\gamma(\mathbf{S}') \leq \gamma(\mathbf{S})$. Cherchons à calculer $\gamma(\mathbf{S}')$. Pour cela nous allons utiliser la proposition 10. En effet,

$$\begin{aligned} I(\mathbf{S}', n) &= P(h(\mathbf{S}', n+1) - h(\mathbf{S}', n) = 1), \\ &= \sum_{v \in V(\mathcal{G})} P(M_n = v) P(U'_{n+1} \in V(v)). \end{aligned}$$

Chapitre 3. Systèmes indépendants

On peut remarquer que la suite $(M_n)_{n \geq 1}$ correspond à la réalisation d'un processus de Markov basé sur une chaîne dont les sommets seraient les sommets de \mathcal{G} avec une transition de v vers v' si et seulement si $v' \in V(v)$, cette transition se faisant alors avec probabilité $1/N$. Une telle chaîne de Markov admet une mesure stationnaire uniforme. On en déduit que pour tout $v \in V(\mathcal{G})$, $\lim_{n \rightarrow \infty} P(M_n = v) = \frac{1}{N}$. Par ailleurs, $P(U'_{n+1} \in V(v)) = \frac{d(v)}{N}$. On en déduit

$$\begin{aligned} \lim_{n \rightarrow \infty} I(\mathbf{S}', n) &= \sum_{v \in V(\mathcal{G})} \frac{d(v)}{N^2}, \\ &= \frac{d_{\text{moy}}(\mathcal{G})}{N}. \end{aligned}$$

Et donc, d'après la Proposition 10,

$$\gamma(\mathbf{S}') = \frac{d_{\text{moy}}(\mathcal{G})}{N}.$$

□

Corollaire 3.6.3 Soit \mathcal{G} un graphe de taille N et \mathbf{p} un vecteur de \mathbb{R}_+^N . On a la borne inférieure suivante pour $\gamma(\mathcal{G}, \mathbf{p})$:

$$\frac{\sum_i p_i \sum_{j, v_j \in V(v_i)} p_j}{\|\mathbf{p}\|} \leq \gamma(\mathcal{G}, \mathbf{p}).$$

Preuve : Supposons dans un premier temps que $\|\mathbf{p}\| = 1$. Soit \mathcal{G}_ε et \mathcal{G}^ε les graphes définis dans la partie 3.5.2. On peut remarquer que $d_{\text{moy}}(\mathcal{G}_\varepsilon)$ et $d_{\text{moy}}(\mathcal{G}^\varepsilon)$ sont équivalents à

$$\frac{\sum_i p_i \sum_{j, v_j \in V(v_i)} p_j}{\varepsilon}$$

quand ε tend vers 0 tandis que les tailles de ces deux graphes sont équivalentes à $1/\varepsilon$. Sous l'hypothèse $\|\mathbf{p}\| = 1$, on a donc

$$\sum_i p_i \sum_{j, v_j \in V(v_i)} p_j \leq \gamma(\mathcal{G}, \mathbf{p}).$$

En utilisant maintenant le fait que pour tout \mathbf{p} ,

$$\gamma(\mathcal{G}, \mathbf{p}) = \|\mathbf{p}\| \gamma(\mathcal{G}, \frac{\mathbf{p}}{\|\mathbf{p}\|}),$$

on peut écrire

$$\|\mathbf{p}\| \left(\sum_i \frac{p_i}{\|\mathbf{p}\|} \sum_{j, v_j \in V(v_i)} \frac{p_j}{\|\mathbf{p}\|} \right) \leq \gamma(\mathcal{G}, \mathbf{p})$$

ce qui donne le résultat souhaité. □

Corollaire 3.6.4 *Soit \mathbf{S} un système indépendant quelconque vérifiant la condition de réalisme 2.2. Afin d'alléger les notations, on définit $L_1 = L(\mathbf{S}, 1)$. L_1 représente donc l'ensemble des requêtes traitées en même temps que la première requête de \mathbf{S} , c'est-à-dire l'ensemble des requêtes de \mathbf{S} traitées pendant la première unité de temps. On a la borne inférieure suivante pour $\gamma(\mathbf{S})$:*

$$\frac{1}{\mathbb{E}|L_1|} \leq \gamma(\mathbf{S}).$$

Preuve : Soit $(\mathcal{G}, \mathbf{p})$ le couple graphe-vecteur décrivant \mathbf{S} . Pour des raisons d'interprétation physique, on suppose $\|\mathbf{p}\| = 1$. $|L_1|$ peut s'exprimer de la façon suivante :

$$|L_1| = \sum_{i=1}^N \mathbb{1}_{v_i \in L_1}.$$

En passant aux espérances :

$$\mathbb{E}|L_1| = \sum_{i=1}^N P(v_i \in L_1).$$

Il est facile de vérifier qu'une condition nécessaire et suffisante pour que la requête v_i soit dans L_1 est que la première occurrence de celle-ci précède les premières occurrences de ses voisines dans \mathcal{G} . On a donc

$$P(v_i \in L_1) = \frac{p_i}{\sum_{j, v_j \in V(v_i)} p_j},$$

ce qui donne

$$\mathbb{E}|L_1| = \sum_{i=1}^N \frac{p_i}{\sum_{j, v_j \in V(v_i)} p_j}.$$

$1/\mathbb{E}|L_1|$ s'exprime donc comme une moyenne harmonique pondérée par les coefficients p_i des $\sum_{j, v_j \in V(v_i)} p_j$. Cette quantité est donc majorée par une moyenne "classique" des mêmes termes pondérés par les mêmes coefficients. Ce qui amène :

$$\frac{1}{\mathbb{E}|L_1|} \leq \sum_i p_i \sum_{j, v_j \in V(v_i)} p_j,$$

d'où l'on tire

$$\frac{1}{\mathbb{E}|L_1|} \leq \gamma(\mathbf{S}).$$

□

Remarque 1 : Étant donnés \mathcal{G} et \mathbf{p} , ce résultat offre peu d'intérêt pour l'estimation de $\gamma(\mathcal{G}, \mathbf{p})$. Cependant, il peut s'interpréter de la façon suivante : étant donné un système saturé \mathbf{S} , son codébit asymptotique est minoré par l'inverse de son débit moyen sur la première unité de temps. Autrement dit, pour un système physique *pouvant être modélisé par un système de requêtes* mais tel qu'on ignore quels peuvent être \mathcal{G} et \mathbf{p} , on peut avoir une borne inférieure pour le codébit asymptotique en se contentant d'observer le comportement moyen du système sur la première unité de temps. Il semble difficile de généraliser ce résultat à une ligne n quelconque.

3.6.3 Relation entre $\gamma(\mathcal{G})$ et $\chi(\mathcal{G})$

Théorème 12 Soit \mathcal{G} un graphe réflexif de taille N . On note $\chi(\mathcal{G})$ le nombre chromatique de \mathcal{G} [Berge83]. On a alors la borne inférieure suivante pour $\gamma(\mathcal{G})$:

$$\frac{\chi(\mathcal{G})}{N} \leq \gamma(\mathcal{G}).$$

Preuve : Rappelons tout d'abord la notion de graphe χ -critique [Berge83].

Définition 3.6.1 *Un graphe \mathcal{G} est dit χ -critique si la suppression d'un quelconque de ses sommets diminue strictement son nombre chromatique.*

Un des principaux résultats sur les graphes χ -critiques est le suivant :

Proposition 18 *Soit \mathcal{G} un graphe réflexif χ -critique de nombre chromatique p . Tous les sommets de \mathcal{G} sont de degré supérieur ou égal à p .*

On pourra trouver la démonstration de cette proposition dans [Berge83] (prop. 2, page 329), en faisant toutefois attention au fait que Berge considère des graphes non réflexifs, ce qui induit une différence de 1 sur les degrés des sommets. Passons maintenant à la preuve du théorème proprement dite. Celle-ci est relativement simple.

Soit \mathcal{G}_1 inclus dans \mathcal{G} , de même nombre chromatique que \mathcal{G} et qui soit χ -critique. Soit N_1 la taille de \mathcal{G}_1 . On suppose les sommets de \mathcal{G} numérotés de telle sorte que

$$\begin{aligned} V(\mathcal{G}) &= \{v_1, \dots, v_N\}, \\ V(\mathcal{G}_1) &= \{v_1, \dots, v_{N_1}\}. \end{aligned}$$

Soit \mathbf{p} un vecteur de \mathbb{R}^N dont les N_1 premières composantes sont égales à $1/N$ et les $(N - N_1)$ dernières composantes égales à 0. Soit $\mathbf{p}_1 = (N/N_1)\mathbf{p}$. D'après la proposition 3.5.1,

$$\begin{aligned} \gamma(\mathcal{G}) &\geq \gamma(\mathcal{G}, \mathbf{p}) \\ &\geq \frac{N_1}{N} \gamma(\mathcal{G}, \mathbf{p}_1) \\ &\geq \frac{N_1}{N} \gamma(\mathcal{G}_1) \end{aligned} \tag{3.19}$$

\mathcal{G}_1 étant de degré minimum supérieur ou égal à $\chi(\mathcal{G})$, il est de degré moyen supérieur ou égal à $\chi(\mathcal{G})$. D'après le Théorème 11, on a donc

$$\frac{\chi(\mathcal{G})}{N_1} \leq \gamma(\mathcal{G}_1). \tag{3.20}$$

Les inégalités 3.19 et 3.20 amènent

$$\frac{\chi(\mathcal{G})}{N} \leq \gamma(\mathcal{G}).$$

□

Ce résultat peut être mis en relation avec un résultat dû à Barbosa et Gafni [Barbosa et al.89]. Nous exposons brièvement celui-ci, sans reprendre toutefois les notations des auteurs afin d'en rendre l'exposé plus clair. Dans leur article, Barbosa et Gafni considèrent une classe de politiques d'ordonnancement pour les systèmes de ressources partagées. Chacune de ces politiques d'ordonnancement a pour caractéristique, notamment, de prévenir tout phénomène de famine. La structure logique de ces systèmes est la même que celle que nous avons adoptée : un graphe d'exclusion \mathcal{G} . La dynamique de ceux-ci est par contre assez différente de celle que nous avons considérée : une file d'attente est mise en place au niveau de chaque sommet du graphe (En d'autres termes, on dispose d'une file d'attente par type de requête possible). A chaque instant, seuls certains types de requêtes (correspondant à un stable de \mathcal{G}) sont autorisés à être exécutés. Les stables successifs de \mathcal{G} autorisés à l'exécution sont précisés par la politique d'ordonnancement. Pour une politique P d'ordonnancement fixée, les auteurs définissent $\gamma(P)$ comme le codébit que cette politique va induire dans le système. Ils s'intéressent ensuite à une politique P^* offrant le meilleur (*i. e.*, le plus petit) codébit et montrent que $\gamma(P^*) \leq \chi(\mathcal{G})/N$. Ils montrent ensuite que la détermination de cette politique optimale est un problème NP -difficile. Le théorème 12 implique que le codébit obtenu par P^* est meilleur que $\gamma(\mathcal{G})$. Attention, cependant à l'interprétation de cet énoncé : nous ne nous contentons pas de dire que "l'ordonnancement optimal est meilleur que l'ordonnancement aléatoire". En effet, les dynamiques des systèmes dans cette thèse et dans [Barbosa et al.89] ne sont pas les mêmes : "ordonnancement optimal" signifie "optimal dans la classe d'ordonnancement considérés par Barbosa et Gafni". Or, la politique d'ordonnancement que nous considérons ne fait pas partie de cette classe.

Remarque 1 : Pour un graphe \mathcal{G} quelconque, non nécessairement réflexif, la proposition 18 se réécrit

Proposition 19 *Les sommets d'un graphe \mathcal{G} , χ -critique, sont tous de degré au moins $\chi(\mathcal{G}) - 1$.*

Ainsi, on peut prouver, en calquant notre démonstration sur celle du théorème 12,

Proposition 20 *Pour tout graphe \mathcal{G} , de taille N et non nécessairement réflexif,*

$$\frac{\chi(\mathcal{G}) - 1}{N} \leq \gamma(\mathcal{G}).$$

3.6.4 Quelques exemples

Nous présentons sur la figure 3.9 quelques graphes pour lesquels nous donnons la valeur de γ (simulée, voir partie 3.1.3) ainsi que les valeurs des bornes présentées dans les théorèmes 10, 11 et 12. On note b_1 la borne du théorème 10, b_2 celle du théorème 11 et b_3 celle du théorème 12. On remarquera sur ces exemples que les bornes b_2 et b_3 ne se comparent pas.

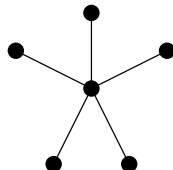
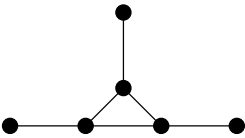
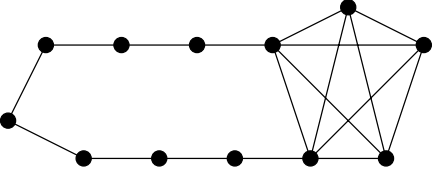
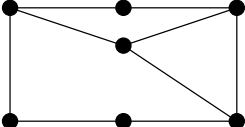
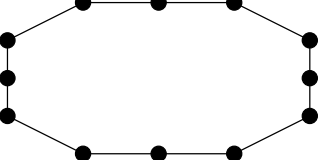

	b_2	b_3	γ	b_1
	0.44	0.33	0.51	0.77
	0.50	0.50	0.59	0.82
	0.33	0.42	0.44	0.71
	0.51	0.43	0.59	0.79
	0.25	0.17	0.31	0.50
	0.44	0.33	0.52	0.75

FIG. 3.9 – Quelques graphes et les valeurs de b_1 , b_2 et b_3 pour ces graphes. Tous les graphes présentés ici sont implicitement réflexifs.

Chapitre 4

Études asymptotiques

Ce chapitre est consacré à l'étude des codébits de familles de graphes particulières. Nous commençons par rappeler dans la partie 4.1 les résultats obtenus dans le chapitre 3 et en déduisons en corollaire le comportement asymptotique de γ pour une suite de graphes de degré constant. La partie 4.2 est centrée sur la comparaison terme à terme de deux familles de graphes. Nous en profitons pour étudier sur cet exemple la qualité de deux des bornes que nous avons présentées. La partie 4.3 présente l'étude d'un cas asymptotique pour lequel le résultat présenté dans la partie 4.1 apporte peu d'information.

4.1 Rappel des bornes. Cas de graphes de degré constant

Nous présentons dans cette partie un rapide résumé des bornes obtenues jusqu'à présent. Nous ferons par la suite une remarque dans le cas des graphes de degré constant.

- Pour un graphe \mathcal{G} quelconque,

$$\frac{\chi(\mathcal{G}) - 1}{N} \leq \gamma(\mathcal{G})$$

$$\frac{d_{\text{moy}}(\mathcal{G})}{N} \leq \gamma(\mathcal{G}) \leq f^{-1}\left(\frac{\rho(\mathcal{G})}{N}\right) \leq e \frac{d_{\text{max}}(\mathcal{G})}{N}.$$

- Pour un graphe \mathcal{G} réflexif,

$$\frac{\chi(\mathcal{G})}{N} \leq \gamma(\mathcal{G}).$$

– Pour \mathcal{G} et \mathbf{p} quelconques,

$$\frac{\sum_i p_i \sum_{j, v_j \in V(v_i)} p_j}{\|\mathbf{p}\|} \leq \gamma(\mathcal{G}, \mathbf{p}) \leq e \max_i \sum_{j, v_j \in V(v_i)} p_j.$$

– Pour \mathcal{G} réflexif et \mathbf{p} tel que $\|\mathbf{p}\| = 1$

$$\frac{1}{|L_1|} \leq \gamma(\mathcal{G}, \mathbf{p}) \leq u_m(\mathcal{G}, \mathbf{p}).$$

Corollaire 4.1.1 *Soit $\mathcal{G}(n)$ une suite de graphes, $\mathcal{G}(n)$ étant de degré constant $d(n)$ et de taille $t(n)$,*

$$\gamma(\mathcal{G}(n)) = \Theta\left(\frac{d(n)}{t(n)}\right).$$

Preuve : Immédiate, puisque

$$\frac{d(n)}{t(n)} \leq \gamma(\mathcal{G}(n)) \leq \frac{ed(n)}{t(n)}.$$

□

On peut donc remarquer que deux graphes de degré constant, de mêmes tailles et de mêmes degrés, ne peuvent avoir des codébits différant dans un rapport supérieur à e .

4.2 Comparaison de deux familles de systèmes de ressources

Soit R_n et C_n des systèmes de ressources uniformes définis comme suit :

$$\begin{aligned} \text{Res}(R_n) = \text{Res}(C_n) &= \{r_i, 1 \leq i \leq n\}, \\ \text{Req}(R_n) &= \cup_{j=(i+1)[n]} \{r_i, r_j\}, \\ \text{Req}(C_n) &= \cup_{j \neq i} \{r_i, r_j\}. \end{aligned}$$

4.2 Comparaison de deux familles de systèmes de ressources

Comparer les codébits de R_n et C_n pour n fixé est un problème lié à la comparaison d'algorithmes de relaxation synchrone [Eick et al.91] sur deux types de réseaux différents : l'anneau (*Ring*) ou le graphe complet (*Complete*). Attention cependant : il ne faut pas confondre les graphes permettant de modéliser ces réseaux (qui ont donné leurs noms aux systèmes) avec les graphes d'exclusion que nous avons considérés jusqu'alors. Ainsi, le graphe d'exclusion associé au système C_n n'est pas une clique.

Il est possible de constater, en estimant à l'aide d'une simulation $\gamma(R_n)$ et $\gamma(C_n)$ que le résultat suivant semble juste pour tout n :

$$\gamma(R_n) \leq \gamma(C_n). \quad (4.1)$$

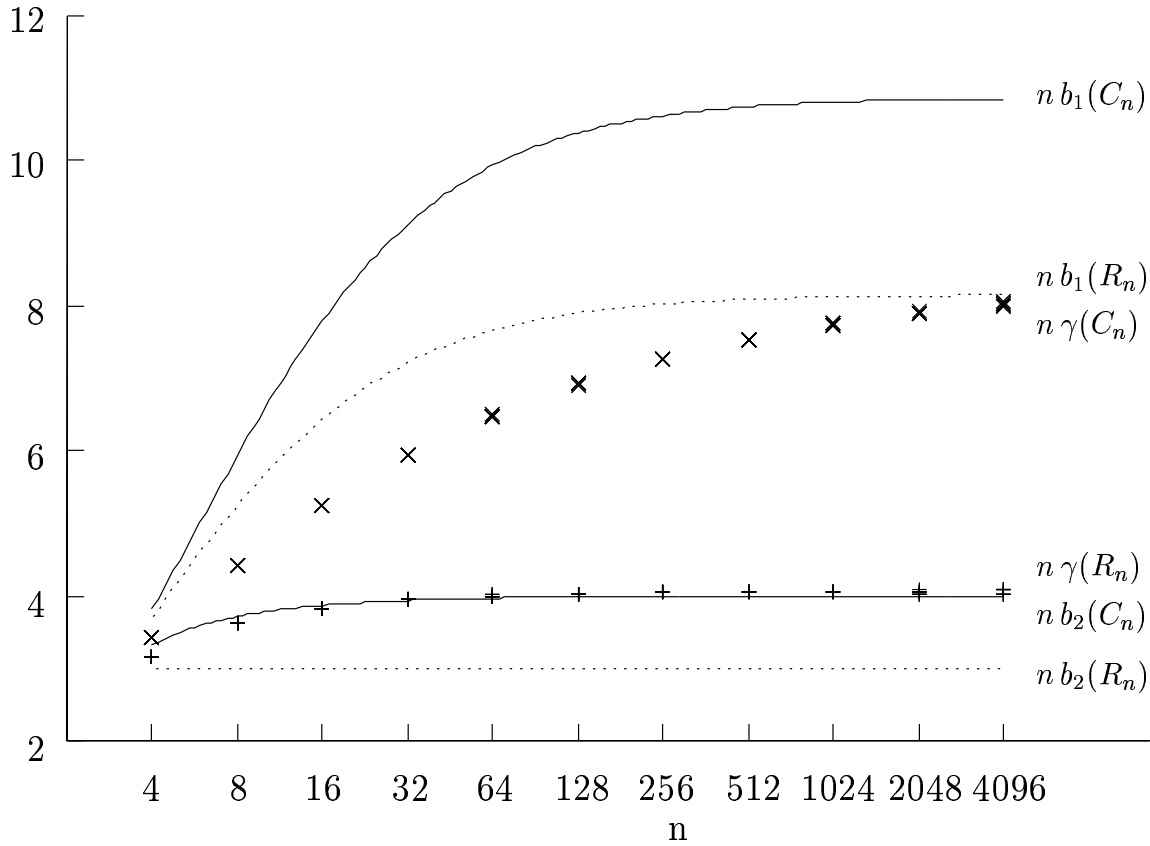
Nous présentons sur la figure 4.1 les valeurs estimées de $n\gamma(R_n)$ et de $n\gamma(C_n)$ et de leurs bornes déduites des théorèmes 10 et 11 pour diverses valeurs de n . On note b_1 la borne déduite du Théorème 10 et b_2 la borne déduite du Théorème 11. Il est à noter que nous n'avons pu donner d'intervalles de confiance pour les valeurs simulées. En effet, les systèmes étudiés ici sont beaucoup trop grands pour que l'approche présentée dans la partie 3.1.3 puisse être conduite en temps raisonnable. Pour chaque système et chaque valeur de n nous avons donc fait 10 simulations (5 qui renvoient la valeur simulée de $h(n)/n$ et 5 qui renvoient la valeur de $\underline{h}(n)/n$) et reporté les résultats de ces 10 simulations sur la figure. Bien que l'argument ne soit pas rigoureux, la proximité des points résultant des diverses simulations semble garantir que les valeurs obtenues sont proches des valeurs réelles.

Au cours d'une collaboration avec Jean Mairesse et Bruno Gaujal, nous avons tenté, à l'aide de certaines approches présentées dans cette thèse, de prouver l'inégalité 4.1. Cependant, bien que l'étude d'un tel problème ait été intéressante d'un point de vue théorique (elle nous a permis d'acquérir une certaine compréhension des systèmes de ressources et par là même des systèmes de requêtes), nous n'avons pas abouti. J'ai par la suite tenté d'autres approches, infructueuses elles aussi. Le meilleur résultat auquel je sois arrivé est une application triviale des bornes b_1 et b_2 qui donne

$$\gamma(R_n) \leq \frac{3e}{n} \text{ et } \gamma(C_n) \geq \frac{4n - 6}{n(n - 1)}$$

d'où l'on peut conclure

$$\gamma(R_n) \leq \frac{3e}{4}(\gamma(C_n) + o(1)),$$


 FIG. 4.1 – Codébits des systèmes de ressources R_n et C_n pour de grandes valeurs de n .

ce qui est insuffisant. Vincent Dumas, à l'aide de processus de branchements [Dumas], obtient la borne supérieure suivante pour $\gamma(R_n)$:

$$\gamma(R_n) \leq \frac{5.512}{n}.$$

En utilisant cette nouvelle borne supérieure, on obtient cette fois ci :

$$\gamma(R_n) \leq 1.378(\gamma(C_n) + o(1)),$$

ce qui est plus fin, mais toujours insuffisant.

On peut enfin remarquer que l'inégalité 4.1 est triviale pour $n = 2$ ou $n = 3$ (puisque dans ce cas les codébits valent tous 1) et que dans le cas $n = 4$, on a $R_4 = \mathcal{G}[2]$ et $C_4 = \mathcal{G}[3]$ (cf partie 3.4.2). Connaissant donc explicitement $\gamma(R_4)$ et $\gamma(C_4)$ on peut conclure que l'inégalité est vraie pour $n = 4$.

4.3 Graphes étoiles

Nous présentons dans cette partie une étude des graphes étoiles (on notera E_n l'étoile à n branches). Cette étude a été motivée par deux raisons principales : d'une part, il est facile de vérifier que l'application du théorème 7 à l'étoile à n branches en la considérant comme la composition de deux graphes, l'un égal à son centre seul et l'autre à la réunion des extrémités de ses branches, donne

$$\gamma(E_n) = 1 - 2 + 1 + \gamma(E_n).$$

Ceci fait de l'étoile une sorte de cas "pathologique", par là même intéressant à étudier. d'autre part, comme on l'a vu dans la partie 4.1, pour une suite de graphes $\mathcal{G}(n)$, chaque $\mathcal{G}(n)$ étant de degré constant $d(n)$ et de taille $t(n)$, on a

$$\gamma(\mathcal{G}(n)) = \Theta\left(\frac{d(n)}{t(n)}\right),$$

c'est-à-dire que l'on connaît le comportement asymptotique de $\gamma(\mathcal{G}(n))$. Les étoiles présentent la particularité qu'elles sont des graphes de degré non constant tout en étant simples. On peut vérifier que $d_{moy}(E_n) = \frac{3n+1}{n+1}$ et que $\rho(E_n) = 1 + \sqrt{n-1}$, et donc, que l'application des théorèmes 11 et 10 donne

$$\gamma(E_n) = \Omega(1/n) \text{ et } \gamma(E_n) = O(1/\sqrt{n}).$$

On voit que c'est insuffisant pour décrire le comportement asymptotique de $\gamma(E_n)$.

Nous allons donc, dans un premier temps, citer un résultat dû à Alain Jean-Marie qui exprime $\gamma(E_n)$ sous forme intégrale. Nous donnerons ensuite une analyse du comportement asymptotique de $\gamma(E_n)$ quand n tend vers l'infini. Ce travail a été fait en collaboration avec Alain Jean-Marie et Bruno Gaujal de l'INRIA-Sophia.

4.3.1 Expression intégrale de $\gamma(E_n)$

Théorème 13 (Jean-Marie) *Soit v_0 le centre de l'étoile à n branches et $v_1 \cdots v_n$ les extrémités de ses branches. Soit $\mathbf{p} = (p_0, \dots, p_n)$ tel que $\|\mathbf{p}\| = 1$. On peut exprimer $\gamma(E_n, \mathbf{p})$ comme :*

$$\gamma(E_n, \mathbf{p}) = p_0 + \sum_{i=1}^n p_i \phi\left(\sigma^{n-i}(p_1, \dots, p_n)\right),$$

où σ est la permutation circulaire à gauche, et

$$\phi(p_1, \dots, p_n) = \left(\frac{1}{2i\pi} \right)^{n-1} \int_{\Gamma_1} \dots \int_{\Gamma_{n-1}} \Pi(x_1, \dots, x_{n-1}) \frac{dz_1}{z_1 - 1} \dots \frac{dz_{n-1}}{z_{n-1} - 1},$$

avec

$$\Pi(x_1, \dots, x_{n-1}) = \frac{p_0}{1 - \sum_{i=1}^{n-1} p_i/x_i - p_n \prod_{i=1}^{n-1} x_i}.$$

Dans cette formule, les contours $\Gamma_1, \dots, \Gamma_{n-1}$ sont simples, dans le domaine $\{|z| > 1\}$ et dans le domaine d'analyticité de la fonction Π .

Remarque 1 : Il est assez simple de donner une expression de $\gamma(E_n)$ sous forme de sommes multiples (faisant alors intervenir $n + 1$ signes de sommation!). Cependant, bien qu'un tel résultat puisse sembler plus parlant que le théorème 13, il est plus mauvais d'un point de vue numérique et n'amène aucun résultat concret.

4.3.2 Comportement asymptotique de $\gamma(E_n)$

Proposition 21 $\gamma(E_n) = \Theta\left(\frac{\ln n}{n \ln \ln n}\right).$

Preuve : Rappelons tout d'abord un résultat classique portant sur des jets de boules dans des urnes [Kolchin et al.78].

Lemme 4.3.1 *Soit n urnes dans lesquelles on jette m boules au hasard uniformément. Les jets de deux boules sont supposés indépendants. Soit $M(n, m)$ le nombre moyen de boules que contient alors l'urne la plus remplie. Alors, si $m = \Theta(n)$,*

$$M(n, m) \simeq_{n \rightarrow \infty} \frac{\ln n}{\ln \ln n}.$$

Montrons maintenant comment rattacher ce problème à l'étude du comportement asymptotique de $\gamma(E_n)$. Nous pouvons tout d'abord remarquer que E_n peut être représenté par un système de ressources $\mathbf{S}(n)$ construit sur n ressources et comportant $n + 1$ requêtes,

n de ces requêtes demandant chacune une ressource différente et la $(n + 1)$ -ième requête demandant, quant à elle, l'ensemble des n ressources. D'après la Proposition 17,

$$\gamma(E_n) = \frac{1}{n+1} (1 + \mathbb{E}h(\mathbf{S}'(n), X_n)),$$

où X_n est une variable aléatoire ayant une distribution géométrique de paramètre $1/(n+1)$ et où $\mathbf{S}'(n)$ est un système de ressources uniforme basé sur n ressources et possédant n requêtes, chaque requête demandant une ressource différente. Il est facile de vérifier que $\mathbb{E}h(\mathbf{S}'(n), X_n) = M(n, X_n)$. Or,

$$M(n, X_n) \geq P(X_n \geq n)M(n, n).$$

Quand n tend vers l'infini, $P(X_n \geq n) = (n/n+1)^n$ tend vers $1/e$. Ainsi, $M(n, X_n) = \Omega(\ln n / \ln \ln n)$. Par ailleurs, si on pose $X'_n = (n+1)\lceil X_n/n+1 \rceil$, la croissance en m de $M(n, m)$ implique que

$$\begin{aligned} M(n, X_n) &\leq M(n, X'_n), \\ &\leq \sum_{k=1}^{\infty} M(n, k(n+1))P(X'_n = k(n+1)), \\ &\leq M(n, n+1) \left(\sum_{k=1}^{\infty} kP(X'_n = k(n+1)) \right), \\ &\leq M(n, n+1) \frac{\mathbb{E}X'_n}{n+1}, \\ &\leq M(n, n+1) \frac{n+1 + \mathbb{E}X_n}{n+1}, \\ &\leq M(n, n+1)O(1), \\ &= O\left(\frac{\ln n}{\ln \ln n}\right). \end{aligned}$$

On a donc

$$M(n, X_n) = \Theta\left(\frac{\ln n}{\ln \ln n}\right)$$

et

$$\gamma(E_n) = \Theta\left(\frac{\ln n}{n \ln \ln n}\right).$$



Chapitre 5

Conclusion

Nous avons présenté dans cette thèse un modèle de systèmes de ressources partagées. Nous nous sommes ensuite intéressés à l'évaluation, sur ce modèle, d'un paramètre de performance fondamental, le codébit. Ce paramètre de performance caractérise, en effet, de façon fine la dynamique du système auquel il est associé. Nous montrons par exemple en annexe comment son évaluation permet de donner des conditions de stabilité ou de non-stabilité de la file d'attente d'un système particulier.

Le calcul exact du codébit d'un système quelconque semble être un problème particulièrement difficile. Nous n'avons en effet réussi à le calculer que pour quelques cas de systèmes relativement simples. Pour cette raison, nous avons cherché à en donner des bornes. Sans revenir en détail sur celles-ci, on peut faire trois remarques d'ordre général. Tout d'abord, les diverses bornes que nous avons obtenues sont dérivées d'expressions analytiques du codébit différentes, bien qu'équivalentes. Il semblerait donc que chacune de ces expressions "capture" un aspect différent de la dynamique des systèmes que l'on a étudiés. D'autre part, nous avons obtenu deux bornes qui, pourvu que chacune des requêtes soit en concurrence avec un nombre fixe de requêtes, permettent une bonne estimation du codébit puisqu'elles l'encadrent dans un rapport au plus $\exp(1)$. Enfin, il semble que la vision "systèmes de requêtes" soit plus pertinente, bien que moins intuitive, que la vision "systèmes de ressources". En effet, elles amène des bornes directement reliées au graphe \mathcal{G} .

Nous montrons enfin en annexe quels sont les systèmes $(\max, +)$ que notre approche en termes de graphes permet de traiter et comment il serait possible de généraliser le modèle que l'on a étudié afin de lui faire correspondre un plus grand nombre de systèmes $(\max, +)$.

Annexe A

Applications

A.1 Stabilité d'un système de ressources partagées

Nous présentons ici, de façon succincte et sur un exemple, une application possible des bornes obtenues précédemment.

Soit un système de ressources partagées composé de $N = 2^n$ processeurs et de N mémoires. Les processeurs sont connectés aux mémoires par un réseau d'interconnection de type "Papillon". La figure A.1 montre un tel réseau d'interconnection pour $N = 8$.

Lorsqu'un processeur veut lire ou écrire dans une mémoire, il verrouille le chemin qui les relie, empêchant donc éventuellement d'autres accès à d'autres mémoires par d'autres processeurs. Pour un tel système, les ressources correspondent donc aux noeuds du réseau d'interconnection et les requêtes aux demandes d'accès à une mémoire par les processeurs. On fait l'hypothèse que les temps de prise des ressources par les requêtes sont tous unitaires, que les diverses requêtes possibles sont équiprobables et que chaque processeur demande à accéder à une mémoire avec une fréquence f . Étant donnée f , on peut se poser la question suivante : notre système est-il stable ? En d'autres termes, le nombre moyen de requêtes en attente de traitement est-il uniformément borné dans le temps ? Ce qui revient encore à dire : le temps moyen qu'une requête attend avant d'être traitée est-il uniformément borné dans le temps ? La réponse théorique à toutes ces questions est "oui" si et seulement si $f < \frac{1}{N\gamma}$ [Baccelli et al.95], où γ est le codébit de la version saturée du système que nous étudions. Le problème est que l'on ne connaît pas γ . Par contre, comme il est possible de borner inférieurement et supérieurement celui-ci, on va pouvoir donner des conditions suffisantes de stabilité ou de non-stabilité du système non saturé. En effet, on peut vérifier que le graphe \mathcal{G} associé au système saturé est de taille N^2 et de degré

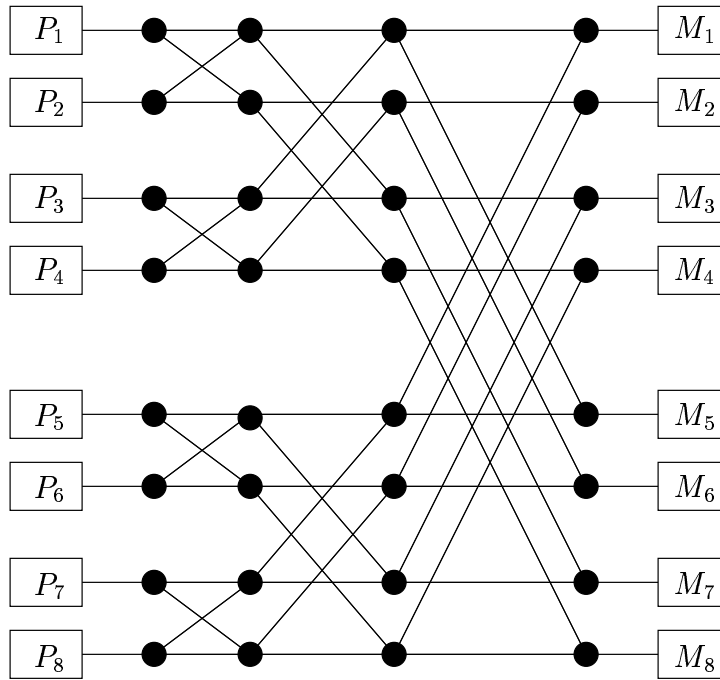


FIG. A.1 – Un réseau d'interconnexion en "Papillon".

constant $N + \frac{N}{2} \ln N$. On en déduit, d'après les bornes des théorèmes 10 et 11,

$$\frac{2 + \ln N}{2N} < \gamma < \frac{e(2 + \ln N)}{2N}.$$

Ainsi, on obtient

$$f \leq \frac{2}{e(2 + \ln N)} \Rightarrow \text{Le système est stable,}$$

$$f \geq \frac{2}{2 + \ln N} \Rightarrow \text{Le système est instable.}$$

ce que nous avons résumé sur la figure A.2.

Application numérique : Pour un réseau de taille 256, on assure donc la stabilité pour $f \leq 0.073$ tandis qu'on est sûr de l'instabilité pour $f \geq 0.2$.

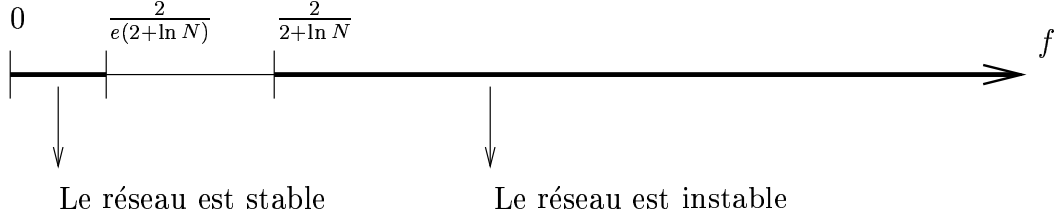


FIG. A.2 – Zones dans lesquelles on sait si le réseau d’interconnection est stable ou instable.

A.2 Application à $(\max, +)$

Nous avons montré dans la partie 2.2 comment les équations d’évolution que l’on a étudiées se récrivent dans $(\max, +)$. La question que l’on peut maintenant se poser est la suivante : supposons résolu le problème du calcul de $\gamma(\mathcal{G}, \mathbf{p})$ pour un graphe \mathcal{G} non orienté quelconque et un vecteur \mathbf{p} quelconque. Quels sont alors les types de systèmes stochastiques $(\max, +)$ dont on connaîtrait les exposants de Lyapunov ? En d’autres termes, pour quel type de matrices stochastiques M saurait-on calculer $\gamma(M)$, où $\gamma(M)$ est défini comme l’exposant de Lyapunov associé à une suite de matrices stochastiques indépendantes entre elles et distribuées identiquement à M (voir partie 3.1.2).

Afin de répondre à cette question, on définit

$$\mathcal{M}(N, l) = \{M = (m_{i,j})_{1 \leq i,j \leq N} \text{ t.q. } \left(\begin{array}{ll} m_{i,i} = 0 & \text{si } i \neq l, \\ m_{i,j} = -\infty & \text{si } i \neq l, j \neq i, \\ m_{l,j} \in \{-\infty, 1\} & \text{sinon.} \end{array} \right)\}.$$

Proposition 22 Soit une matrice stochastique M telle qu’il existe d’une part $M_1 \cdots M_N$ appartenant respectivement à $\mathcal{M}(N, 1) \cdots \mathcal{M}(N, N)$ et d’autre part $\mathbf{p} = (p_1, \dots, p_N)$, $\|\mathbf{p}\| = 1$, vérifiant

$$\begin{aligned} P(M = M_i) &= p_i \text{ pour } 1 \leq i \leq N, \\ M^\oplus &= \oplus_i M_i \text{ est symétrique.} \end{aligned}$$

Alors, $\gamma(M) = \gamma(\mathcal{G}, \mathbf{p})$ où \mathcal{G} est le graphe suivant :

$$\begin{aligned} V(\mathcal{G}) &= \{v_1, \dots, v_N\}, \\ E(\mathcal{G}) &= \{\{v_i, v_j\} \text{ t.q. } m_{i,j}^\oplus = 1\}. \end{aligned}$$

Preuve : Il suffit de vérifier que le système $(\max, +)$ associé à $(\mathcal{G}, \mathbf{p})$ est bien le système $(\max, +)$ associé à M (voir l'équation 2.5 dans la partie 2.2.2).

□

Exemple 1 : Soit

$$M_1 = \begin{pmatrix} 1 & -\infty & 1 \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{pmatrix},$$

$$M_2 = \begin{pmatrix} 0 & -\infty & -\infty \\ -\infty & 1 & 1 \\ -\infty & -\infty & 0 \end{pmatrix},$$

$$M_3 = \begin{pmatrix} 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \\ 1 & 1 & 1 \end{pmatrix},$$

et M définie par

$$P(M = M_1) = p_1,$$

$$P(M = M_2) = p_2,$$

$$P(M = M_3) = 1 - p_1 - p_2.$$

On vérifie que

$$M^\oplus = \begin{pmatrix} 1 & -\infty & 1 \\ -\infty & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

est symétrique. On définit donc le graphe \mathcal{G} par

$$V(\mathcal{G}) = \{v_1, v_2, v_3\},$$

$$E(\mathcal{G}) = \{\{v_1, v_1\}, \{v_1, v_3\}, \{v_2, v_2\}, \{v_2, v_3\}, \{v_3, v_3\}\}.$$

On a alors

$$\begin{aligned} \gamma(M) &= \gamma(\mathcal{G}, \mathbf{p}) \\ &= \frac{1}{2} \frac{p_1 + p_2 - 4p_1p_2 + (2 - p_1 - p_2)\sqrt{1 - 4p_1p_2}}{\sqrt{1 - 4p_1p_2}} \end{aligned}$$

d'après l'équation 3.6.

Remarque 1 : Il serait possible de se débarrasser de la condition “ M^\oplus est symétrique” en définissant γ pour des graphes orientés. Étendre la définition de γ à des graphes orientés est en effet tout à fait possible puisque les équations d'évolution 2.3 ne particularisent pas le fait que \mathcal{G} est non-orienté. Cependant, nous ne nous sommes pas intéressés à cette généralisation pour les deux raisons suivantes: on se serait trop écarté du problème de départ, qui est la modélisation de systèmes de ressources partagées. Par ailleurs, les bornes présentées dans cette thèse, à l'exception de la borne donnée par le théorème 10, ne se généralisent pas à des graphes orientés.

A.3 Table des notations

Notations “classiques” :

\mathbb{N}	Entiers naturels
\mathbb{R}	Nombres réels
$\mathbf{1}_n$	Vecteur de \mathbb{R}^n ayant toutes ses composantes égales à 1
$\mathcal{P}(A)$	Ensemble des parties de A
$ $	Valeur absolue (ou taille d'une ligne)
$ x _\infty$	$\max_i x_i $
$ x $	$\sum_i x_i $
M	Une matrice (ou un motif)
e	$\exp(1)$
$f = \Omega(g)$	$\forall \varepsilon > 0 \exists x_0, x \geq x_0 \Rightarrow \varepsilon f(x) > g(x) $
$f = o(g)$	$\forall \varepsilon > 0 \exists x_0, x \geq x_0 \Rightarrow f(x) < \varepsilon g(x) $
$f = O(g)$	$\exists c \exists x_0, x > x_0 \Rightarrow f(x) < c g(x) $
$f = \Theta(g)$	$f = O(g)$ et $g = O(f)$
P	Probabilité
\mathbb{I}	Indicatrice
\mathbb{E}	Espérance
Var	Variance
Cov	Covariance
E	Événement

Notations définies pour un système \mathbf{S} :

Res	Ensemble des ressources
K	Nombre de ressources
r_i	Une ressource
Req	Ensemble des requêtes
N	Nombre de requêtes
v_i	Une requête
$\mathbf{p} = (p_1, \dots, p_N)$	Vecteur de poids des requêtes
$s(n)$	n -ième requête
$T_i(n)$	Date de libération de r_i après traitement des n premières requêtes
$C_i(n)$	Date de fin de traitement de la n -ième requête
$h(n)$	$\max_i C_i(n)$
$\underline{h}(n)$	$\min_i C_i(n)$
γ	Codébit du système \mathbf{S}
$T_{\text{Req}}(\mathbf{S})$	Système de requêtes associé à un système de ressources \mathbf{S}
$T_{\text{Res}}(\mathbf{S})$	Système de ressources associé à un système de requêtes \mathbf{S}
$v \parallel v'$	v et v' ne sont pas concurrentes
$v \perp v'$	v et v' sont concurrentes
$\text{Ind}(\mathbf{S})$	Ensemble des ensembles de requêtes non concurrentes du système \mathbf{S}
M	Un motif (ou une matrice)
$L(n)$	n -ième ligne
$ $	Taille d'une ligne (ou valeur absolue)
R_n	Système de ressources dit "anneau"
C_n	Système de ressources dit "complet"

Notations définies pour un graphe \mathcal{G} :

$V(\mathcal{G})$	Ensemble des sommets de \mathcal{G}
$E(\mathcal{G})$	Ensemble des arêtes de \mathcal{G}
v	Un sommet de \mathcal{G} (<i>i.e.</i> une requête du système correspondant)
$V(v)$	Ensemble des voisins du sommet v
d_{moy}	Degré moyen
d_{max}	Degré maximum
χ	Nombre chromatique
E_n	Étoile à n branches

Bibliographie

- [Asmussen87] Asmussen (S.). – *Applied Probability and Queues*. – John Wiley & Sons, 1987.
- [Baccelli et al.89] Baccelli (F.) et Makowski (A.). – Queuing Models for Systems with Synchronization Constraints. *In: Special issue on Dynamics of Discrete Event Systems*. – Proceedings of the IEEE.
- [Baccelli et al.91] Baccelli (F.) et Konstantopoulos (P.). – Estimates of Cycle Time in Stochastic Petri Nets. *In: Proceedings of Workshop on Stochastic Analysis*, éd. par Karatzas (I.). – Rutgers University, 1991.
- [Baccelli et al.92a] Baccelli (F.), Cohen (G.), Olsder (G.) et Quadrat (J-P.). – *Synchronization and Linearity. An Algebra for Discrete Event Systems*. – John Wiley and Sons, 1992.
- [Baccelli et al.92b] Baccelli (F.) et Liu (Z.). – Comparaison Properties of Stochastic Decision Free Petri Nets. *IEEE Transactions on Automatic Control*, vol. 37 (12), December 1992, pp. 1905–1920.
- [Baccelli et al.95] Baccelli (F.) et Mairesse (J.). – *Ergodic Theory of Stochastic Operators and Discrete Event Networks*. – Technical Report RR-2641, INRIA, 1995. To appear in *Idempotency*, Publications of the Isaac Newton Institute, Cambridge University Press, in 1996.
- [Barbosa et al.89] Barbosa (V. C.) et Gafni (E.). – Concurrency in Heavily Loaded Neighborhood-Constrained Systems. *ACM Transactions on Programming Languages and Systems*, vol. 11 (4), October 1989, pp. 562–584.
- [Berge83] Berge (C.). – *Graphes*. – Paris, Gauthier-Villars, 1983.
- [Berge87] Berge (C.). – *Hypergraphes*. – Paris, Gauthier-Villars, 1987.
- [Brilman et al.95a] Brilman (M.) et Vincent (J-M.). – On the Estimation of the Throughput for a Class of Stochastic Resources Sharing Systems.

BIBLIOGRAPHIE

- November 1995. – Submitted to Mathematics of Operations Research.
- [Brilman et al.95b] Brilman (M.) et Vincent (J.-M.). – *Synchronization by Resource Sharing: a Performance Analysis*. – Technical Report 15, Projet IMAG MAI, February 1995.
- [Brilman93] Brilman (M.). – *Modélisation et Évaluation de Performances de Systèmes Parallèles*. – Rapport de DEA, ENS Lyon, June 1993.
- [Chandy et al.84] Chandy (K. M.) et Misra (J.). – The Drinking Philosophers Problem. *ACM Transactions on Programming Languages and Systems*, vol. 6 (4), October 1984, pp. 632–646.
- [Chandy et al.88] Chandy (K.M.) et Misra (J.). – *Parallel Program Design: a Foundation*. – Addison-Wesley, Reading, 1988.
- [Dijkstra68a] Dijkstra (E. W.). – *Cooperating Sequential Processes*, pp. 43–112. – Academic press, New York, 1968.
- [Dijkstra68b] Dijkstra (E. W.). – The Structure of the “THE”-Multiprogramming System. *Comm. A.C.M.*, vol. 11 (5), May 1968, pp. 341–346.
- [Dumas] Dumas (V.). – Communication privée.
- [Eick et al.91] Eick (S.G.), Greenberg (A.G.), Lubachevsky (B.D.) et Weiss (A.). – Synchronous Relaxation for Parallel Simulations with Applications to Circuit-Switched Networks. *In: SCS Multiconferences, Simulation Series*.
- [Forbes et al.96] Forbes (F.) et Ycart (B.). – The Philosophers’ Process on Ladder Graphs. *Communications in Statistics - Stochastic Models*, vol. 12 (4), 1996.
- [Forbes96] Forbes (F.). – *Modèles Markoviens de Ressources Partagées*. – PhD thesis, Université Joseph Fourier, September 1996.
- [Gantmacher60] Gantmacher (F.R.). – *The Theory of Matrices*. – Chelsea Publishing Company, 1960.
- [Gaubert93] Gaubert (S.). – *Performance Evaluation of Timed Automata*. – Technical Report 1922, INRIA, 1993. To appear in IEEE Transactions on Automatic Control.
- [Glasserman et al.95] Glasserman (P.) et Yao (D.). – Stochastic Vector Difference Equations with Stationary Coefficients. *Journal of Applied Probability*, vol. 32 (4), 1995, pp. 851–866.

- [Golub et al.87] Golub (G.H.) et Van Loan (C.F.). – *Matrix Computations*. – Hopkins University Press, 1987.
- [Griffiths90] Griffiths (R.). – Frenkel-Kontorova Models of Commensurate-Incommensurate Phase Transitions. *In: Fundamental problems in statistical mechanics*.
- [Ibragimov62] Ibragimov (I. A.). – Some Limit Theorems for Stationary Processes. *Theory Prob. Applications*, vol. 7, 1962, pp. 349–382.
- [JM] Jean-Marie (A.). – Communication privée.
- [Kelly85] Kelly (F.P.). – Stochastic Models of Computer Communication Systems. *J. Roy. Statist.*, vol. 47 (3), 1985, pp. 379–395.
- [Kermani et al.79] Kermani (P.) et Kleinrock (L.). – Virtual Cut-Through: A new Computer Communication Switching Technique. *Computer Networks*, vol. 3, 1979.
- [Kolchin et al.78] Kolchin (V.F.), Sevast'yanov (B.A.) et Chistyakov (V.P.). – *Random Allocations*. – W.H. Winston and Sons, Washington, 1978.
- [Mairesse93a] Mairesse (J.). – *A Graphical Representation for Matrices in the $(Max, +)$ Algebra*. – Technical Report 2078, INRIA, September 1993.
- [Mairesse93b] Mairesse (J.). – *Products of Irreducible Random Matrices in the $(Max, +)$ Algebra - Part1*. – Technical Report 1939, INRIA, June 1993.
- [Mairesse95] Mairesse (J.). – *Stabilité des systèmes à événements discrets stochastiques. Approche algébrique*. – PhD thesis, École polytechnique, 1995.
- [Marsan et al.86] Marsan (M.A.), Balbo (G.) et Conte (G.). – *Performance models of Multiprocessor systems*. – Cambridge, USA, The MIT Press, 1986.
- [Mitra et al.84] Mitra (D.) et Weinberger (P. J.). – Probabilistic models of database locking: Solutions, computational algorithms, and asymptotics. *J. of the A.C.M.*, vol. 31 (4), October 1984, pp. 855–878.
- [Mitra85] Mitra (D.). – Probabilistic models and asymptotic results for concurrent processing with exclusive and non-exclusive locks. *SIAM J. Comput.*, vol. 14 (4), November 1985, pp. 1030–1051.
- [Morris et al.84] Morris (R.J.T.) et Wong (W.S.). – Performance of concurrency control algorithms with nonexclusive access. *In: Performance'84*, éd. par Gelenbe (E.). – Elsevier.

BIBLIOGRAPHIE

- [Murata89] Murata (T.). – Petri Nets: Properties, Analysis and Applications. *In: Proceedings of the IEEE*, pp. 541–580.
- [Obrien74] O’Brien (G.L.). – Limit Theorems for Sums of Chain-Dependent Processes. *J. Appl. Prob.*, vol. 11, 1974, pp. 582–587.
- [Resing et al.90] Resing (J.A.C.), de Vries (R.E.), Hooghiemstra (G.), Keane (M.S.) et Olsder (G.J.). – Asymptotic Behaviour of Random Discrete Event Systems. *Stoch. Proc. and their Applications*, vol. 36, 1990, pp. 195–216.
- [Ross91] Ross (S.M.). – *A Course in Simulation*. – New-York, Macmillan Publishing Company, 1991.
- [Vincent93] Vincent (J-M.). – *Some Ergodic Results on Stochastic Iterative Discrete Event Systems*. – Technical Report 4, Grenoble, Apache - IMAG, December 1993. Accepted in dynamics of discrete event systems revue.
- [Yakovenko et al.92] Yakovenko (S.) et Kontorer (L.). – Nonlinear semigroups and infinite horizon optimization. *In: Idempotent analysis*, éd. par in Sov. Math. AMS (Adv.).