

Théorie des graphes, et de l'approximation pour le routage, la coloration et l'apprentissage d'équilibres.

Johanne Cohen

Section 27

Table des matières

1	Introduction	7
2	Arbre de plus court chemin	11
2.1	Motivation	11
2.2	Arbre du plus court chemin stable	12
2.3	Analyse des arbres stables	14
2.3.1	Conditions d'existence d'arbres stables	14
2.3.2	Algorithme réparti	15
2.4	Arbres stables avec couleurs	17
2.5	Analogie avec la théorie des jeux	18
3	Algorithmes d'approximation.	21
3.1	Problèmes d'optimisation	21
3.2	Numérotation de sommets sur les graphes d'intervalle	23
3.3	Ordonnancement pour la redistribution de données	26
3.4	Ordonnancement de messages dans les réseaux optiques	28
4	Coloration de graphes	31
4.1	Introduction	31
4.2	A propos de la a -coloration.	32
4.3	A propos de la b -coloration.	33
4.4	A propos de la f -coloration	37
5	Jeux classiques en théorie algorithmique des jeux	39
5.1	Jeux statiques (non-répétés)	39
5.1.1	Meilleure réponse	40
5.1.2	Équilibre de Nash	41
5.1.3	Jeux à deux joueurs symétriques	42
5.2	Les jeux de potentiel ordinal	44
5.2.1	Les jeux de congestion	45
5.2.2	Les jeux d'ordonnancement	46
5.2.3	Les jeux d'allocation de tâches	48
5.3	Évaluation de la qualité des équilibres de Nash	50
6	Des jeux répétés et dynamiques	53
6.1	Jeux répétés	54
6.2	Vers la dynamique des joueurs fictifs	57
6.3	Vers les protocoles de population	60

6.3.1	Description d'un protocole de population	60
6.3.2	Des jeux dans les protocoles de population	61
7	Apprentissage des équilibres de Nash	65
7.1	Dynamique de meilleure réponse	66
7.2	Classes d'algorithmes d'apprentissage.	68
7.3	Exemples de dynamiques	70
7.3.1	Dynamique de réplication	70
7.3.2	Dynamique de <i>logit</i>	71
7.3.3	Dynamique de la meilleure réponse seuillée	72
7.4	Propriétés de convergence.	73
7.4.1	Bornes sur la vitesse de convergence	73
7.4.2	Exemple : le jeu symétrique à deux joueurs	74
7.5	Algorithmes d'apprentissage pour les jeux de potentiel ordinal	76
7.5.1	Caractérisation via des fonctions de Lyapunov	77
7.5.2	Techniques de Martingale	78
8	Conclusion	81
A	Problème d'optimisation locale	87
A.1	La classe PLS	87
A.2	La classe PPAD	88

Préambule

Dans ce document, les publications sous la forme [Dijkstra, 1971] renvoient à la bibliographie page 91.

Les références sous la forme [1] correspondent à des publications personnelles. On trouvera la liste de mes publications page 101.

Le travail présenté dans ce document se focalise sur quelques travaux récents, privilégiant souvent des soumissions à des travaux antérieurs déjà publiés.

Chapitre 1

Introduction

Ce document considère une sélection de mes travaux de recherche. Le fil conducteur de mes recherches se situe dans le domaine de la théorie des graphes, de la complexité, et de la théorie algorithmique des jeux.

Mon travail se place majoritairement dans des domaines d'applications tels que celui des télécommunications, celui de la grille, et celui de la bio-informatique.

Principalement, mes travaux de recherche se placent dans les télécommunications. Les réseaux de télécommunications peuvent se modéliser par un graphe. L'ensemble des sommets correspond aux nœuds du réseau. L'ensemble des arêtes (ou des arcs) représente les liens de communications entre les nœuds. La gestion des communications dans le réseau est donnée par un mode de commutation et par une fonction de routage. Le mode de commutation (exemple : voisin-à-voisin, commutation de circuits, ...) définit la méthode utilisée pour faire transiter l'information ; la fonction de routage définit les itinéraires utilisés par les messages.

Les contraintes matérielles et physiques sur chacun des composants du réseau (routeurs, médiums, topologie ...) se traduisent par différentes contraintes. Par exemple, la présence de liens de communications qui peuvent ou ne peuvent pas traiter les communications bidirectionnelles (full-duplex) conduit à supposer le graphe orienté ou non-orienté. La possibilité que le médium de communication est filaire ou sans fil, permet de poser des contraintes sur les communications (mode 1-port ou Δ -port).

Il est incontestable que la place d'Internet et des réseaux de télécommunications est au centre de nos vies personnelles et professionnelles et que nous sommes de plus en plus consommateurs de services et d'applications garantissant une certaine qualité de services.

La qualité de service est une problématique récurrente dans les réseaux. Elle peut se traduire sous différentes formes en fonction de l'application comme par exemple la minimisation de certaines ressources comme le temps de transmission. Ceci se traduit soit sous forme de problèmes de décision, ou soit sous forme de problèmes d'optimisation avec lesquelles nous évaluons leur complexité ou la difficulté de les résoudre.

Actuellement, différents partenaires économiques apparaissent dans le paysage des réseaux. Ils ont fréquemment des intérêts économiques propres divergents et il est de moins en moins réaliste de supposer que chacun agit uniquement dans l'intérêt des performances globales de l'ensemble.

Par conséquent, d'un point de vue très pragmatique, il est très important d'être capable de garantir que la présence d'un ou de quelques partenaires non-altruistes dans un protocole ne peut mener à une dégradation notable des performances des protocoles

utilisés.

Il ne suffit plus d'élaborer des protocoles ou des algorithmes pour optimiser au mieux les ressources des réseaux mais aussi il faut garantir des performances tout en considérant les aspects économiques de chaque acteur.

Dans cette optique, nous nous sommes intéressées à

1. optimiser des ressources du réseau (par exemple [5, 14, 9] et [23]) ;
2. assurer une qualité de service (par exemple [21],[2, 7]) ;
3. prendre en compte les différents partenaires économiques (par exemple [10, 3],[18]).

Nous reprenons maintenant l'ensemble de nos travaux publiés depuis notre thèse dans ces directions. La suite du document se focalisera sur une sélection de ceux-ci, qui nous paraît la plus intéressante, ou la plus prometteuse.

Dans un premier temps, optimiser les ressources du réseau correspond à construire des algorithmes adaptés à un problème précis. Malheureusement, les méthodes sont différentes si l'on s'intéresse à la redistribution de données entre deux grappes d'ordinateurs (voir section 3.3), aux routages sans déflexion dans un réseau tout optique (voir [9]) ou si l'on considère les communications de groupe dans des applications de simulations distribuées (voir [5]).

Plus particulièrement, dans [5], nous sommes intéressées à la problématique de la Simulation Interactive Distribuée (DIS) correspondant à l'exécution distribuée de programmes d'entraînement militaire de simulation en temps réel sur un réseau. Ce réseau connecte de véritables simulateurs pilotés par des hommes sur des stations de travail qui réalisent le décor évoluant pendant la simulation. Notre travail a consisté à réduire les flots de données, en déterminant un ensemble de groupes suivant certains critères (proximité, type, . . .) pour faire communiquer chaque groupe sur un port multicast. Ainsi les groupes ne recevront que les données propres à leurs membres. Ce groupement permet de mieux utiliser les ressources du réseau, notamment la bande passante. Chaque participant de la simulation est sur un quelconque nœud du réseau (LAN et/ou MAN) et les participants s'échangent des informations en fonction d'une zone de visibilité. En effet, si deux zones de visibilité s'intersectent alors, les participants doivent envoyer des informations. Ces échanges d'informations sont représentés par des *graphes d'Helly*. Nous avons évalué la complexité du problème et proposé des heuristiques basées sur les propriétés de tels graphes. Ce travail s'intègre au travail de thèse de Corentin Durbach (que je ne développerai pas plus dans ce document).

Dans un second temps, assurer une qualité de service permet d'assurer à l'utilisateur des performances sur son utilisation du réseau. Nous avons étudié ce genre de problématiques dans les réseaux optiques en étudiant la dégradation de l'utilisation du réseau si l'on s'assurait une gigue faible dans [20, 21] ou si l'on assurait aucune perte de messages dans le réseau dans [2]. L'utilisation de routages eulériens permet de garantir que tout paquet en transit arrive à destination en un temps fini. Garantir qu'il est possible de router de cette façon en un temps donné se ramène alors à déterminer si le graphe possède de "bons" circuits eulériens.

Dans un troisième temps, à la vue de certains phénomènes observés expérimentalement sur le réseau actuel (l'exemple le plus connu est le partage de fichiers en commun [Adar and Huberman, 2000]), ou ne serait-ce que parce que les différents partenaires impliqués dans les réseaux sont des sociétés commerciales en nombre croissant qui ont des

intérêts économiques propres divergents, il est de moins en moins réaliste de supposer que chacun agit uniquement dans l'intérêt des performances globales de l'ensemble.

Par conséquent, d'un point de vue très pragmatique, il est très important d'être capable de garantir que la présence d'un ou de quelques partenaires non-altruistes dans un protocole ne peut mener à une dégradation notable des performances des protocoles utilisés.

Nous développerons plus en détail cette partie de notre travail dans les chapitres 2, 6 et 7.

Parallèlement, nous nous avons étudié des problèmes issus de la théorie de graphes : par exemple en étudiant les différentes colorations de graphes (voir chapitre 4), ou en définissant un plongement des tores carrés dans une classe particulière de réseaux optiques dans [22] ou en reconnaissant en temps polynomial des graphes de Fibonacci et de Knödel dans [11]. Ces derniers graphes correspondent aux topologies pour lesquelles on a la garantie d'un algorithme d'échange total optimal en nombre de phases en mode voisin-à-voisin 1-port. Cet algorithme de reconnaissance de graphes se base sur la décomposition en cycles et sur leur dénombrement.

Aussi, dans le contexte de la bio-informatique, nous avons étudié des problèmes de théorie des graphes par exemple calculer l'arrangement linéaire optimal (voir chapitre 3), calculer les séparateurs de graphes représentant les réactions chimiques (voir [38] et aussi la thèse d'Antoine Joulie [Joulie, 2007]) intervenant dans le fonctionnement des cellules.

De la même façon, nous avons traité du problème de la génération de séquences aléatoires pour l'analyse des génomes dans [4]. La génération de séquences aléatoires a pour objectif de comparer si les propriétés observées dans les séquences naturelles le sont aussi dans les séquences aléatoires. Cette comparaison permet de déterminer dans quelle mesure ces propriétés sont "biologiquement" pertinentes. Par exemple, le problème est de construire un ensemble de séquences dont un motif particulier est sur-représenté (ou inversement sous-représenté). Notre travail porte sur la génération de séquences ayant des contraintes sur les motifs. Par exemple, une de ces contraintes est de contenir un ensemble de motifs où chaque motif a un nombre d'occurrences bien déterminé. Cette génération porte sur des notions de chemins eulériens dans des graphes de séquences. De plus nous avons montré la difficulté de déterminer si une séquence satisfait les contraintes liées aux motifs est NP-complet. Malgré ce résultat négatif, une heuristique a été proposée et son expérimentation a montré qu'en pratique elle donnait un bon échantillonnage.

Organisation du document Ce mémoire d'habilitation présente une sélection de nos travaux de recherche. Cette sélection vise à illustrer plusieurs facettes des différentes techniques utilisées dans nos recherches.

Le chapitre 2 correspond à une première présentation de quelques conséquences de la présence de partenaires économiques dans les réseaux. Il présente ainsi un survol du problème algorithmique du calcul d'arbre de plus court chemin en présence d'adversaires, et quelques résultats personnels.

Les chapitres 3 et 4 discutent de la frontière entre les problèmes tractables et non-tractables sous deux angles différents. Dans le chapitre 3, nous présentons la théorie de l'approximation, en l'illustrant par trois de nos travaux. Deux des algorithmes d'approximation décrits permettent d'assurer une certaine garantie sur l'utilisation de ressources du réseau, et sur une qualité de service. Le troisième traite d'approximation d'un problème de la théorie des graphes (minimisation d'arrangements linéaires pour les graphes d'in-

tervalle). Le chapitre 4 présente des problèmes de coloration de graphes non-classiques. Il donne un aperçu du fait que changer légèrement les instances du problème permettent de radicalement changer la complexité du problème de la coloration.

Le chapitre 5 est une introduction à la théorie algorithmique des jeux, et des jeux classiques étudiés en informatique. Il sert essentiellement à définir les concepts utilisés par la suite.

Le chapitre 6 introduit certains modèles de dynamisme en théorie des jeux. Il présente la notion de jeux répétés, et les propriétés de leurs équilibres de Nash. Il présente les dynamiques de joueurs fictifs, et les dynamiques de meilleure réponse dans le contexte des protocoles de population. Ce dernier point présente nos travaux visant à comprendre ce qui est programmable par de telles dynamiques.

Le chapitre 7 présente nos travaux à propos de l'apprentissage d'équilibres de Nash. C'est-à-dire visant à construire des comportements tels que si chacun des joueurs adopte ces comportements alors le système global converge vers un équilibre de Nash. Nous nous intéressons en particulier à l'apprentissage d'équilibres sur les jeux de potentiel ordinal.

Chapitre 2

Arbre de plus court chemin

Dans ce chapitre, nous nous intéressons au problème de la construction d'un arbre de plus court chemin. Nous nous focalisons sur une variante naturelle qui consiste à construire des arbres de plus court chemin lorsque chacun des noeuds possède un intérêt propre.

Nous motivons le problème dans la première section. Dans la deuxième section, nous définissons formellement le problème de la construction d'un arbre stable de plus court chemin. Dans la troisième section, nous présentons quelques résultats connus sur le problème. Dans la quatrième section, nous présentons une variante de ce problème où nous considérons que les noeuds peuvent appartenir à des coalitions (couleurs). Dans la cinquième section, nous présentons quelques liens avec la théorie des jeux.

Ce chapitre vise à présenter d'une part quelques résultats personnels, essentiellement dans la quatrième section [10], [42]. Mes résultats personnels relatifs à ce chapitre consistent principalement en d'une part une preuve de la complexité du problème de plus court chemin avec couleurs, d'autre part en des conditions nécessaires pour l'existence d'un arbre stable et enfin un algorithme auto-stabilisant de construction d'arbres stables.

D'autre part, et surtout, ce chapitre vise à présenter par cet exemple introductif le type de problèmes et le type de questions algorithmiques que nous nous posons. En particulier, il nous sert à introduire quelques concepts de la théorie algorithmique des jeux sur lesquels nous reviendrons.

Les travaux personnels de ce chapitre ont partiellement été obtenus en collaboration avec A. Dasgupta, S. Delaët, S. Ghosh, et S. Tixeuil.

2.1 Motivation

Nous nous intéressons au problème classique de la construction d'arbre de plus court chemin. Ce problème est clairement relié au problème du routage dans les réseaux.

Vu les évolutions des réseaux (routage inter-domaine, système de peering, etc ...), l'algorithmique répartie doit maintenant prendre en compte le fait que les noeuds du réseau ont leur intérêt propre même s'ils participent à un processus global. La théorie des jeux est un outil naturel pour modéliser ces phénomènes. Comme nous allons le voir, elle n'apporte pas de solution algorithmique directe pour résoudre la concurrence entre les participants, mais plutôt un outil naturel de modélisation.

Voici une définition plus formelle du problème de l'arbre de plus court chemin :

Problème : Arbre du plus court chemin

Instance : un graphe $G = (V, E)$ où V est l'ensemble des sommets et E est l'ensemble des arêtes, une fonction de poids w sur les arêtes, et un sommet distingué r dans V

Objectif : Construire un arbre de plus court chemin enraciné en r .

Le problème de la construction d'arbre de plus court chemin enraciné en un sommet r dans un graphe $G = (V, E)$ a été étudié de façon intensive dans le cadre centralisé où l'algorithme a une vue complète du graphe. Historiquement, deux principaux algorithmes sont connus : celui de Bellman et Ford [Cormen et al., 2001] et celui de Dijkstra [Dijkstra, 1971]. Par la suite, des variantes réparties (chaque sommet possède une vue locale du graphe : son voisinage et son propre état) de ces algorithmes ont été réalisées, puis adaptées aux réseaux sous forme de protocoles de routage.

Les protocoles de routage construisent les chemins entre deux endroits du réseau pour faire transiter les données. Par exemple, le protocole *Routing Information Protocol (RIP)* correspond à une adaptation de l'algorithme de Bellman et Ford, et le protocole *Open Shortest Path First (OSPF)* à celui de Dijkstra : voir [Huitema, 2001]. En effet les protocoles de routage dynamiques des réseaux locaux (une seule entité gérante) tel que *RIP* et *OSPF* sont essentiellement basés sur des algorithmes distribués pour la résolution du problème du plus court chemin.

Dans le cadre du réseau Internet, ce n'est plus le cas. En effet, Internet est composé de systèmes autonomes (*Autonomous System*) qui s'échangent des messages afin de pouvoir assurer le routage. Chaque système autonome correspond à un ensemble de réseaux et de routeurs sous une administration unique et possède sa propre politique de routage correspondant aux préférences d'administration de cette partie du réseau.

Actuellement, un routage adapté au fait que les nœud du réseau aient leur propre politique de routage est le protocole *Border gateway protocol BGP* (utilisé pour le routage inter-domaine dans l'Internet). Il permet à chacun des systèmes autonomes d'utiliser ses propres politiques de routage pour déterminer ses propres routes. Par conséquent, il peut diverger dans le sens où les différentes politiques de routage peuvent être en conflit et cela peut impliquer une certaine instabilité des routes. Dans [Griffin et al., 2002], la stabilité des routes est étudiée. Il y est prouvé qu'étant donné un ordre de préférence par sommet, le problème de décider la stabilité des routes est NP-complet. De plus, des propriétés sur les instances ayant des routes stables y sont extraites et, à partir de cela, un algorithme auto-stabilisant (simple) est donné. Nous revenons maintenant sur certaines de ces propriétés.

2.2 Arbre du plus court chemin stable

Nous considérons le problème du plus court chemin dans un graphe $G = (V, E)$. En algorithmique classique, des algorithmes centralisés et des algorithmes distribués voire auto-stabilisants (voir [Tel, 2000]) existent pour résoudre ce problème. Quel que soit le modèle retenu (centralisé ou distribué), les sommets collaborent pour construire une structure minimisant la même fonction objectif. Or ces algorithmes ne tolèrent pas que les acteurs impliqués suivent leur propre objectif. La suite de ce chapitre vise à adapter ces algorithmes pour cela.

En effet, nous supposons ici que tous les sommets V de G ont d'abord un but commun : construire un arbre couvrant enraciné en un sommet distingué noté r . Par

ailleurs, ils ont aussi un autre but privé : minimiser le coût de leur chemin dans l'arbre à destination de r . Ceci peut entraîner des conflits entre les ensembles de sommets définis par une métrique commune.

Plusieurs modélisations ont été proposées dans la littérature pour prendre en compte cette concurrence. Par exemple celles développées dans [Griffin et al., 2002] ou [10]. Dans ce document, nous allons présenter le premier travail motivé par le problème du routage inter-domaine introduit dans [Griffin et al., 2002].

Avant de définir formellement le problème, nous allons introduire quelques notations sur les chemins. Un chemin $P = v_0v_1 \dots v_k$ est un chemin dirigé allant du sommet source v_0 vers un sommet destination v_k . De plus si P et Q sont deux chemins respectivement allant de v vers u et de u vers z , alors $P \cdot Q$ sera la notation du chemin obtenu par la concaténation de P avec Q .

Maintenant, le problème de l'arbre de routage est défini comme suit :

Problème : Arbre de routage du plus court chemin stable

Instance : $G = (V, E)$ un graphe simple non-orienté, un sommet distingué r ; chaque sommet v a une liste \mathcal{P}_v de chemins autorisés de v vers r avec un ordre de préférence \preceq_v sur les chemins.

Objectif : Construire un arbre tel que tous les sommets v établissent un chemin entre eux et r satisfaisant la propriété suivante : si v choisit $P = u \cdot Q$ sachant que u est un voisin de v et Q est un chemin de u vers r , alors u choisit le chemin Q .

Par exemple, dans la figure 2.1, le sommet 1 peut choisir entre deux chemins : celui qui passe par le sommet 2 que l'on notera $12r$, et celui qui va directement à r que l'on notera $1r$. De plus il préfère le premier chemin $12r$ par rapport au second $1r$: $1r \preceq_1 12r$. Par contre le sommet 2 préfère le chemin $21r$ à celui $2r$ (c'est-à-dire $2r \preceq_2 21r$). Ici, les intérêts individuels de ces deux sommets ne sont pas compatibles.

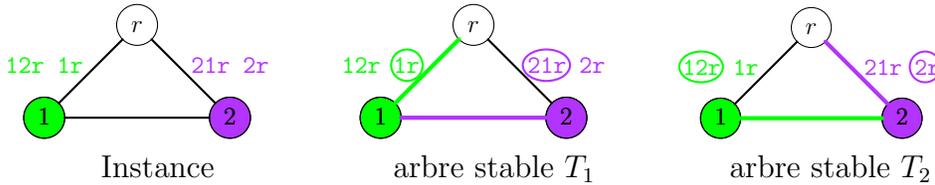


FIG. 2.1 – Instance ayant deux arbres stables T_1 et T_2

Notons qu'un arbre couvrant¹ T de G possède un unique chemin noté $T_{v \rightarrow u}$ entre n'importe quelle paire (v, u) de sommets. Le but est de construire un arbre couvrant *stable* c'est-à-dire non améliorable du point de vue de chaque sommet. En effet, l'objectif de chaque sommet v est de maximiser sa préférence de son chemin vers la racine r .

Nous formulons maintenant les définitions de sommets et d'arbres stables. Notons $\Gamma_G(v)$, le voisinage du sommet v dans le graphe G (c'est-à-dire l'ensemble des sommets voisins de v dans G).

Un sommet v est stable pour un arbre T s'il ne peut pas choisir un autre voisin $z \in \Gamma_G(v)$ qui lui permettrait d'augmenter sa préférence dans l'arbre T' engendré.

Définition 1 (Stabilité des sommets et des arbres) Soit $G = (V, E)$ un graphe avec pour chaque sommet v une liste de chemins \mathcal{P}_v avec leur préférence \preceq_v et r un sommet

¹un arbre couvrant de $G = (V, E)$ est un arbre dont l'ensemble des sommets est V .

distingué de V . Soit T un arbre couvrant de G . Le sommet v est stable dans T si et seulement s'il n'existe aucun chemin p dans \mathcal{P}_v tel que $T_{v \rightarrow r} \preceq_v p$ et si $T_{v \rightarrow r} \in \mathcal{P}_v$. On dira qu'un arbre T est stable dans G si tous ses sommets sont stables.

Considérons un graphe de trois sommets $\{r, 1, 2\}$ de la figure 2.1. Les deux autres dessins représentent des arbres stables T_1 et T_2 pour l'instance du problème. L'arbre T_1 est stable puisque le sommet 2 choisit le chemin de plus forte préférence tandis que le sommet 1 ne peut pas modifier son chemin sans perte de la propriété d'avoir un chemin entre lui et r .

Par contre, il existe des instances où aucun arbre stable existe. Par exemple, considérons l'exemple suivant repris de [Griffin et al., 2002] : le graphe de quatre sommets $\{r, 0, 1, 2\}$ est représenté par la figure 2.2. Remarquons, que les listes des préférences des chemins sont représentées dans l'ordre de croissant.

Pour chacun des sommets j avec $j \in \{0, 1, 2\}$, le chemin de préférence la plus grande est celui passant par le voisin $(j + 1)$. Les indices doivent être interprétés modulo 3. Un arbre couvrant du graphe contient nécessairement au moins l'une des arêtes (j, r) pour atteindre le sommet r . Sous l'hypothèse que le sommet j soit sélectionné par le chemin (jr) , le sommet $(j - 1)$ n'est stable que s'il sélectionne le chemin $(j - 1)jr$ car c'est le chemin de plus grande préférence. Le sommet $(j + 1)$ ne peut que sélectionner le chemin $(j + 1)r$. Par conséquent, le sommet j n'est pas stable car en choisissant le chemin $j(j + 1)r$, il choisit un chemin de préférence la plus grande tout en conservant un chemin entre lui et r . Nous venons de démontrer qu'aucun arbre couvrant de ce graphe ne peut être stable.

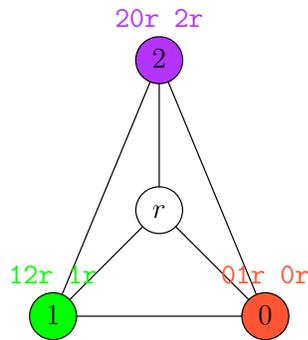


FIG. 2.2 – Instance n'admettant aucun arbre stable.

Dans la section suivante, une analyse plus fine est présentée sur les conditions d'existence et de construction d'arbres stables.

2.3 Analyse des arbres stables

L'exemple présenté dans la section 2.2 (page 14) peut se généraliser en un type de familles d'instances que l'on nommera instances de type *roue avec conflit*.

2.3.1 Conditions d'existence d'arbres stables

En effet, aucun arbre stable existe pour certains éléments de la famille d'instances décrite dans la définition 2 suivante :

Définition 2 (roue avec conflit) Une instance $\mathcal{I} = (G, (\mathcal{P}_v)_{v \in V}, (\preceq_v)_{v \in V})$ est de type roue avec conflit (en anglais *dispute wheel*) ayant k rayons si et seulement s'il existe un sous-ensemble $U = \{u_0, u_1, \dots, u_{k-1}\}$ de sommets de V , des ensembles de chemins $\mathcal{R} = \{R_0, R_1, \dots, R_{k-1}\}$ et $\mathcal{Q} = \{Q_0, Q_1, \dots, Q_{k-1}\}$ tels que pour $j \in [0, \dots, k-1]$,

1. le chemin R_j (resp. Q_j) relie u_j à u_{j+1} (resp. r)
2. $Q_j \in \mathcal{P}_{u_j}$
3. $R_j \cdot Q_{j+1} \in \mathcal{P}_{u_j}$
4. $Q_j \preceq_{u_j} R_j \cdot Q_{j+1}$

Les indices doivent être interprétés comme modulo k . La figure 2.3 correspond à une représentation graphique de graphe de type *roue avec conflit*.

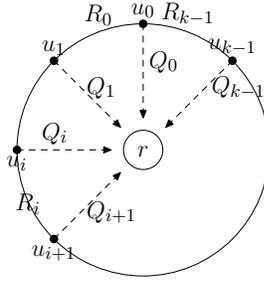


FIG. 2.3 – un graphe de type *roue avec conflit*.

Nous pouvons remarquer que l'instance représentée dans la figure 2.2 est une instance de type *roue avec conflit* de taille 3. En fait, pour les instances de ce type ayant un nombre impair de rayons, aucun arbre stable n'existe. A partir de cette constatation, il a été montré le théorème suivant :

Théorème 1 ([Griffin et al., 2002]) *Décider s'il existe un arbre de routage de plus court chemin stable est NP-complet.*

La réduction polynomiale utilisée dans la preuve de ce théorème se base sur la construction d'un graphe contenant des instances de type *roue avec conflit* à partir de 3-SAT. L'astuce de la preuve est d'associer une clause à un graphe de type roue de telle façon à ce que pour toute affectation qui ne satisfait pas une clause, alors, ce graphe de type *roue avec conflit* ne possède pas d'arbre stable (et réciproquement).

De plus,

Théorème 2 ([Griffin et al., 2002]) *Pour toute instance ne contenant pas une roue avec conflit, il existe un unique arbre de routage de plus court chemin stable.*

A partir de ce résultat, un algorithme réparti glouton, va pouvoir construire un arbre stable de plus court chemin. Avant de le décrire, nous allons introduire quelques notions classiques.

2.3.2 Algorithme réparti

L'algorithme localement exécuté sur chaque sommet est présenté au moyen de variables partagées et d'actions. Chaque sommet possède ses propres variables et communique avec les autres processeurs par passage de messages par des canaux de communication FIFO.

Un système de transitions est un couple $S = (C, \rightarrow)$, où C est un ensemble de configurations et \rightarrow une relation binaire sur C . Une configuration est le produit cartésien à un instant donné de toutes les variables de tous les sommets du réseau. La relation \rightarrow permet de modéliser les changements d'états locaux des sommets par le programme.

Une exécution de S est une suite maximale de configurations (finie ou infinie) $E = (\gamma_0, \gamma_1, \dots, \gamma_i, \dots)$ telle que pour tout $i \geq 0$, $\gamma_i \rightarrow \gamma_{i+1}$.

L'algorithme que l'on va considérer est simple. Chaque sommet u conserve son choix, son chemin entre lui et la destination dans une variable locale. Lors de la réception d'un message d'un sommet voisin v , le sommet u conserve cette information dans une table et il vérifie que son choix courant est bien le meilleur. Si ce n'est pas le cas il le réactualise et envoie son nouveau chemin à tous ses voisins. Le traitement local se réalise de façon atomique².

Théorème 3 ([Griffin et al., 2002]) *Cet algorithme glouton converge vers un arbre stable si l'instance ne contient pas de roue avec conflit.*

Pour cela, il a été prouvé qu'à partir de toute exécution débutant par une configuration arbitraire, un arbre stable est construit (propriété de sûreté³).

Il est à noter que les graphes contenant des *roues avec conflit* peuvent parfois avoir des arbres stables. Il suffit par exemple d'étendre l'exemple de 2.1 pour former *une roue avec conflit* de 4 rayons. Cet algorithme ne converge pas sur cet exemple, et cela implique qu'il peut diverger pour certains types d'exécutions même si l'instance du problème possède un arbre stable.

Un algorithme pour y remédier a été proposé par [Griffin and Wilfong, 2000]. Il détecte des cycles liés à des conflits de politiques grâce à l'historique des chemins. Plusieurs améliorations ont été proposées ultérieurement en utilisant des mécanismes de jetons [Ahronovitz et al., 2006], des algorithmes aléatoires [Ibrahim and Matta, 2004].

Dans [Griffin et al., 2002], la seule hypothèse sur l'ordre de préférence est qu'elle est fixée, connue dès le départ et surtout qu'elle ne change pas au cours du temps. Que se passe-t-il si l'ordre de préférence peut se modifier au cours du temps ? Cela revient à comprendre quelles sont les propriétés (s'il en existe) de l'ordre de préférence qui permettent de conserver la convergence de l'algorithme de [Griffin et al., 2002]. A ma connaissance, ces questions ne sont pas complètement fermées. Nous considérons dans la section suivante des réponses partielles en considérant des algorithmes auto-stabilisants.

Définition 3 (algorithme autostabilisant) *Un système de transition $S = (C, \rightarrow)$ est auto-stabilisant pour la spécification P (un prédicat sur l'ensemble des exécutions) s'il existe un ensemble $L \subseteq C$ de configurations légitimes vérifiant les propriétés suivantes :*

1. Correction : toute exécution débutant par une configuration dans L satisfait P ;
2. Convergence : toute exécution atteint une configuration dans L .

²cette procédure ne peut pas être interrompue.

³toute exécution débutant par une configuration arbitraire atteint une configuration satisfaisant le prédicat P .

2.4 Arbres stables avec couleurs

Dans [10], [42], nous avons étendu le problème de la construction d'arbre stable de plus court chemin au cas où les sommets pourraient ne pas avoir les mêmes politiques de routage.

Pour cela, l'ensemble des sommets V est découpé en p parties disjointes V_1, V_2, \dots, V_p tel que $V = \bigcup_{i=1}^p V_i$: chaque élément de cette partition correspondant à une couleur est associée à un ordre de préférence.

Dans [10], [42] et [Dasgupta et al., 2006], l'ordre de préférence est défini comme l'opposé du coût d'un chemin : moins le coût d'un chemin est élevé, plus le joueur préfère le chemin. Pour modéliser le coût, le graphe G est pondéré par une fonction $w : E \rightarrow \mathbb{N}^p$. Pour chaque entier $i \in [1 \dots p]$, la fonction $w_i : E \rightarrow \mathbb{N}^*$ est définie comme

$$\forall e \in E, w_i(e) = x_i \text{ si et seulement si } w(e) = (x_1, \dots, x_i, \dots, x_p).$$

Le coût du chemin C pour la couleur i est $w_i(C) = \sum_{e \in C} w_i(e)$. Par rapport aux notations précédentes, pour un sommet v , l'ensemble des chemins autorisés \mathcal{P}_v est l'ensemble de tous les chemins le reliant à r . Nous pouvons remarquer que dans certains graphes, la cardinalité de cet ensemble peut être exponentielle par rapport au nombre de sommets du graphe.

L'ordre \preceq_v pour le sommet v se définit alors de la façon suivante : pour tout couple (P, Q) de deux chemins, nous avons

$$P \preceq_v Q \text{ si et seulement si } w_i(Q) \leq w_i(P)$$

sachant que le sommet v est de couleur i .

Plus formellement,

Problème : Arbre du plus court chemin stable

Instance : $G = (V, E)$ un graphe non-orienté, un sommet distingué r ;
ayant une fonction poids sur les arêtes $w : E \rightarrow \mathbb{N}^p$.

Objectif : Construire un arbre T tel que tous les sommets v établissent un chemin entre eux et r et tel que T soit stable, i.e :

$$\forall z \in \Gamma_G(v), w_i(z \cdot T_{z \rightarrow r}) \geq w_i(T_{z \rightarrow r}) \text{ sachant que } v \text{ a la couleur } i.$$

Nous pouvons constater que l'exemple de la figure 2.2 correspond aussi à un problème d'arbre stable dans le cas où le nombre de couleurs est égal à trois. Par contre, puisque chaque sommet a son propre ordre de préférence, cet exemple ne correspond pas à une instance du problème d'arbre stable ayant deux couleurs.

Dans [10] nous avons construit des exemples d'instances ne possédant pas d'arbre stable et prouvé :

Théorème 4 ([10]) *Décider s'il existe un arbre de plus court chemin stable est NP-complet même si le nombre de couleurs est 2.*

De plus, nous avons décrit dans [42] des familles d'instances possédant au moins un arbre stable.

Contrairement à ce qui se passe dans la section précédente, il n'y a pas unicité de l'arbre. Pour cette raison, il a été prouvé dans [Dasgupta et al., 2006] que l'algorithme glouton est seulement faiblement auto-stabilisant.

Théorème 5 ([Dasgupta et al., 2006]) *L’algorithme glouton calcule un arbre stable et il est faiblement auto-stabilisant.*

Notons, qu’un système est *faiblement auto-stabilisant* s’il existe une exécution qui atteint une configuration légitime à partir de toute configuration de départ. En fait, par rapport à la notion d’auto-stabilisation classique, seule la propriété de convergence est modifiée : pour la notion d’auto-stabilisation faible, seule une exécution atteignant une configuration légitime suffit contrairement à la notion de (fortement) auto-stabilisant où toutes les exécutions doivent atteindre une configuration légitime.

Nous avons étendu le résultat précédent :

Théorème 6 ([42]) *L’algorithme glouton converge vers un arbre stable de façon auto-stabilisante si l’instance ne contient pas certains motifs (roues avec conflit généralisées).*

Mais cet algorithme est complexe en terme d’échanges de messages puisqu’il est obligé de construire pour chaque sommet tout le graphe avec sa métrique de coût.

Nous conjecturons que l’algorithme proposé dans [Dasgupta et al., 2006] est en fait auto-stabilisant pour les mêmes instances que le théorème 6. Cette conjecture s’inspire d’arguments liés aux travaux sur les jeux de potentiel ordinal (voir les chapitres 5, 7) dont nous parlerons plus longuement lors de l’étude de ces jeux.

A ce jour, nos travaux portent essentiellement sur l’ordre de préférence évoqué plus haut basé sur le coût (longueur) des chemins. Un de mes travaux en cours [35] porte sur la construction d’un arbre *de flot maximum* (en anglais *maximum flow tree*). Les résultats sont identiques à ceux obtenus pour le problème de la construction du plus court chemin pour cette métrique. Cependant, les techniques de preuve diffèrent et il nous paraît difficile de généraliser ces résultats pour un ordre de préférence générique ou pour de larges classes de métriques.

2.5 Analogie avec la théorie des jeux

Dans le contexte précédent, chaque sommet v a son propre objectif correspondant à l’ordre de préférence \preceq_v . Cette optique est proche de celle de la théorie des jeux permettant de modéliser la concurrence entre deux ou plusieurs joueurs via un jeu. C’est pour cette raison que nous introduisons ici certains concepts de la théorie des jeux.

Définition 4 (jeu) *Un jeu (non-répété) est composé d’un ensemble de joueurs J . Chaque joueur i possède un ensemble noté S_i de stratégies pures. Notons \mathcal{S} le produit cartésien des ensembles de stratégies pures $\mathcal{S} = \times_{i \in J} S_i$. Un profil pur de stratégies pures $s = (s_1, \dots, s_n)$ est un élément de \mathcal{S} .*

Pour chaque profil $s \in \mathcal{S}$, chaque joueur $i \in J$ possède une *fonction de coût* $c_i : \mathcal{S} \rightarrow \mathbb{R}$ qui représente l’ordre de préférence. Il est à noter, que traditionnellement dans la théorie des jeux, la notion de *fonction d’utilité* joue le rôle de modélisation de la préférence d’un joueur : chaque joueur cherche à maximiser son utilité.

Par contre, dans nombre de contextes liés à la théorie algorithmique des jeux, il est plus naturel de considérer des coûts : chaque joueur cherche à minimiser son coût. Toutes les définitions liées à la théorie des jeux seront exprimées dans ce document par rapport aux coûts. On peut toujours voir un coût comme l’opposé d’une utilité.

La représentation sous la *forme normale du jeu* est un triplet $(J, \mathcal{S}, (c_i)_{i \in J})$ avec J ensemble des joueurs, \mathcal{S} ensemble des profils, et $(c_i)_{i \in J}$, l'ensemble des fonctions de coût.

Définition 5 (Equilibre de Nash pur) *Un équilibre de Nash pur est un profil pur $s = (s_1, \dots, s_n)$ tel que*

$$\forall i \quad \forall s'_i \in S_i \quad c_i(s_1, \dots, s_i, \dots, s_n) \leq c_i(s_1, \dots, s'_i, \dots, s_n).$$

Par exemple, dans le problème de l'arbre de routage de plus court chemin stable, l'ensemble de joueurs J correspond à l'ensemble V des sommets du graphe. L'ensemble des stratégies de chaque joueur v est la liste \mathcal{P}_v de chemins autorisés et la fonction de coût de v correspond à son ordre de préférence \preceq_v sur les chemins pour [Griffin et al., 2002] et aux coût des chemins dans [10], [42] et [Dasgupta et al., 2006]. Dans le jeu représenté par la figure 2.1, il existe deux joueurs : le joueur 1 ayant 1r, 12r comme stratégies pures et le joueur 2 ayant 2r, 21r comme stratégies pures.

Plus précisément, la fonction de coût quantifie l'ordre de préférence des joueurs correspondant en fait à une liste triée des différentes stratégies (chemins). Il n'existe pas de règle universelle pour quantifier la notion de préférence d'un joueur entre les stratégies et par rapport à l'extérieur (stratégies des autres joueurs). Par exemple, les tableaux de la table 2.1 du jeu défini à la page 13 représenté par la figure 2.1 représentent des coûts de différents joueurs qui respecte l'ordre de préférence du jeu. Bien sûr, il y en a d'autres.

		joueur 2	
		21r	2r
joueur 1		12r	2
		1r	0
		1	1
		Coût du joueur 1	

		joueur 1	
		12r	1r
joueur 2		21r	2
		2r	0
		1	1
		Coût du joueur 2	

TAB. 2.1 – Coût du jeu représenté dans la figure 2.1

La représentation sous la *forme stratégique du jeu* de 2 joueurs est un tableau tel que chaque élément du tableau représente un profil du jeu et tel qu'il contient les différents coûts des deux joueurs. La première colonne (resp. ligne) représente toutes les stratégies pures du joueur 1 (resp. 2). Chaque case du tableau, contient un couple dont le premier (resp. deuxième) élément est le coût du joueur 1 (resp. 2) si le jeu est dans ce profil. Voir la table 2.2 pour avoir la forme stratégique du jeu défini à la page 13 représenté par la figure 2.1.

		joueur 2	
		21r	2r
joueur 1		12r	(2,2)
		1r	(0,1)
		(1,0)	(1,1)

TAB. 2.2 – Coût du jeu représenté dans la figure 2.1

La table 2.3 représente la correspondance entre les différentes notions introduites dans la section 2.2 pour le problème de l'arbre de plus court chemin et la théorie des jeux.

Nous pouvons interpréter l'algorithme de [Griffin et al., 2002] comme un algorithme d'apprentissage d'équilibre de Nash pur. A chaque étape tous les joueurs ont choisi leur

Théorie des jeux	Problème de plus court chemin
Joueurs	Sommets du graphe
Ensemble de stratégies	Ensembles des chemins
Fonction d'utilité	Ordre de préférence
Fonction de coût	L'opposé de l'ordre de préférence
Équilibre de Nash	Arbre stable

TAB. 2.3 – Équivalence entre la théorie des jeux et l'arbre de plus court chemin stable

stratégie pure. A la fin de cette étape, un seul joueur réactualise sa stratégie pure en sélectionnant toujours sa meilleure réponse. S'il y a convergence, l'algorithme converge vers un arbre stable : un arbre stable correspond à un équilibre de Nash pur au sens de la théorie des jeux. Cet algorithme correspond à une approximation d'une dynamique de la meilleure réponse. Ces termes vont être définis et discutés beaucoup plus longuement dans le chapitre 8.

Chapitre 3

Algorithmes d'approximation.

Dans ce chapitre, nous reprenons certains de nos résultats relatifs à la construction d'algorithmes d'approximation ou la preuve de non-approximabilité de problèmes.

Dans la première section, nous rappelons les bases de la théorie de l'approximation. Nous présentons ensuite plusieurs de nos résultats relatifs à la numérotation de sommets sur les graphes d'intervalle (essentiellement [27]). Nous présentons ensuite nos travaux sur l'ordonnancement pour la redistribution de données (essentiellement [32], [14]) puis sur l'ordonnancement de messages dans les réseaux optiques (essentiellement [7], [20], [21], [33]).

Chacun de ces travaux est présenté via une discussion de l'état de l'art des problèmes correspondants.

3.1 Problèmes d'optimisation

Dans cette section, nous introduisons la notion d'approximation d'un problème d'optimisation sous l'angle de la complexité. Face à un problème d'optimisation (par exemple calculer le nombre chromatique d'un graphe, la distance ou le chemin entre deux sommets du graphe), avoir un algorithme le plus *efficace* possible est important. La notion d'efficacité est souvent cruciale. En particulier lorsque les problèmes de décision sont NP-difficiles, calculer la solution exacte peut prendre beaucoup de temps de façon pratique (des mois, voir des années). Dans de tels cas, il paraît judicieux d'avoir des algorithmes calculant une solution proche de l'optimum en un temps polynomial.

A partir de cette constatation, la théorie de l'approximation permet d'évaluer la qualité d'une solution calculée par un algorithme et de classifier les différents problèmes.

Voici une définition plus formelle d'un problème d'optimisation :

Définition 6 (Problème d'optimisation) *Un problème d'optimisation Π est un quadruplet $(I_\Pi, SOL_\Pi, m_\Pi, goal_\Pi)$ où*

- I_Π est l'ensemble des instances ;
- $SOL_\Pi(x)$ est l'ensemble des solutions possibles pour chaque instance $x \in I_\Pi$;
- m_Π est une fonction (mesure) qui associe une valeur à chaque solution possible d'une instance ;
- $goal_\Pi$ détermine s'il s'agit de maximiser ($goal_\Pi = max$) ou de minimiser ($goal_\Pi = min$).

Bien sûr, l'objectif d'un problème d'optimisation Π est de construire une solution y^* de l'instance

$$m_{\Pi}(x, y^*) = \text{goal}_{\Pi} \{m_{\Pi}(x, y) : y \in \text{SOL}_{\Pi}(x)\}.$$

Par exemple, le problème de calculer le nombre chromatique d'un graphe correspond à déterminer le nombre de couleurs pour colorier ce graphe sans que deux sommets adjacents aient la même couleur.

Le problème de décision associé est le suivant :

Instance : un graphe non-orienté G et un entier K

Question : existe-t-il une coloration propre du graphe ayant un nombre de couleurs inférieur à K ?

Ce problème est NP-complet.

Sa formulation en tant que problème d'optimisation avec la définition précédente est la suivante :

- I_{Π} : est un ensemble des graphes non-orientés G ;
- $\text{SOL}_{\Pi}(G)$: est l'ensemble des colorations propres de G ;
- m_{Π} : est une fonction retournant le nombre de couleurs dans une coloration ;
- c'est un problème de minimisation ($\text{goal}_{\Pi} = \min$).

Nous introduisons les notations suivantes pour une instance x d'un problème d'optimisation Π :

- $|x|$ est la taille de l'instance, c'est-à-dire la longueur de sa représentation binaire.
- $m_{\Pi}^*(x)$ est la valeur de la fonction m_{Π} d'une solution optimale du problème pour l'instance x .

Maintenant nous pouvons définir :

Définition 7 (La classe NPO) *Un problème d'optimisation Π est dans la classe NPO Π si*

- l'ensemble des instances est reconnaissable en temps polynomial ;
- il existe un polynôme q tel que pour chaque instance x et pour toute solution $y \in \text{SOL}_{\Pi}(x)$, on a $|y| \leq q(|x|)$;
- pour chaque instance x et pour tout y tel que $|y| \leq q(|x|)$, le problème de décider si $y \in \text{SOL}_{\Pi}(x)$ peut être résolu en temps polynomial ;
- La fonction m_{Π} est calculable en temps polynomial.

Remarquons que les problèmes de décision dans NP correspondent à des problèmes d'optimisation dans la classe NPO. Maintenant, nous allons introduire la notion d'évaluation de la qualité des solutions.

Définition 8 (Rapport à l'optimum) *Étant donnée une instance x d'un problème d'optimisation Π , une solution y possible de x a le rapport à l'optimum*

$$r_{\Pi}(x, y) = \max \left\{ \frac{m_{\Pi}(x, y)}{m_{\Pi}^*(x)}, \frac{m_{\Pi}^*(x)}{m_{\Pi}(x, y)} \right\}$$

Définition 9 (Algorithme d'approximation) *Un algorithme \mathcal{A} donne une approximation à un facteur $f(n)$ près en temps polynomial pour un problème d'optimisation Π si pour chacune de ses instances x de taille n , il calcule en temps polynomial une solution y possible pour x telle que $r_{\Pi}(x, y) \leq f(n)$.*

De plus, si un tel algorithme existe pour le problème d'optimisation Π , Π est dit approximable à un facteur $f(n)$ près.

Voici, une classe intéressante de problèmes ayant des algorithmes d'approximation de bonne qualité :

Définition 10 (La classe APX) La classe APX est la classe de tous les problèmes de NPO approximables à un facteur constant près (c'est-à-dire tels qu'il existe une approximation à un facteur r près en temps polynomial pour une certaine constante $r \geq 1$).

Malheureusement, il existe des problèmes d'optimisation qui semblent être durs à approximer. Une technique classique (utilisée dans [34] par exemple) pour prouver la non approximation d'un problème est celle de la *réduction de préservation d'intervalle* (en anglais *gap technique*) : on applique le théorème ci-dessous (voir [Ausiello et al., 1999] pour la preuve).

Théorème 7 Soit \mathcal{P} un problème de décision NP-complet, et soit Π un problème d'optimisation de minimisation dans NPO. Supposons qu'il existe des fonctions calculables en temps polynomial $f : I_{\mathcal{P}} \mapsto I_{\Pi}$ et $c : I_{\Pi} \mapsto \mathbb{N}$ et une constante $r > 0$ telles que pour toute instance x de \mathcal{P}

$$m_{\Pi}^*(f(x)) = \begin{cases} c(x) & \text{si } x \text{ est une instance positive} \\ (1+r)c(x) & \text{sinon} \end{cases}$$

Alors, à moins que $P = NP$, il n'existe pas d'algorithme d'approximation à un facteur r près de Π en temps polynomial.

Bien sûr, la technique reste la même pour un problème de maximisation.

3.2 Numérotation de sommets sur les graphes d'intervalle

Certains problèmes d'optimisation liés aux numérotations de sommets restent ouverts. Plus précisément, une *numérotation de sommets* d'un graphe $G = (V, E)$ est une numérotation des sommets $L : V \rightarrow [1, \dots, n]$ avec n correspondant au nombre de sommets de V . Le *poids* d'une arête $e = (u, v)$ d'une numérotation de sommets L dans G est $w_L(e) = |L(u) - L(v)|$. A partir de cette définition, plusieurs problèmes d'optimisation ont été introduits comme par exemple :

- Minimiser la *largeur de bande* (*bandwidth* en anglais). La largeur de bande d'une numérotation L d'un graphe G que l'on notera $bw_L(G)$ correspond à

$$bw_L(G) = \max \{w_L(e) : e \in E\}$$

- Minimiser l'*arrangement linéaire* (*linear arrangement* en anglais). L'arrangement linéaire d'une numérotation L d'un graphe G que l'on notera $ola_L(G)$ correspond à

$$ola_L(G) = \sum_{e \in E} w_L(e)$$

- Minimiser la *largeur de coupe* (*cutwidth* en anglais). La largeur de coupe d'une numérotation L d'un graphe G que l'on notera $cw_L(G)$ correspond à

$$cw_L(G) = \max_{1 \leq i \leq n} |\{u, v\} \in E | L(u) \leq i \leq L(v)\}|$$

Les problèmes de décision (minimisation) associés à ces trois problèmes sont NP-difficiles dans les graphes quelconques [Garey and Johnson, 1979].

Nous allons discuter de ces paramètres sur des types de graphes classiques comme par exemple les graphes d'intervalle.

Définition 11 (graphe d'intervalle) *Un graphe est un graphe d'intervalle s'il existe une bijection entre les sommets du graphe et un ensemble de segments d'une droite telle que deux sommets u et v sont reliés par une arête si et seulement si les segments correspondant à u et v se superposent.*

Beaucoup de problèmes classiquement NP-complets deviennent polynomiaux pour les graphes d'intervalle. En fait, la majorité des algorithmes polynomiaux se basent sur leur représentation graphique ou sur leur numérotation de sommets *reo* ou *leo*.

Définition 12 *La numérotation sommet *reo* (resp. *leo*) du graphe G par rapport à une représentation graphique est telle que l'ordre des sommets est défini par l'extrémité droite (resp. gauche) des intervalles.*

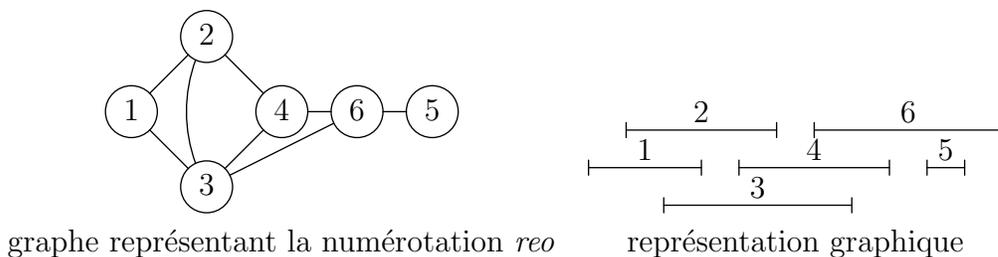


FIG. 3.1 – Graphe d'intervalle et sa représentation graphique

A propos de la largeur de bande. Le problème de la minimisation de la largeur de bande est NP-difficile dans les graphes quelconques. Il reste aussi dur si les graphes sont des arbres [Monien, 1986]. De plus, en général, la largeur de bande ne peut pas s'approximer à un facteur constant multiplicatif en temps polynomial [Unger, 1998] mais il peut s'approximer en un temps polynomial avec un facteur multiplicatif $O(\log^{9/2} n)$ [Feige, 2000].

Pour les graphes d'intervalle, un algorithme décrit dans [Kleitman and Vohra, 1990], résout un problème de décision 'A-t-on $bw(G) \leq k$?' en $O(nk)$ opérations et construit une numérotation de sommets ayant une largeur de bande minimum en $O(n^2 \log n)$ opérations. De plus, dans [Sprague, 1994] une amélioration de cet algorithme a été proposée en répondant à la question de décision en $O(n \log n)$ opérations et en construisant effectivement une numérotation en $O(n \log^2 n)$ opérations. Par contre, la complexité de ce problème pour les graphes de permutation est encore une question ouverte.

Pour les graphes d'intervalle, par la remarque faite dans [Kratsch and Stewart, 2002], la numérotation sommet (*reo*) satisfait la propriété suivante : pour chaque paire de sommets adjacents u et v tel que $reo(u) < reo(v)$, chaque sommet w tel que $reo(u) < reo(w) < reo(v)$ est adjacent à v . Ce qui implique que

Théorème 8 ([Kratsch and Stewart, 2002]) *Les numérotations reo et leo donnent une 2-approximation pour le problème de la largeur de bande.*

A propos de l'arrangement linéaire Calculer un arrangement linéaire optimal est NP-complet dans un graphe biparti [Even and Shiloach, 1975]. Par contre, il est possible de le calculer en temps polynomial pour des arbres [Goldberg and Klipker, 1976], [Chung, 1988], [Shiloach, 1979], mais aussi pour des classes de graphes comme les grilles, les hypercubes [Díaz et al., 2002].

Le meilleur algorithme d'approximation connu est à un facteur multiplicatif $O(\log n)$ [Uriel Feige and Lee, 2007]. Pour les classes classiques comme les graphes d'intervalles, les graphes de permutation et les co-graphes, ce problème est NP-complet :

Théorème 9 ([27]) *Déterminer si $ola(G) \leq k$ est NP-complet pour les graphes d'intervalle, les graphes de permutation et les graphes co-comparables.*

Nous nous sommes intéressés à ce problème lié à la modélisation des interactions de protéines de [Farach-Colton et al., 2004]. Farach-Colton et al. ont utilisé les graphes d'intervalle pour étudier la formation d'un type de ribosomes via des interactions des différentes protéines.

Si l'on considère que le graphe est construit par une succession d'insertions de nouveaux sommets en fonction de la numérotation *reo*, alors à chaque insertion, une étoile est rajoutée de centre le nouveau sommet. Ce qui implique que le coût supplémentaire des ces ajouts d'arêtes correspond à celui d'une étoile. Par une simple analyse, pour une étoile de n sommets, il est facile de voir que la meilleure numérotation correspond à celle numérotant le centre de l'étoile à $\lceil n/2 \rceil$ et que la pire correspond à celle numérotant le centre de l'étoile à n ou à 1. Le rapport de ce paramètre pour ces deux numérotations est de 2.

Cela signifie que l'algorithme qui met sous forme d'intervalles le graphe et qui trie les intervalles en les numérotant selon l'ordre *reo* est une 2-approximation.

Théorème 10 ([27]) *L'ordre reo est une approximation à un facteur multiplicatif 2 près.*

Cette borne est atteinte par le graphe correspondant à l'étoile (voir figure 3.2)

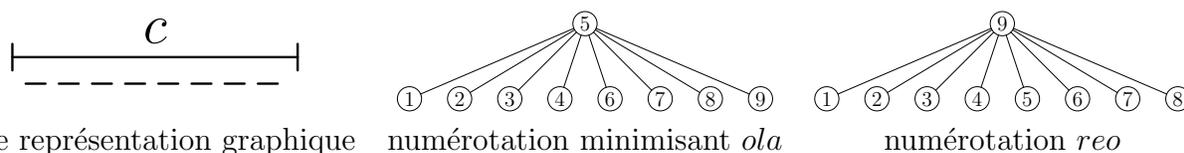


FIG. 3.2 – Etoile ayant huit feuilles et ses deux numérotations de sommets la meilleure et la pire.

De plus, pour les graphes co-comparables,

Théorème 11 ([27]) *Il existe un algorithme polynomial d'approximation pour les graphes co-comparables (basé sur leur ordre) à un facteur multiplicatif huit près.*

Pour l'instant, aucun exemple de graphe a été construit permettant d'affirmer que cette borne supérieure de huit est atteinte. Les cas pathologiques identifiés sont à six de l'optimal.

A propos de la largeur de coupe Ce problème est aussi NP-complet même si le graphe est planaire ayant un degré maximum de trois [Monien and Sudborough, 1986], grilles incomplètes [Díaz et al., 2001]. Par contre, pour les arbres, le problème devient polynomial [Yannakakis, 1985]. Il existe un polynomial algorithme d'approximation pour un graphe général avec un facteur multiplicatif $O(\log^2 n)$ [Leighton and Rao, 1988].

Par contre, déterminer si $cw(G) \leq k$ est NP-complet pour les graphes d'intervalles est une question ouverte. Récemment, un algorithme polynomial calcule ce paramètre dans des graphes d'intervalle particuliers [Heggernes et al., 2008] (graphes d'intervalle propres et graphes seuillés (*threshold* en anglais)).

Tous ces travaux tentent de trouver la frontière entre les problèmes difficiles et ceux polynomiaux à résoudre pour des graphes classiques.

3.3 Ordonnancement pour la redistribution de données

Nous avons travaillé sur le problème de la redistribution de données. Ce travail s'inscrivait dans l'Action de Recherche Incitative de l'INRIA *RedGrid*. La redistribution de données a été étudiée dans le contexte du calcul parallèle dans [Afrati et al., 2005], [Afrati et al., 2002], [Bongiovanni et al., 1981], [Crescenzi et al., 2001], ainsi que dans [Desprez et al., 1998]. Elle survient lorsque des données réparties sur un ensemble de processeurs doivent être transmises (éventuellement vers un autre ensemble de processeurs) le plus efficacement possible. Il s'agit d'ordonner les communications de manière à minimiser la durée totale de celles-ci.

Nous nous sommes penchées sur la résolution de ce problème dans un scénario particulier : des ordinateurs de deux grappes s'échangent des données via un réseau haut débit. Par exemple, supposons que le premier ensemble d'ordinateurs soit un ensemble de 50 PCs avec des cartes réseaux à 1 Gbit/s et que le second soit un autre ensemble de 200 PCs avec des cartes à 100 Mbit/s. Supposons que le réseau les connectant soit un réseau à 10 Gbit/s. Alors le premier ensemble ne pourra pas émettre plus de 10 communications à 1 Gbit tandis que le second ne pourra pas recevoir plus de 100 communications à 100 Mbits. Le nombre de communications simultanées maximum est le minimum des PCs dans les deux grappes soit $k = 10$.

En fonction des contraintes physiques, un modèle de communications peut se dessiner. Par exemple, chaque ordinateur ne peut gérer qu'une seule communication à la fois (mode 1-port). De plus, le réseau haut débit est une ressource critique, et le nombre de communications maximales que peut supporter le réseau inter-connectant les deux grappes d'ordinateurs au même instant est k .

Les communications sont uniquement entre les ordinateurs des différentes grappes. Elles sont représentées sous forme de matrices où chaque élément correspond à la quantité d'information qui doit être échangée. La modélisation naturelle est de représenter

ce problème sous forme de graphe biparti $G = (V_1, V_2, E)$ pondéré par une fonction $w : E \rightarrow \mathbb{Q}^+$ [Crescenzi et al., 2001].

Chaque ordinateur de la première grappe correspond à un sommet de V_1 . Il en est de même pour la seconde grappe. Une arête représente un échange d'information et son poids correspond à la durée du transfert total entre deux ordinateurs.

Pour exécuter les communications nous décomposons la redistribution en phases. Chaque phase représente des communications entre un ensemble de processeurs des deux ensembles d'ordinateurs en respectant le modèle 1-port, et en autorisant la préemption : un transfert de données peut être effectué en plusieurs étapes. A chaque phase, plusieurs communications sont réalisées simultanément et chaque phase a un coût d'initialisation dénoté par β et un coût de transfert de données dépendant de la quantité d'information à traiter.

De plus, des travaux comme [Afrati et al., 2005], [Crescenzi et al., 2001] supposent qu'il n'existe pas de goulot d'étranglement, c'est-à-dire que le réseau haut débit peut supporter autant de communications simultanées. Ce n'est pas la ressource critique. Dans [32], nous avons pris en compte cette contrainte. En généralisant le problème de la redistribution dans le cas où le nombre de communications simultanées est borné par un paramètre k entre les deux grappes de machines. Ce problème survient en particulier lorsque l'on veut redistribuer des données entre des processeurs de deux grappes de PCs reliées par un réseau à haut débit dans le cas d'applications de couplage de code ou de visualisation interactive à distance d'images calculées en parallèle.

Ce problème a été partiellement étudié dans le contexte d'accès au médium de communication pour les satellites [Gopal et al., 1982], [Gopal and Wong, 1985]. En particulier, dans [Bongiovanni et al., 1981], un algorithme polynomial a été étudié pour le cas où la phase d'initialisation est négligée ($\beta = 0$). Dans ce même contexte, lorsque le découpage des messages (préemption) n'est pas autorisé, le problème devient NP-complet et des heuristiques ont été proposées dans le même article.

Dans le cas où la phase d'initialisation de chaque étape possède un coût, lorsque le réseau haut débit n'est pas un goulot d'étranglement, il a été montré que minimiser le temps de communication est NP-complet [Even et al., 1976], [Gopal and Wong, 1985] et qu'il n'est pas approximable à $\frac{7}{6}$ [Crescenzi et al., 2001] à moins que $P = NP$. Ce problème reste NP-complet même si deux communications simultanées entre les deux grappes sont autorisées [32].

En fait, résoudre le problème revient à décomposer le graphe biparti en s couplages pondérés où chaque couplage correspond à une phase de communications. Tous les algorithmes d'approximation se basent sur la construction d'une décomposition de couplages dans un graphe biparti inspirée des techniques de coloration des arêtes pour les graphes bipartis [Berge, 1987].

Une approximation de ce problème à un facteur multiplicatif 2 près a été proposée dans [Crescenzi et al., 2001]. De plus, une modification de cette approximation a été apportée et permet de diminuer le rapport d'approximation $2 - \frac{1}{\beta+1}$ dans [Afrati et al., 2005]. Dans [14], nous proposons des algorithmes d'approximation à un facteur $8/3$ de l'optimal. Pour modéliser le goulot d'étranglement, des sommets et des arêtes ont été rajoutés dans le graphe biparti pour que chaque couplage dans ce nouveau graphe ait au plus k arêtes modélisant des communications.

Dans [14], nous avons validé par simulation et expérimentalement avec deux grappes de processeurs que nos algorithmes étaient plus efficaces que plusieurs autres approches

et aussi évalué l'impact du coût de la fragmentation des messages.

Ce travail fut un point de départ de la thèse de Frédéric Wagner qui a étendu ce problème à des scénarios plus complexes de redistributions de données [Wagner, 2005].

Aussi, le problème d'ordonnancement de messages apparaît dans le contexte de la redistribution des données mais aussi dans le contexte de la commutation par paquets pour multiplexage en longueur d'onde (WDM) des réseaux optiques [Choi et al., 1996], [Pieris and G.H., 1994], [Rouskas and Sivaraman, 1996]. Ces problèmes ont été traités comme des problèmes de colorations de chemins dans des graphes où les chemins représentent les communications avec la contrainte qu'une arête peut être traversée que par un nombre limité de chemins de même couleur. Dans les arbres, des algorithmes polynomiaux d'approximation à un facteur multiplicatif 4 ont été décrits dans [Chekuri et al., 2007] et dans [Erlebach et al., 2003].

3.4 Ordonnancement de messages dans les réseaux optiques

Ce travail s'inscrivait dans le projet européen DAVID [project of the 5th PCRD, 2005] du 5ième PCRD (*Data and Voice Integration on DWDM*). Ce projet travaillait autour du développement de méthodes pour la résolution du problème du routage dans des réseaux optiques sans possibilité de stockage. Dans ce projet, l'architecture de réseaux de télécommunications envisagée consiste à connecter des réseaux MAN, ayant la technologie WDM (Wave Division Multiplexing), dont la topologie correspond à une union d'anneaux, par un réseau européen WAN.

Ce travail a été fait en collaboration avec Stéphane Rousseau dont la thèse portait sur l'algorithmique du routage dans de tel réseaux [Rousseau, 2006].

En fait, nous nous sommes penchées sur une topologie bien particulière : une union d'anneaux. De tels réseaux transportent des cellules consécutives de taille identique fixée, tournant continûment sur l'anneau, vides ou contenant une information sous forme d'un paquet de données (on parle d'anneaux *slottés*). Le réseau est synchrone et les paquets optiques circulant sur les réseaux sont de taille fixe. Le temps peut-être discrétisé en étapes (appelées *slots*). Nous nous plaçons dans un réseau de n nœuds. Chaque lien contient $k - 1$ slots et sur chaque nœud est présent un slot. Cela implique qu'au plus $n \times k$ paquets optiques peuvent être en même temps sur le réseau. Un paquet optique envoyé au nœud i à la date t arrivera au nœud $i + 1$ à la date $t + k$. Le paquet optique est caractérisé par son nœud d'origine, son nœud de destination, et sa date d'envoi.

Le fonctionnement de ce réseau ressemble au problème d'ordonnancement que l'on appelle *Pinwheel* [Chen and Mok, 2004] pour l'accès au médium de communication pour les satellites dans le contexte du temps réel avec des émissions des messages de façon périodique.

À chaque étape de communication, une fois envoyés sur le réseau, les paquets ne peuvent pas être stockés par un nœud intermédiaire et ils sont acheminés jusqu'à atteindre leur destination. Cette contrainte caractérise les réseaux tout-optiques.

Un message est donc constitué de plusieurs paquets. La difficulté est d'assurer une contiguïté des paquets d'un même message sur le réseau pour garantir certaines qualités de service (on parlera de *communications contiguës*).

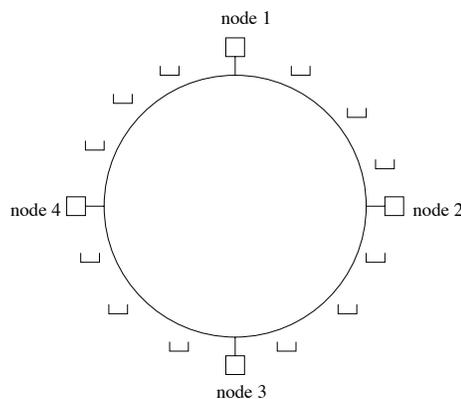


FIG. 3.3 – Réseau composé d’un seul anneau slotté

Une des questions abordées dans le projet DAVID est de comprendre l’impact de cette contrainte dans le réseau. Une première question est de comprendre comment insérer des messages en respectant ces messages. Nous avons partiellement répondu à cette question [7].

Une seconde question porte sur la gestion et l’utilisation des ressources liées à des problèmes d’optimisation. En fonction d’un ensemble de messages (caractérisés par leur source, leur destination, et leur date de création), la question est de construire un algorithme d’accès au médium qui permette d’assurer une certaine qualité de service (minimisation du temps d’attente, du délai moyen, du temps d’arrivée de tous les messages, maximisation du taux d’occupation). Les problèmes de décision associés sont tous NP-complets et des heuristiques ont été développées [20, 21].

Nous sommes concentrées plus particulièrement sur la date de fin d’acheminement de tous les messages.

Lorsque la taille des messages est identique, minimiser le temps de fin d’arrivée de tous les paquets est NP-complet lorsque la topologie est un anneau ou une chaîne. Dans tous ces cas il existe une approximation glouton à un facteur multiplicatif 2 près : dès que un noeud peut insérer un message, il l’insère. Cette approximation a l’avantage qu’elle est répartie puisque toutes les décisions sont locales et que ses performances ne se dégradent pas si les paquets arrivent à la volée au fur et à mesure [21]. Il est à noter que lorsque la taille des messages varie, cet algorithme devient une approximation à un facteur $\ell + 1$ près avec ℓ la taille maximale des messages.

Par contre, le problème devient polynomial lorsque tous les messages dans le réseau arrivent en même temps et que la topologie est une chaîne. Il suffit simplement de considérer que le problème équivaut à une coloration de sommets d’un graphe d’intervalle en considérant les sommets dans l’ordre *leo* [21].

Ceci se rapproche des problèmes d’ordonnancement que l’on appelle *Flowshop* (voir [Brucker, 2001]) où toutes les tâches doivent être exécutées sur plusieurs machines dans le même ordre. Mais, dans le problème Flowshop, il n’existe en général aucune contrainte sur le temps d’attente entre les machines. Ici, dans notre contexte, il y a de telles contraintes.

Dans la continuité de ce travail, en collaboration avec Henry Amet, Freddy Deffner, Marie-Claude Portmann, Stéphane Rousseau, nous avons travaillé sur la création d’heuristiques centralisées de ce problème [33].

Chapitre 4

Coloration de graphes

Dans ce chapitre, nous discutons des différentes colorations de graphes et des problèmes de complexité associés. Nous nous focalisons essentiellement sur la b -coloration et la f -coloration, et aux nombres chromatiques associés.

La première section correspond à une introduction aux problèmes de coloration, en rappelant ce qui est connu sur les colorations classiques. Nous présentons dans la deuxième section la a -coloration et le théorème d'interpolation associé. Nous présentons alors dans la troisième section la b -coloration, et quelques résultats sur la structure des b -spectres. Dans la quatrième section, nous présentons la f -coloration et un certain nombre de résultats négatifs.

Je présente au travers de ces discussions quelques résultats personnels, essentiellement à propos de la b -coloration et de la f -coloration. Nous avons ainsi prouvé d'une part que pour tout ensemble d'entiers, il existe un graphe de b -spectre ou f -spectre associé. Nous avons d'autre part caractérisé la complexité des problèmes de décision et d'approximation associés, en prouvant par exemple la non-approximabilité des nombres chromatiques correspondants.

Les travaux présentés dans ce chapitre ont été obtenus en collaboration avec l'étudiant en thèse Taoufik Faik.

4.1 Introduction

Des applications comme l'ordonnancement de tâches, l'allocation de fréquences, l'allocation de registres, de répartition de services dans les réseaux peuvent être modélisées comme un problème de coloration de graphes. Une *coloration propre* d'un graphe G est une fonction associant à tout sommet V du graphe G un entier représentant une couleur dans $[1, \dots, |V|]$, telle que deux sommets adjacents n'aient pas la même couleur.

Le problème d'optimisation classique est de trouver une coloration avec un nombre de couleurs minimum. Le *nombre chromatique* correspond à ce nombre minimal.

Définition 13 (Nombre chromatique) *Le nombre chromatique d'un graphe G (noté $\chi(G)$) est le nombre minimum de couleurs nécessaires pour lui donner une coloration propre.*

Le problème de décision lié à ce nombre (déterminer si $\chi(G) \leq k$) est connu pour être NP-complet [Garey and Johnson, 1979]. Malheureusement, pour un graphe général,

le nombre chromatique $\chi(G)$ ne peut pas être approximé avec un facteur multiplicatif $|V|^{1-\varepsilon}$ pour tout $\varepsilon > 0$ [Zuckerman, 2007] sauf si $P = NP$. L'article [Zuckerman, 2007] est une extension des résultats de [Feige and Kilian, 1998] et de [Håstad, 1997].

On peut considérer un algorithme glouton qui affecte une couleur d'indice la plus petite possible à un sommet v en fonction des voisins déjà coloriés suivant l'ordre donné par une liste des sommets de G arbitraire. En fait, cet algorithme retourne une coloration ayant un nombre de couleurs au plus $\Delta(G) + 1$. Il est à noter que cet algorithme donne une coloration optimale pour les graphes d'intervalle si la liste des sommets est ordonnée en fonction de l'ordre *reo* ou *leo*.

De plus une borne classique sur le nombre chromatique est la suivante [Brooks, 1941].

Théorème 12 ([Brooks, 1941]) *Pour tout graphe G de degré maximum $\Delta(G)$, on a $\chi(G) \leq \Delta(G) + 1$.*

A partir de ce résultat, de nombreux algorithmes d'approximation ont été développés (voir [Paschos, 2003] pour un survol des résultats connus pour ce problème). Mais, il faut souligner que le fait de connaître des propriétés supplémentaires sur le graphe permet d'améliorer le facteur d'approximation. En effet, par exemple, le problème de trouver le nombre chromatique est approximable à un facteur multiplicatif $O(|V|^{3/14} \log^{O(1)} |V|)$ si le graphe possède une coloration propre de 3 couleurs (c'est-à-dire un graphe 3-coloriable) [Blum, 1994].

Suite à l'étude de cette coloration, d'autres colorations ont été introduites comme la a -coloration, la b -coloration, la f -coloration, . . .

4.2 A propos de la a -coloration.

Définition 14 *Une a -coloration est une coloration propre de sommets telle que chaque paire de couleurs possède des sommets adjacents.*

Le nombre a -chromatique d'un graphe G noté $\Psi(G)$ est le nombre maximum de couleurs pour lequel G possède une a -coloration. Par exemple dans un graphe biparti complet moins un couplage parfait $K'_{n,n}$, le nombre a -chromatique est n c'est-à-dire la taille de la bipartition (voir la figure 4.2 pour visualiser la a -coloration ayant un nombre de couleurs maximum).

Le problème de déterminer si $\Psi(G) \geq k$ est NP-complet [Yannakakis and Gavril, 1980], même pour les arbres [Cairnie and Keith, 1997], les graphes d'intervalles et les co-graphes [Bodlaender, 1989]. Suite à ces résultats de complexité, des algorithmes d'approximation ont été proposés : par exemple une approximation à facteur 2 pour les arbres [Krysta and Loryś, 2006], et une approximation à facteur $O(n \log \log n / \log n)$ pour les graphes en général [Kortsarz and Krauthgamer, 2001].

Des propriétés sur le nombre a -chromatique ont été étudiées, notamment :

Théorème 13 (Homomorphism Interpolation Theorem [Harary et al., 1967]) *Pour tout graphe G , et pour tout entier k tel que $\chi(G) \leq k \leq \Psi(G)$, il existe une a -coloration de k couleurs.*

Cette notion d'interpolation a été étudiée pour différents paramètres (voir [Harary, 1983] pour un récapitulatif).

A partir de cette coloration, et du précédent théorème, d'autres types de coloration ont été introduites comme la b -coloration [Irving and Manlove, 1999], puis la f -coloration [Dunbar et al., 2000]. Les travaux sur des colorations de sommets non-classiques ont été initiés par la constatation de [Kratochvíl et al., 2002] qu'une coloration propre de sommets à c couleurs d'un graphe peut se transformer en une coloration utilisant $c+1$ couleurs : il suffit simplement de colorier un sommet ayant une couleur en une autre couleur distincte. Par contre dès que l'on impose une propriété supplémentaire sur la coloration, il est difficile de transformer une coloration de c couleurs en une autre de $c+1$ couleurs en conservant la propriété, comme nous allons le voir.

4.3 A propos de la b -coloration.

Définition 15 *Une b -coloration est une coloration propre de sommets telle que pour chaque couleur, il existe un sommet de cette couleur qui est voisin de sommets de toutes les autres couleurs.*

Tout graphe possède une b -coloration : toute coloration ayant un nombre minimal de couleurs est une b -coloration. En effet, considérons une coloration ayant un nombre minimal de couleurs. Pour chaque couleur, si aucun de ses sommets est le voisin de sommets de toutes les autres couleurs, alors il suffirait de le colorier dans une couleur non présente dans son voisinage.

Pour une couleur c , le sommet v est un *sommet b -chromatique* pour la couleur c s'il est de couleur c et qu'il est le voisin de sommets de toutes les autres couleurs. On appelle nombre b -chromatique le nombre maximum de couleurs pour lequel un graphe admet une b -coloration.

Le concept de b -coloration a été introduit dans [Irving and Manlove, 1999]. Déterminer le nombre $\phi_b(G)$ est NP-complet en général [Irving and Manlove, 1999] même si le graphe est biparti [Kratochvíl et al., 2002]. De plus, dans [Corteel et al., 2005], il a été prouvé que si $P \neq NP$, il n'existe pas de constante $\epsilon > 0$ pour laquelle calculer $\phi_b(G)$ peut être approximé à un facteur multiplicatif de $120/133 - \epsilon$ en temps polynomial. De plus, [Kouider and Zaker, 2006] ont donné des bornes de $\phi_b(G)$ en fonction du nombre de cliques.

Par contre, ce problème devient polynomial pour les arbres [Irving and Manlove, 1999], pour les co-graphes [Bonomo et al., 2009], et pour certains graphes de Kneser [Javadi and Omoomi, 2009]. Par contre, il est surprenant que cette question reste encore ouverte pour les graphes d'intervalle, et pour les graphes triangulés.

Contrairement, à la coloration classique, la b -coloration de c couleurs d'un graphe n'implique pas une de $c+1$ couleurs. En effet, l'hypercube de dimension 3 illustre cette remarque (voir figure 4.1). Il admet deux b -colorations : une de 2 couleurs et une de 4 couleurs puisqu'il est aussi un graphe biparti complet de 8 sommets privé d'un couplage parfait. Par contre, il est facile de vérifier qu'il n'admet pas de b -coloration de trois couleurs.

Ce simple exemple n'est pas une exception. En effet, nous nous sommes intéressées à la notion de b -spectre d'un graphe dans [6].



FIG. 4.1 – hypercube et ses deux b -colorations

Définition 16 *Un b -spectre d'un graphe G correspond à l'ensemble des entiers k pour lesquels G admet une b -coloration avec k couleurs.*

Dans [6], nous avons prouvé que tout ensemble I correspond à un graphe G ayant I comme b -spectre. Pour cela, pour tout ensemble $I \subset (\mathbb{N} \setminus \{0, 1\})$ avec $\min(I) = 2$, un graphe G a été construit dont le b -spectre $S_b(G)$ est I . Cette construction du graphe G se base principalement sur les graphes bipartis complets moins un couplage parfait. En effet, nous considérons trois cas, en fonction du nombre d'éléments de I .

Cas 1 : Pour $I = \{2\}$, il suffit de prendre G qui correspond à une seule arête ou un graphe biparti complet.

Cas 2 : Pour $I = \{2, n\}$, le graphe biparti complet moins un couplage parfait $K'_{n,n}$ a deux b -colorations de couleurs 2 et n (voir figure 4.2).

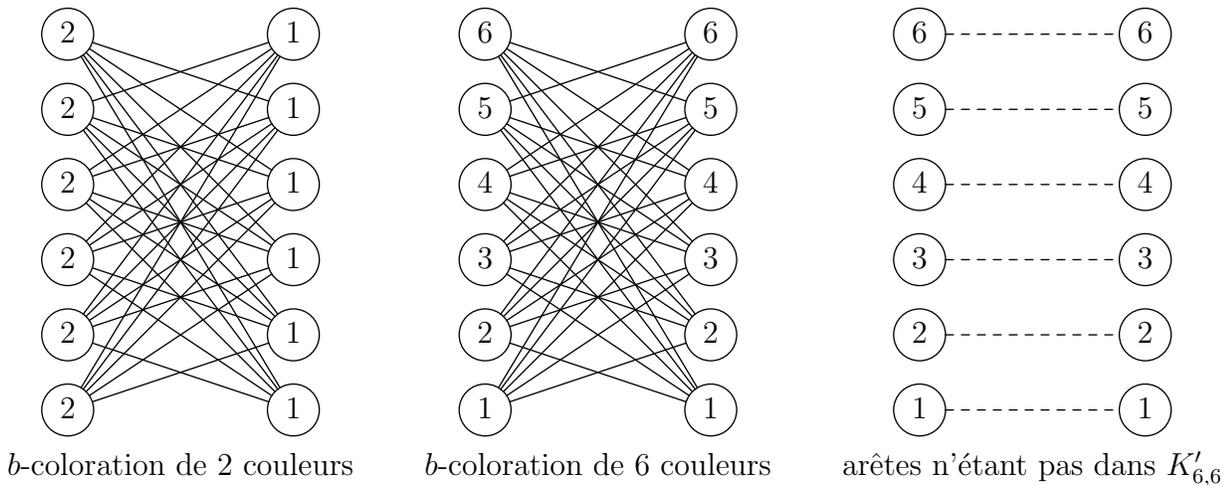


FIG. 4.2 – le graphe $K'_{6,6}$ et ses deux b -colorations

Cas 3 : Pour $I = \{2, n_1, \dots, n_p\}$, avec $2 < n_1 < \dots < n_p$ et $p \geq 2$, le graphe ayant I comme b -spectre est un graphe biparti avec n_p ensembles stables de sommets défini comme suit $G = (\cup_{i=0}^p V_i, \cup_{i=1}^p E_i)$ avec (voir la figure 4.3) :

1. $V_0 = \{v_0^1, \dots, v_0^{n_p}\}, V_p = \{v_p^1, \dots, v_p^{n_p}\},$
2. $\forall i \in \{1, \dots, p-1\}, V_i = \{v_i^1, \dots, v_i^{n_i-1}\},$
3. $\forall \ell, j, \text{ avec } 1 \leq j \leq n_p \text{ et } 1 \leq \ell \leq n_p, [v_0^\ell, v_p^j] \in E_p \Leftrightarrow (\ell \neq j)$

4. $\forall i \in \{1, \dots, p-1\}, \forall \ell, j$, avec $2 \leq j \leq n_i - 1$ et $1 \leq \ell \leq n_p$, $[v_0^\ell, v_i^j] \in E_i \Leftrightarrow (\ell \neq j)$
5. $\forall i \in \{1, \dots, p-1\}, \forall \ell$, $[v_0^\ell, v_i^1] \in E_i \Leftrightarrow (2 \leq \ell \leq n_i - 1)$

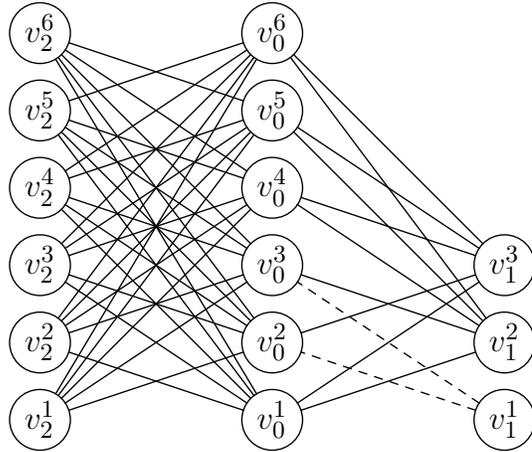


FIG. 4.3 – Un graphe avec le b -spectre $\{2, 4, 6\}$

Pour construire une b -coloration ayant n_k couleurs, il faut tout d'abord donner la couleur i pour tous les sommets $v_{n_k}^i$ et v_0^i , pour $i \in [1, \dots, n_k - 1]$. Ensuite, les sommets v_0^i , ont la couleur 1 pour $i \in [n_k, \dots, n_p]$ et tous les autres sommets sont de couleur n_k . La figure 4.4 montre les différentes b -colorations du graphe aussi donné.

L'idée principale de cette construction est de rendre b -chromatique tous les sommets de V_0 . Pour réaliser cela, et notamment pour des b -colorations ayant un nombre inférieur à n_k , l'astuce est que les sommets $\{v_1^1, \dots, v_{n-k-1}^1\}$ ne sont pas adjacents à tous les sommets de V_0 afin que plusieurs sommets de V_0 auront la possibilité d'avoir la même couleur.

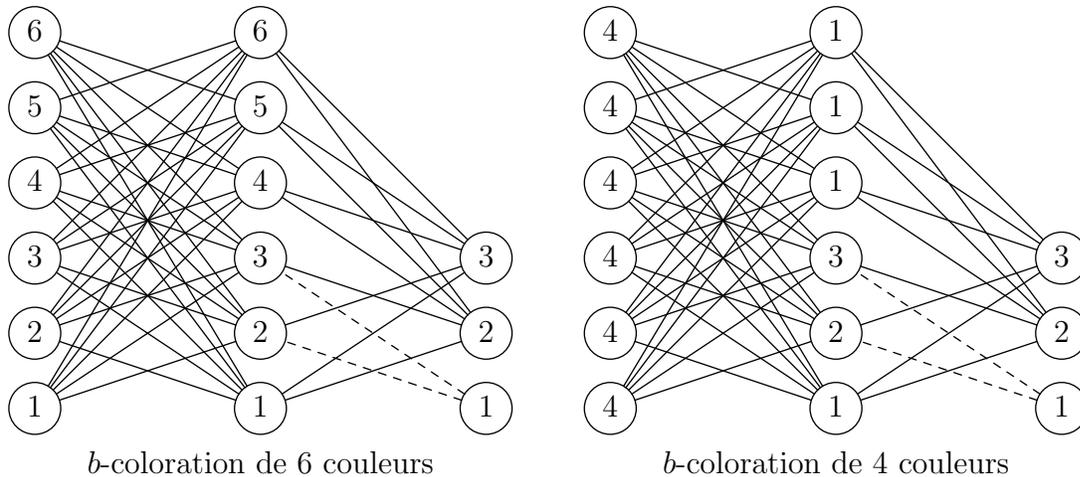


FIG. 4.4 – Un graphe avec le b -spectre $\{2, 4, 6\}$ et ses b -colorations

Cette construction fournit un graphe ayant I comme spectre pour tout ensemble d'entiers $I \subset (\mathbb{N} \setminus \{0, 1\})$ avec $argmin(I) = 2$. Pour enlever la contrainte $I \subset (\mathbb{N} \setminus \{0, 1\})$ avec $min(I) = 2$, il suffit de réaliser une opération de translation : à partir d'un graphe H , on obtient un graphe G correspondant à l'union de ce graphe et d'un graphe K_x complet

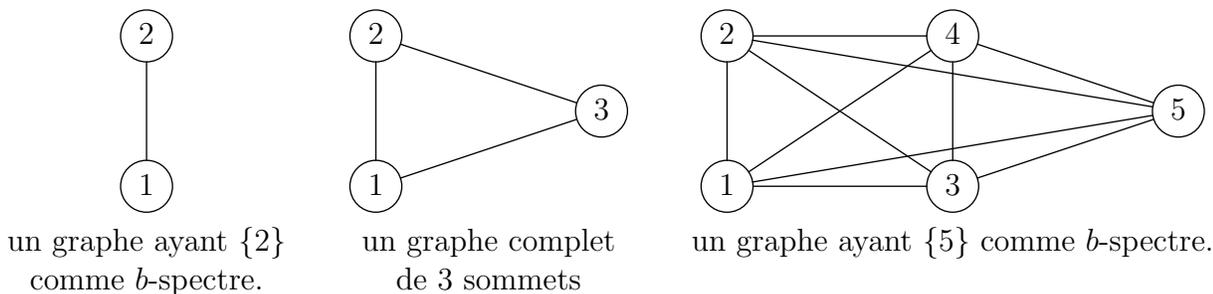


FIG. 4.5 – opération de translation

à x sommets. Tous les sommets de H seront adjacents à tous les sommets de K_x . Cette construction implique qu'il existe une b -coloration de i couleurs dans H si et seulement s'il existe une b -coloration de $i+x$ couleurs dans G . La figure 4.5 illustre cette translation.

Par contre, cette construction ne donne pas un graphe G ayant un nombre minimum de sommets ou d'arêtes ayant I comme b -spectre. Par exemple, pour le graphe dessiné dans la figure 4.3, il suffit de supprimer l'ensemble des arêtes $\{(v_0^j, v_1^i) : 2 \leq i \leq 3 \wedge 4 \leq j \leq 6\}$ pour obtenir un graphe ayant $\{2, 4, 5, 6\}$ comme spectre. Par contre, pour obtenir un graphe avec un tel spectre, notre construction doit rajouter des sommets des arêtes et des sommets. Une question naturelle se pose alors : quels sont les graphes ayant un nombre minimum de sommets ou d'arêtes pour un spectre donné ?

De plus, une question naturelle qui suit à ces précédents travaux est de comprendre la structure des graphes qui ont un intervalle comme b -spectre. Ces problèmes sont liés aux problèmes d'interpolation de [Harary et al., 1967] pour les a -colorations. Plus formellement, quel est le type de graphe dont le b -spectre est un intervalle ?

Définition 17 (Graphe b -continu) *Un graphe est b -continu si et seulement si son b -spectre est composé uniquement d'entiers consécutifs.*

Déterminer si G est b -continu est NP-complet [6]. Ce problème reste aussi difficile même si une b -coloration de $\chi(G)$ couleurs et une b -coloration de $b(G)$ couleurs de G sont données. De plus, ce résultat a été étendu à la classe de graphes bipartis [Faik, 2005].

Heureusement, les arbres [Irving and Manlove, 1999], comme presque tous les graphes 3-réguliers [Irving and Manlove, 1999], ainsi que les graphes plus atypiques P_4 -*sparces* [Bonomo et al., 2009] sont b -continus. Mais aussi, les graphes d'intervalles et les graphes triangulés sont b -continus (voir [36], [37]). La preuve [36] se base sur la structure même de ces graphes (représentation graphique en intervalle, ou ordre d'élimination parfait). A partir d'une b -coloration c de p couleurs, une b -coloration c de $p-1$ couleurs est construite en utilisant la numérotation *reo* pour les graphes d'intervalle (ou un ordre d'élimination parfait pour les graphes triangulés) : par itération successive, l'algorithme supprime un sommet b -chromatique de plus petit indice, jusqu'à qu'une couleur ne possède plus de sommet b -chromatique.

Cette preuve est une preuve existentielle. Elle ne calcule pas des b -colorations de $\phi_b(G)$, $\phi_b(G) - 1, \dots$ couleurs puisqu'à aucun moment, on a construit une b -coloration de $\phi_b(G)$ couleurs.

Dans les graphes d'intervalles et triangulés, la complexité du problème de déterminer le nombre b -chromatique reste encore ouverte. Nous conjecturons que ces problèmes sont NP-complets.

4.4 A propos de la f -coloration

Dans cette section, nous nous intéresserons à une b -coloration particulière : la f -coloration introduit dans [Dunbar et al., 2000]. Voici la définition formelle de cette coloration :

Définition 18 (f -coloration) *Une f -coloration d'un graphe G est une coloration propre de sommets telle que tout sommet du graphe est b -chromatique.*

Contrairement à la coloration propre, et à la b -coloration, il existe des graphes n'admettant pas de f -coloration. Par exemple, le graphe composé d'un graphe complet de $k \geq 3$ sommets et d'un sommet étant voisin d'un seul sommet de ce graphe complet n'admet pas de f -coloration puisque le nombre chromatique de ce graphe est supérieur à 3 et que le sommet de degré 1 ne peut pas être b -chromatique.

De plus, on peut remarquer que tout graphe biparti admet une f -coloration, puisque une coloration propre de deux couleurs est une f -coloration. Cela implique, que l'algorithme polynomial reconnaissant un graphe biparti permet de déterminer un temps polynomial si un graphe admet une f -coloration de 2 couleurs. A-t-on le même résultat pour 3 couleurs ?

Ce n'est pas vrai pour 3 couleurs. Décider si un graphe admet une f -coloration avec 3 couleurs est NP-complet [Dunbar et al., 2000]. Par rapport à la b -coloration, déterminer s'il existe d'une b -coloration pour tout graphe est polynomial. En effet, il suffit de rappeler qu'une coloration d'un graphe G utilisant $\chi(G)$ couleurs est toujours une b -coloration.

Face à ce résultat, la frontière entre la tractabilité et la non-tractabilité a été étudiée. Il a été prouvé que décider si un graphe admet une f -coloration ayant 3 couleurs est NP-complet pour les graphes en général [Dunbar et al., 2000] et aussi pour les graphes bipartis [34], [Laskar and Lyle, 2009]. Ce résultat a été étendu pour des graphes de degré maximum inférieur ou égale à 3 [41].

De plus, peu de classes de graphes ont été étudiées dans ce contexte. Un exemple parmi les plus simples est l'arbre. Comme les arbres possèdent des sommets de degré 1 (les feuilles) et qu'ils sont bipartis, il est facile de déduire que les arbres possèdent une seule f -coloration de 2 couleurs. Récemment, dans [Laskar and Lyle, 2009], la préservation des f -colorations de produits cartésiens a été étudiée. Ceci a permis de déduire que les graphes hypercubes Q_n de dimension particulière $n = 2^k - 1$ possèdent des f -colorations.

A notre connaissance, pour les classes de graphes classiques (graphes d'intervalle, graphes triangulés, ...), les questions suivantes sont encore ouvertes. Peut-on déterminer si le graphe possède une f -coloration en temps polynomial ? Existe-t-il un algorithme qui construit une f -coloration si le graphe en entrée en possède une ?

Malgré ces résultats négatifs, comme pour la coloration propre, les paramètres classiques étudiés pour la coloration s'adaptent pour la f -coloration. En d'autres termes, si une telle coloration existe pour un graphe G , le *nombre f -chromatique* (resp. *nombre f -achromatique*) noté (resp. $\psi_f(G)$) est le nombre de couleurs minimum (resp. maximum) pour lequel G admet une f -coloration.

Dans [34], il a été prouvé que même pour des graphes ayant une f -coloration, le problème de calculer le paramètre $\psi_f(G)$ ne peut pas s'approximer à un facteur multiplicatif $n^{1-\epsilon}$ pour n'importe $\epsilon > 0$ avec n correspondant au nombre de sommets de G , si $P \neq NP$. Contrairement à la coloration propre (voir résultat [Blum, 1994]), connaître l'existence d'une f -coloration ne simplifie pas le problème.

Par contre, à ma connaissance, aucun travail ne s'est penché sur les paramètres $\psi_f(G)$ et $\chi_f(G)$ dans les graphes classiques (d'intervalles, triangulés, co-graphes, ...). Il serait intéressant de résoudre les questions suivantes pour ces types de graphes. Peut-on calculer en temps polynomial les paramètres $\psi_f(G)$ et $\chi_f(G)$ dans de tels types de graphes si le graphe possède une f -coloration ? Peut-on construire polynomialement une f -coloration ayant $\psi_f(G)$ (resp. $\chi_f(G)$) couleurs ?

De la même façon que pour la b -coloration, les notions de f -spectre et de graphes f -continus ont été introduites. Il a été prouvé dans [34] que décider si un graphe est f -continu est NP-complet. De plus si $P = NP$, le problème de calculer le paramètre $\psi_f(G)$ pour un graphe f -continu n'appartient pas à la classe APX même si la propriété de f -continuité est connue.

Malgré tous les résultats négatifs, il a été prouvé dans [34] que pour tout ensemble fini et non vide $I \subset \mathbb{N} \setminus \{0, 1\}$, il existe un graphe G dont le f -spectre est l'ensemble I . Cette construction s'inspire de celle sur les b -colorations mais est plus complexe. En effet, l'opérateur de translation existe toujours et le graphe se construit pas à pas. A chaque étape, le graphe construit s'enrichit d'une f -coloration supplémentaire tout en conservant ses précédentes f -colorations par l'intermédiaire de produits cartésiens de graphes complets. Cette succession d'opérations provoque le fait que le nombre de sommets du graphe ainsi construit est exponentiel par rapport à la cardinalité de I et à son maximum.

Suite à cette remarque, une question naturelle se pose : pour un ensemble d'entiers I , existe-t-il un graphe G ayant I comme f -spectre et ayant un nombre polynomial de sommets par rapport à I ?

Chapitre 5

Jeux classiques en théorie algorithmique des jeux

Ce chapitre vise à présenter et rappeler quelques jeux classiques de la théorie algorithmique des jeux tout en introduisant les concepts de base de la théorie des jeux. Ce chapitre n'est toutefois pas une liste exhaustive des jeux étudiés en informatique. Afin d'obtenir plus d'information, le lecteur peut se référer au livre [Nisan et al., 2007].

Le chapitre s'organise de la façon suivante. Dans la section 5.1, nous allons introduire la notion d'équilibre de Nash (mixte) en discutant la complexité du problème de leur calcul. Ensuite, nous allons instancier sur des jeux simples, comme les jeux symétriques, et puis sur les jeux de potentiel ordinal (avec leurs différentes variantes). Finalement, comme ces classes de jeux possèdent plusieurs équilibres, nous allons discuter quels sont les différents critères pour les évaluer.

Ce chapitre est essentiellement un rapide survol. Nous présentons cependant une contribution personnelle (obtenue en collaboration avec Fanny Pascual) concernant l'existence d'équilibres de Nash purs et d' ϵ -équilibres de Nash purs dans les jeux d'allocation de tâches, lorsqu'un joueur est responsable de plusieurs tâches [43].

5.1 Jeux statiques (non-répétés)

Cette section se consacre aux jeux où n joueurs sont en concurrence. Dans le chapitre 2, section 2.5, les notions de base sur la théorie des jeux ont été introduites comme la notion de joueur, de stratégie pure. Nous allons aussi fixer les notations que l'on conservera dans la suite du document.

L'ensemble des joueurs sera noté par la suite $[n] = \{1, \dots, n\}$. Chaque joueur i a un ensemble \mathcal{S}_i de stratégies pures. Soit m_i la cardinalité de \mathcal{S}_i .

Définition 19 (stratégie mixte) Une stratégie mixte $q_i = (q_{i,1}, q_{i,2}, \dots, q_{i,m_i})$ correspond à un vecteur de probabilité sur les stratégies pures : la stratégie pure l est choisie avec la probabilité $q_{i,l} \in [0, 1]$, avec $\sum_{\ell=1}^{m_i} q_{i,\ell} = 1$.

L'espace des stratégies mixtes du joueur i correspond à un simplexe que nous noterons \mathcal{K}_i . Soit $\mathcal{K} = \prod_{i=1}^n \mathcal{K}_i$ l'espace des stratégies mixtes.

Observons que n'importe quelle stratégie pure l peut être considérée comme une stratégie mixte e_l telle que le vecteur e_l est celui constitué uniquement de zéro à l'ex-

ception de son ℓ ème élément étant à 1. Cela signifie aussi que e_ℓ est un coin du simplexe \mathcal{K}_i .

Définition 20 (profil de stratégies mixtes) Un profil de stratégies mixtes $Q = (q_1, \dots, q_n)$ est un élément de \mathcal{K} qui spécifie les stratégies (pure ou mixtes) de tous les joueurs : le vecteur de probabilité q_i correspond à la stratégie mixte utilisée par le joueur i .

En suivant les conventions classiques de la théorie des jeux, de façon abusive, nous écrirons Q_{-i} , le profil Q privé de la stratégie du joueur i : $Q_{-i} = (q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_n)$ correspond au vecteur de stratégies de tous les joueurs sauf du joueur i .

Les jeux peuvent avoir des coûts aléatoires : sachant un profil des stratégies $Q \in \mathcal{K}$, le coût de chaque joueur i est la valeur espérée du coût. En particulier, le coût espéré par le joueur i jouant la stratégie pure e_ℓ est notée $c_i(e_\ell, Q_{-i})$.

5.1.1 Meilleure réponse

La stratégie x du joueur i par rapport au profil Q est une *meilleure réponse*, si $c_i(x, Q_{-i}) \leq c_i(q_i, Q_{-i})$ pour tout $q_i \in \mathcal{K}_i$. L'ensemble (compact, convexe, non vide) de toutes les meilleures réponses du joueur i par rapport à Q_{-i} (ensemble des stratégies des autres joueurs) est notée $BR_i(Q_{-i})$.

Par exemple, considérons le jeu dont les coûts sont représentés par la table 5.1 (c'est le même que celui pris en exemple dans le chapitre 2, pages 13 et 19).

		joueur 2	
		s_1	s_2
joueur 1	s_1	2	0
	s_2	1	1
		Coût du joueur 1	

		joueur 1	
		s_1	s_2
joueur 2	s_1	2	0
	s_2	1	1
		Coût du joueur 2	

TAB. 5.1 – Coûts du jeu décrit à la page 19

Sans perte de généralité, supposons que p soit un réel compris entre 0 et 1 et que le joueur 2 joue la stratégie mixte $(1 - p, p)$, c'est-à-dire qu'il jouera la stratégie s_1 avec la probabilité $1 - p$, et la stratégie s_2 avec la probabilité p . Alors pour le joueur 1, le coût espéré du joueur 1 est égal à 1 s'il joue la stratégie pure s_2 c'est-à-dire $c_1(s_2, (1 - p, p)) = 1$. Sinon il est à $2 - 2p$. Alors, si $\frac{1}{2} < p$, il est préférable qu'il joue la stratégie pure s_1 plutôt que s_2 . Que se passe-t-il lorsque $p = 1/2$? Les deux stratégies sont des meilleures réponses. De plus, toute stratégie mixte est une meilleure réponse par rapport la stratégie mixte $(\frac{1}{2}, \frac{1}{2})$ du joueur 2. Récapitulons

$$BR_1((1 - p, p)) = \begin{cases} (1, 0) & \text{si } 1 \geq p > \frac{1}{2} \\ \mathcal{K}_1 & \text{si } p = \frac{1}{2} \\ (0, 1) & \text{si } \frac{1}{2} > p \geq 0 \end{cases}$$

Une stratégie mixte est caractérisée par l'ensemble des stratégies pures à laquelle elle attribue une probabilité strictement positive. Cet ensemble s'appelle le *support*. Par exemple, toutes les stratégies pures sont des stratégies mixtes de support réduit à un singleton. De plus,

Propriété 1 *Une stratégie mixte est une meilleure réponse par rapport à un ensemble de stratégies Q si et seulement si toutes les stratégies de son support sont aussi des meilleures réponses par rapport à Q .*

En fait, considérons une stratégie mixte ayant un support contenant au moins deux éléments ℓ et t . Si une stratégie pure ℓ de son support a un coût plus élevé que celle t , alors il est préférable pour le joueur de choisir t à la place de ℓ lorsque la stratégie ℓ est choisie. Donc ainsi est construite une stratégie mixte ayant un coût plus petit que celle de départ.

La première étape pour modéliser un système en concurrence peut être de l'exprimer sous forme de jeu. L'étape suivante est de prédire dans quel état stable il va être.

John Nash prouvait dans [Nash, 1950] que tout jeu possède une situation d'équilibre mixte, que l'on nomme équilibre de Nash dans laquelle aucun joueur n'a intérêt unilatéralement de s'écarter. Ce concept d'équilibre de Nash pour les jeux a été un point important de la théorie de jeux.

5.1.2 Équilibre de Nash

Définition 21 ([Nash, 1950]) *Un équilibre de Nash mixte est un profil $Q^* = (q_1^*, \dots, q_n^*)$ tel que*

$$\forall i, \forall q'_i \in \mathcal{K}_i \text{ on a } c_i(q_i^*, Q_{-i}^*) \leq c_i(q'_i, Q_{-i}^*).$$

Rappelons que \mathcal{K}_i correspond à l'ensemble des stratégies mixtes du joueur i .

La notion d'équilibre de Nash peut s'écrire en utilisant la notion de meilleure réponse. En effet, un profil $Q^* = (q_1^*, \dots, q_n^*)$ est un équilibre de Nash mixte, si pour tout joueur i , on a $q_i^* \in BR_i(Q_{-i}^*)$. A partir du calcul de l'ensemble des meilleures réponses, il est facile de calculer quels sont les équilibres de Nash pour le jeu dont la forme stratégique correspond à la table 5.1 : deux équilibres de Nash purs (s_1, s_2) , (s_2, s_1) , et un équilibre de Nash pur (q, q) avec q correspondant à la stratégie mixte $(\frac{1}{2}, \frac{1}{2})$.

Théorème 14 ([Nash, 1950]) *Il existe un équilibre de Nash mixte dans tout jeu fini.*

Par contre, certains jeux ne possèdent pas toujours un équilibre de Nash pur : voir l'exemple 2.2 décrit à la page 14. Et même pour les jeux décrits dans le chapitre 2, décider s'il existe un équilibre de Nash est NP-complet.

Pour les équilibres mixtes, il en existe toujours un. Une question naturelle se pose. Un équilibre de Nash, est-il toujours calculable ? Les équilibres de Nash ont été introduits par l'idée de prédire vers quels états le système va converger si les acteurs sont rationnels. Discuter la complexité du calcul des équilibres de Nash est lié la pertinence de cette notion : on peut se dire que s'ils sont difficiles à calculer, il n'est pas clair que le système va nécessairement tendre vers ces états. Par exemple, dans les différents jeux, les joueurs peuvent être des machines/routeurs/capteurs/applications, il n'est pas du tout évident que de tels systèmes vont converger vers de tels équilibres.

La preuve du théorème 14 se base sur le théorème de point fixe de Brouwer qui affirme qu'une fonction continue sur $\mathcal{K} \rightarrow \mathcal{K}$ admet un point fixe. Cette preuve n'est pas constructive puisque trouver un point fixe de Brouwer paraît difficile [Hirsch et al., 1989] (il est au moins nécessaire d'effectuer un nombre exponentiel d'opérations).

Actuellement, l'algorithme de Lemke-Howson [Lemke and Howson, 1964] est l'algorithme combinatoire le plus efficace pour calculer les équilibres de Nash mixtes. L'idée de l'algorithme est de parcourir l'espace des profils en se basant sur les propriétés des meilleures réponses (notamment sur la propriété des supports des équilibres de Nash). A chaque itération, le support de l'état courant est modifié légèrement afin d'accéder vers un état ayant un support qui "respecte le plus" la propriété 1. Remarquons que cet algorithme ressemble beaucoup à la méthode du simplexe de la programmation linéaire puisque la modification du support est basée sur une succession d'applications d'une méthode du pivot pour résoudre des systèmes linéaires.

Du point de vue de la théorie des graphes, l'algorithme de Lemke-Howson parcourt un graphe dont chaque sommet correspond à un profil du jeu et dont chaque sommet a un degré sortant et un degré entrant qui vaut 0 ou 1. En effet, cela correspond au fait que la modification d'un support d'un profil vers un autre est déterministe. De plus, les équilibres de Nash correspondent à des puits (aucune meilleure modification du support est possible).

Remarquons que le nombre de sommets du graphe peut-être exponentiel. Un point crucial de cet algorithme est que connaître le voisinage d'un sommet se réalise en un temps polynomial. L'algorithme construit ainsi le graphe à la volée.

Par contre, il peut explorer un nombre exponentiel de sommets même pour des jeux particuliers comme les jeux symétriques à 2 joueurs [Savani and von Stengel, 2004] définis dans la section suivante.

La classe de complexité *PPAD* (en anglais *Polynomial Parity Arguments on Directed graphs*) a été introduite dans [Papadimitriou, 1994] (voir Annexe A pour une description). Cette classe correspond à construire une solution d'un problème de décision ayant que des instances positives.

Il a été prouvé que,

Théorème 15 ([Chen and Deng, 2005]) *Calculer un équilibre de Nash est PPAD-complet même si le jeu ne possède que deux joueurs.*

Résoudre un problème PPAD-complet semble compliqué : trouver un point fixe de Brouwer est PPAD-complet [Papadimitriou, 1994] et le nombre d'opérations pour trouver un point fixe de Brouwer est minoré par une exponentielle [Hirsch et al., 1989].

Maintenant, nous allons nous concentrer sur les jeux symétriques à deux joueurs.

5.1.3 Jeux à deux joueurs symétriques

Voici la définition formelle d'un jeu symétrique.

Définition 22 (Jeu symétrique) *Un jeu $J = (J, \mathcal{S}, (c_i)_{i \in J})$ est symétrique si les joueurs ont le même ensemble de stratégies pures et la même fonction de coût.*

Tout d'abord, nous intéressons aux jeux symétriques à deux joueurs $\{1, 2\}$ ayant deux stratégies pures $\{\mathbf{s}_1, \mathbf{s}_2\}$. Un exemple de jeu symétrique est celui de la section 2.1 (page 14) avec la table 2.1 de coût (page 19).

Nous allons classer les jeux en suivant [Weibull, 1995] en fonction de leur matrice de coût $A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$. Rappelons ici, que la modélisation de la notion de préférence

correspond à une fonction de coût que les joueurs cherchent à minimiser contrairement à [Weibull, 1995] qui utilise des utilités que les joueurs cherchent à maximiser. Afin de se conformer à cette classification, les conditions d'appartenance à une catégorie sont inversées par rapport à [Weibull, 1995].

La matrice A peut se transformer en la matrice (symétrique) $\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}$ avec $a_1 = a_{1,1} - a_{2,1}$ et $a_2 = a_{2,2} - a_{1,2}$ sans modifier les relations entre les différentes stratégies. En effet, soustraire $a_{2,1}$ à la colonne 1 provoque simplement une translation des coûts des joueurs sans modifier la différence entre les coûts et sans modifier quelle(s) stratégie(s) est la meilleure réponse lorsque l'adversaire joue la première stratégie. Les mêmes arguments s'appliquent pour l'opération de soustraction de $a_{1,2}$ à la colonne 2.

Les jeux peuvent alors se classifier en quatre catégories en fonction du signe de a_1 , et de a_2 :

- Catégorie *I* avec $a_1 > 0$ et $a_2 < 0$: La stratégie \mathbf{s}_2 domine strictement la stratégie \mathbf{s}_1 puisque pour tout choix de l'adversaire, la meilleure réponse est toujours la stratégie \mathbf{s}_2 . Par conséquent, le jeu possède un équilibre pur ayant (s_2, s_2) comme profil.
- Catégorie *II* avec $a_1 < 0$ et $a_2 < 0$: aucune stratégie dominante existe. Pourtant, par calcul d'ensemble des meilleures réponses (calcul analogue à la page 40 pour l'exemple 5.1), le jeu possède deux équilibres de Nash pur évidents $((s_1, s_1)$ et $(s_2, s_2))$ et un équilibre de Nash mixte dont le profil est (q, q) avec $q = \left(\frac{a_2}{a_1+a_2}, \frac{a_1}{a_1+a_2}\right)$.
- Catégorie *III* avec $a_1 > 0$ et $a_2 > 0$: même argument que pour la catégorie *II*. Le jeu possède deux équilibres de Nash pur évidents $((s_1, s_2)$ et $(s_2, s_1))$ et un équilibre de Nash mixte dont le profil est (q, q) .
- Catégorie *IV* avec $a_1 < 0$ et $a_2 > 0$: même argument que pour la catégorie *I*. Le jeu possède un équilibre pur ayant (s_1, s_1) comme profil.

Voici un récapitulatif des équilibres de Nash en fonction des catégories.

	$a_2 \geq 0$	$a_2 \leq 0$
$a_1 \leq 0$	Un équilibre de Nash pur $\{(s_1, s_1)\}$	Deux équilibres de Nash purs $\{(s_1, s_1), (s_2, s_2)\}$ et un équilibre de Nash mixte $\{(q, q)\}$
$a_1 \geq 0$	Deux équilibres de Nash purs $\{(s_1, s_2), (s_2, s_1)\}$ et un équilibre de Nash mixte $\{(q, q)\}$	Un équilibre de Nash pur $\{(s_2, s_2)\}$

TAB. 5.2 – Classification des jeux symétriques à deux joueurs en posant $q = \left(\frac{a_2}{a_1+a_2}, \frac{a_1}{a_1+a_2}\right)$.

Maintenant, nous allons discuter des jeux symétriques à deux joueurs mais en considérant $m \neq 2$ stratégies. Il est difficile de décider l'existence d'un équilibre avec certaines propriétés. En effet,

Théorème 16 ([Gilboa and Zemel, 1989]) *Étant donné un jeu à deux joueurs donné sous forme stratégique, les problèmes suivants : décider s'il existe*

- au moins deux équilibres de Nash ;
- un équilibre de Nash dans lequel un joueur possède un coût inférieur à k ;
- un équilibre de Nash dans lequel le coût total des joueurs est inférieur à k ;

- un équilibre de Nash dans lequel les supports des stratégies sont de taille inférieure à k ;
- un équilibre de Nash dans lequel un joueur a une stratégie mixte ayant un support de taille inférieure à k ;
- un équilibre de Nash dans lequel une stratégie appartient (ou pas) au support ;
- s'il existe un équilibre de Nash Pareto optimal ;

sont des problèmes NP-complets. Ils restent NP-complets même si le jeu est symétrique.

Cette preuve a été simplifiée dans [Conitzer and Sandholm, 2008] en apportant des résultats sur la non approximation :

Théorème 17 ([Conitzer and Sandholm, 2008]) *A moins que $P = NP$, il n'existe pas d'algorithme d'approximation en temps polynomial à un facteur r près qui approxime*

- le minimum des coûts des joueurs d'un équilibre de Nash,
- le coût social (respectivement coût moyen, le coût maximum des joueurs) d'un équilibre de Nash,

avec $r > 1$. Ces résultats restent valides si le jeu est symétrique.

Même compter les équilibres de Nash est $\#$ -complet [Conitzer and Sandholm, 2008].

Maintenant, nous allons décrire des classes de jeux à n joueurs ayant des propriétés sur les fonctions de coûts permettant d'assurer l'existence d'au moins un équilibre de Nash pur.

5.2 Les jeux de potentiel ordinal

Nous allons introduire une classe de jeux nommée les jeux de potentiel ordinal introduit par [Monderer and Shapley, 1996]. A ces jeux est associée une fonction qui permet de garantir l'existence d'équilibres de Nash purs.

Définition 23 (Jeu de potentiel ordinal) *Un jeu est un jeu de potentiel ordinal s'il existe une fonction ϕ définie à partir de l'espace des stratégies pures vers \mathbb{R} telle que pour toutes les stratégies pures Q_{-i} , q_i , et q'_i , on a la condition suivante :*

$$c_i(q_i, Q_{-i}) - c_i(q'_i, Q_{-i}) > 0 \text{ si et seulement si } \phi(q_i, Q_{-i}) - \phi(q'_i, Q_{-i}) > 0$$

La fonction sera aussi appelée fonction de potentiel ordinal.

Par définition d'un jeu de potentiel ordinal, un profil $Q = (q_1, \dots, q_n)$ est un équilibre de Nash pur si et seulement si $\phi(q_i, Q_{-i}) - \phi(q'_i, Q_{-i}) < 0$ pour tout joueur i , pour toute stratégie pure q'_i du joueur i .

Théorème 18 ([Monderer and Shapley, 1996]) *Tout jeu de potentiel ordinal possède un équilibre de Nash.*

En effet, il suffit de considérer une suite de profils purs $Q^0, Q^1, Q^2, \dots, Q^k \dots$ telle que pour tout entier k , parmi tous les joueurs, un seul joueur i a une stratégie différente entre le profil Q^k et Q^{k+1} avec $c_i(Q^k) > c_i(Q^{k+1})$. Par conséquence, $\phi(Q^0) > \phi(Q^1) > \phi(Q^2) > \dots > \phi(Q^k) > \dots$. Comme ϕ est bornée inférieurement, puisque définie sur un espace fini,

cela implique que cette suite est finie et que le dernier élément correspond à un équilibre de Nash.

Une autre classe de jeux est obtenue en ajoutant une contrainte supplémentaire à la fonction de potentiel :

Définition 24 (Jeu de potentiel (exact)) *Un jeu est un jeu de potentiel exact (ou appelé jeu de potentiel) s'il existe une fonction ϕ définie à partir de l'espace des stratégies pures vers \mathbb{R} telle que pour toutes les stratégies pures Q_{-i} , q_i , et q'_i , la fonction ϕ satisfait à la condition suivante :*

$$c_i(q_i, Q_{-i}) - c_i(q'_i, Q_{-i}) = \phi(q_i, Q_{-i}) - \phi(q'_i, Q_{-i}).$$

Un jeu de potentiel (exact) est donc un jeu de potentiel ordinal.

Le reste du chapitre décrit des exemples de ces jeux : les jeux de congestion, les jeux d'ordonnancement, et les jeux d'allocations de tâches. Dans ces jeux, les joueurs sont en concurrence sur un ensemble de ressources noté $[m] = \{1, 2, \dots, m\}$. Chaque ressource $1 \leq r \leq m$ possède une fonction décroissante de coût $C_r : [n] \rightarrow \mathbb{R}$.

5.2.1 Les jeux de congestion

Les jeux de *congestion* ont été définis dans [Rosenthal, 1973] : les ressources $[m]$ correspondent aux arêtes d'un graphe et les joueurs choisissent un sous-ensemble de ces arêtes. L'espace des stratégies pures \mathcal{S}_i du joueur i correspond à l'ensemble de toutes les parties de l'ensemble des ressources $[m]$ et sa cardinalité est égale à $2^{|[m]|}$. Une stratégie pure $q_i \in \mathcal{S}_i$ pour le joueur i est donc un sous-ensemble de $[m]$. Le coût du joueur i pour le profil de stratégies pures Q correspond à $c_i(Q) = \sum_{r \in q_i} C_r(\lambda_r(Q))$ où $\lambda_r(Q)$ est le nombre de q_i avec $r \in q_i$.

Théorème 19 ([Monderer and Shapley, 1996]) *Les jeux de congestion sont des jeux de potentiel exact. Tout jeu de potentiel exact est équivalent à un jeu de congestion.*

Pour prouver la première assertion, il suffit de constater que la fonction

$$\phi(Q) = \sum_{e \in [m]} \sum_{k=1}^{\lambda_e(Q)} C_e(k)$$

est une fonction de potentiel.

On peut constater que tous les équilibres de Nash purs dans les jeux de congestion ne sont pas forcément des minimums locaux de la fonction de potentiel. En effet, il suffit de considérer l'exemple suivant. Le jeu est composé de deux joueurs ayant deux stratégies pures et ayant trois ressources $\{1, 2, 3\}$. Le joueur 1 a deux stratégies pures $\{1\}$ et $\{2\}$ et le joueur 2 a $\{1\}$ et $\{2, 3\}$ comme stratégies pures. Le coût des ressources est le suivant : $C_1(x) = C_2(x) = x$ et $C_3(x) = 2$.

La table 5.3 représente la forme stratégique de ce jeu. Par calcul, il est facile de voir qu'il existe deux équilibres de Nash purs ayant $(\{2\}, \{1\})$ et $(\{1\}, \{2, 3\})$ comme profil et un équilibre mixte tel que le joueur 1 (resp. 2) a $(2/3, 1/3)$ (resp. $(1/2, 1/2)$) comme stratégie mixte. L'équilibre de Nash pur $(\{1\}, \{2, 3\})$ n'est pas un minimum local de ϕ (voir la table 5.4 représentant les valeurs de ϕ).

		joueur 2	
		{1}	{2, 3}
joueur 1	{1}	(2,2)	(1,3)
	{2}	(1,1)	(2,4)

TAB. 5.3 – forme stratégique du jeu

		joueur 2	
		{1}	{2, 3}
joueur 1	{1}	3	4
	{2}	2	4

TAB. 5.4 – Valeur de ϕ du jeu de la table 5.3

Ces jeux ont été étudiés dans le cadre du routage : les joueurs veulent réserver un chemin entre deux sommets du réseau. Dans ce cadre le problème de calculer un équilibre de Nash pur est PLS-complet [Fabrikant et al., 2004]. La classe PLS définie dans [Johnson et al., 1988] est une classe correspondant à trouver des minimums locaux de fonctions sachant qu'il en n'existe un avec des contraintes supplémentaires. Voir l'annexe A pour une définition plus formelle.

Par contre, lorsque les joueurs ont même source et destination, trouver un équilibre de Nash revient à trouver un flot entier maximal de coût minimum dans un graphe. Dans ce cas, déterminer un tel équilibre est polynomial [Fabrikant et al., 2004].

Une extension de ce jeu est de considérer que les joueurs doivent transiter une quantité d'information d'un point à un autre. Alors, un poids w_i est associé à chaque joueur i et dans ce cadre la fonction $\lambda_r(Q)$ est définie par $\lambda_r(Q) = \sum_{j:r \in q_j} w_j$. Ces jeux s'appellent des jeux de *congestion pondérés*.

Théorème 20 ([Fotakis et al., 2005]) *Les jeux de congestion pondérés sont des jeux de potentiel exact si les fonctions de coût sont linéaires.*

Par contre, ce n'est pas le cas si les fonctions de coût ne sont pas linéaires. En effet, certaines instances ne possèdent pas d'équilibres de Nash purs. Considérons l'exemple tiré de [Roughgarden, 2007]. Le réseau est représenté par la figure 5.1 et les poids des arêtes correspondent à leur fonction de coût. Deux joueurs sont en concurrence voulant transiter de l'information du sommet s vers t avec $w_1 = 1$ et $w_2 = 2$. Cette instance ne possède pas d'équilibre de Nash pur. En effet, si le joueur 1 choisit le chemin st , alors la meilleure réponse pour le joueur 2 est le chemin svt . Et de plus si le joueur 2 choisit le chemin svt , alors la meilleure réponse pour le joueur 1 est le chemin $svut$, ainsi de suite.

5.2.2 Les jeux d'ordonnement

Dans les jeux d'*ordonnement* définis dans [Koutsoupias and Papadimitriou, 1999], les ressources sont en fait les machines, et les joueurs correspondent aux tâches. Les joueurs choisissent une seule machine pour exécuter leur tâche (un singleton de $[m]$). Cependant, l'espace des stratégies pures \mathcal{S}_i du joueur i responsable d'une tâche de longueur w_i peut être soit $[m]$ ou soit un sous-ensemble de $[m]$, et une stratégie pure $q_i \in \mathcal{S}_i$ pour le joueur i est un élément $r \in [m]$. On notera le jeu d'ordonnement à *accès restreint* si l'espace des stratégies pures pour chacun des joueurs n'est pas $[m]$.

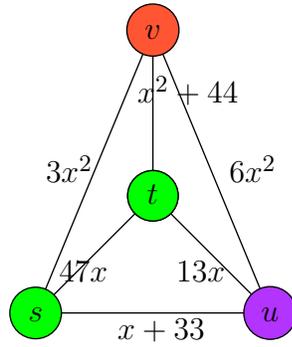


FIG. 5.1 – Jeu de congestion pondéré où il n'existe pas d'équilibre de Nash pur

Le coût pour le joueur (tâche) i pour le profil de stratégies pures (placement de toutes tâches sur les machines) $Q = (q_1, \dots, q_n)$ correspond à $c_i(Q) = C_{q_i}(\lambda_{q_i}(Q))$, où $\lambda_r(Q)$ est la charge de la machine r : $\lambda_r(Q) = \sum_{j:q_j=r} w_j$, c'est-à-dire la somme des longueurs de toutes les tâches placées sur r .

En fait, la fonction de coût d'une ressource permet de modéliser les différentes vitesses d'exécution. Par exemple,

1. Les machines peuvent être identiques : les fonctions de coût sont identiques ;
2. Les machines n'ont pas la même vitesse mais elles sont comparables entre elles : les fonctions de coûts sont linéaires. Ce modèle s'appelle le *modèle uniformément relié*.
3. Les machines ne sont pas comparables. C'est le modèle n'ayant aucune contrainte sur les vitesses des machines. Nous ne discuterons pas de ce modèle mais pour avoir plus d'informations le lecteur pourra se référer aux articles [Immorlica et al., 2009, Dürr and Thang, 2009].

Théorème 21 ([Christodoulou et al., 2004]) *Les jeux d'ordonnancement ayant des machines identiques ou uniformément reliées sont des jeux de potentiel exact.*

Ce qui implique que de tels jeux possèdent des équilibres de Nash pur. De plus, l'algorithme d'ordonnancement de liste¹ utilisant l'ordre des tâches de longueurs décroissantes (LPT) construit un équilibre de Nash pur.

Théorème 22 ([Even-Dar et al., 2003, Fotakis et al., 2002]) *Calculer un équilibre de Nash pur peut se résoudre en temps polynomial pour les jeux d'ordonnancement.*

La figure 5.2 représente trois équilibres de Nash purs dans un jeu d'ordonnancement composé de 5 tâches ($w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 4, w_5 = 6$) sur deux machines identiques. Dans ces trois configurations, aucune tâche ne peut migrer en diminuant son coût. Il est à noter que l'équilibre de Nash pur représenté dans figure 5.2.(b) est obtenu par l'algorithme d'ordonnancement de liste LPT.

Notons, que les jeux d'ordonnancement correspondent aux jeux de congestion pondérés dont le réseau est composé de deux noeuds et de plusieurs liens parallèles. La table 5.5 résume les principales caractéristiques de ces deux classes de jeux.

¹Un algorithme d'ordonnancement de liste est un algorithme de placement de tâches au plus tôt selon l'ordre donné par une liste des tâches.

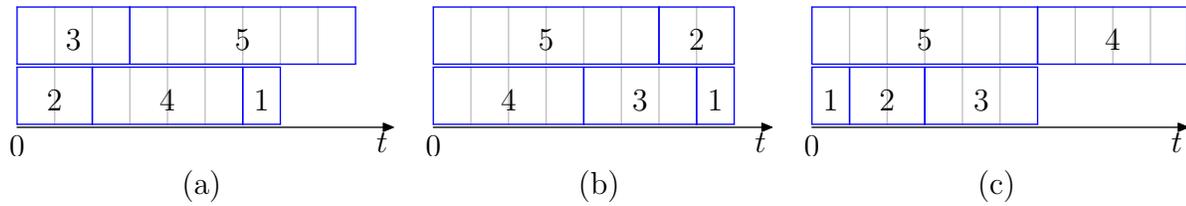


FIG. 5.2 – Trois équilibres de Nash purs dans le jeu d’ordonnement composé de 5 tâches ($w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 4, w_5 = 6$) sur deux machines identiques.

	Jeux de congestion		Jeux d’ordonnement
	non pondéré	pondéré avec des coûts	
		linéaire	
Ressources	Arêtes d’un graphe		Machines
Stratégie	Ensemble d’arêtes		Machine
jeu de potentiel	OUI	NON	OUI
Équilibre de Nash pur	OUI	NON	OUI
Calcul d’un équilibre pur	PLS-complet	NP-complet	Polynomial

TAB. 5.5 – Comparaison entre les jeux de congestion et les jeux d’ordonnement.

5.2.3 Les jeux d’allocation de tâches

Des variantes ont été étudiées : par exemple [Christodoulou et al., 2004]) considère des jeux que l’on appellera des jeux d’allocation de tâches. Comme pour les jeux d’ordonnement, le joueur est responsable d’une tâche et dans ce contexte, son coût correspondra à la date de fin d’exécution de sa tâche. Mais chaque ressource r a une fonction C_r qui retourne un coût $C_{r,j}$ pour chaque joueur j un en fonction de l’ensemble $\lambda \subset [n]$ de tâches allouées. Le coût du joueur i pour le profil de stratégies pures Q est donc donné par $c_i(Q) = C_{q_i,i}(\{j|q_j = q_i\})$. Les fonctions $C_{r,j}$ peuvent modéliser les vitesses, les politiques d’ordonnement des machines mais aussi les coût associés pour chaque joueur comme le temps de fin des tâches. Nous considérons trois politiques classiques suivantes :

- la politique *SPT* (*Shortest Processing Time* en anglais) ordonnent les tâches dans l’ordre croissant des longueurs de tâches.
- la politique *LPT* (*Longest Processing Time* en anglais) ordonnent les tâches dans l’ordre décroissant des longueurs des tâches.
- la politique *aléatoire* (*Randomized policy* en anglais) ordonnent aléatoirement les tâches suivant une loi uniforme de probabilité.

Remarquons que les jeux d’ordonnement sont équivalents aux jeux d’allocation de tâches dont les machines ont la politique d’ordonnement aléatoire. En effet, l’espérance de la fin d’une exécution d’une tâche sur une machine r dans ce modèle est égal à la moitié de la somme de la charge r plus sa longueur [Koutsoupias and Papadimitriou, 1999]. Ce n’est pas le cas pour les politiques d’ordonnement *SPT* et *LPT* (voir figure 5.3).

Tous ces types deux jeux restreints aux machines identiques ou uniformément reliées possèdent des équilibre de Nash purs.

Théorème 23 ([Christodoulou et al., 2004, Immorlica et al., 2009]) *Quelque soit*

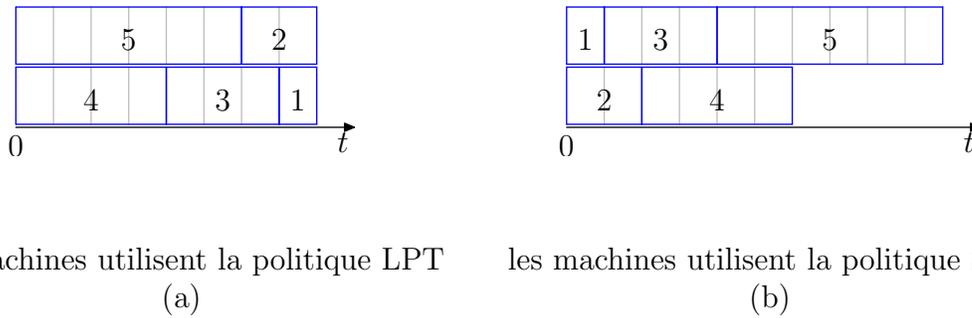


FIG. 5.3 – Équilibre de Nash dans le jeu d’ordonnement composé de cinq tâches ($w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 4, w_5 = 6$) sur deux machines identiques. Lorsque les machines utilisent les politiques d’ordonnement SPT, l’équilibre de Nash pur (b) n’est pas un équilibre pour la version du jeu d’ordonnement.

la politique d’ordonnement des machines qu’on suppose la même sur toutes les machines, les jeux d’allocation de tâches ayant des machines identiques ou uniformément reliées sont des jeux de potentiel ordinal.

En effet, dans [Christodoulou et al., 2004], l’existence d’un équilibre de Nash pur a été prouvée de la façon suivante : à chaque profil pur est associé un vecteur regroupant les coûts de tous les joueurs triés en suivant l’ordre décroissant lexicographique. A partir d’un profil pur arbitraire, une suite de profils est construite de telle façon que le profil au rang t est identique à celui du rang $t - 1$ à l’exception d’un joueur qui améliore son coût en choisissant une autre stratégie (correspondant à une meilleur réponse). En fait, à partir de cette construction, il est facile de construire une fonction de potentiel ordinal $\phi(Q) = \sum_i c_i(Q)Z^{n-i}$ avec Z une borne supérieure du coût.

Une extension naturelle est d’étendre ces jeux, en considérant maintenant qu’un joueur est responsable de plusieurs tâches. Ainsi, le coût d’un joueur peut être redéfini comme la somme des dates de fin d’exécution de ses tâches.

Nous avons prouvé [43] qu’il n’existe pas d’équilibre de Nash pur même si le jeu ne possède que deux machines identiques ayant la même politique d’ordonnement (LPT, SPT, aléatoire). La figure 5.4 représente le contre-exemple : quelque soit la configuration, un des deux joueurs peut diminuer son coût.

Suite à ces résultats négatifs, une première approche est de considérer la notion ϵ -équilibre de Nash.

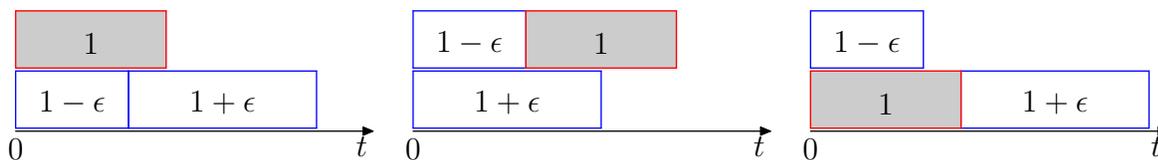


FIG. 5.4 – Exemple de jeu d’allocation n’ayant pas d’équilibre de Nash pur et de $(\frac{3}{2} - \epsilon)$ -équilibre de Nash : ses ressources sont deux machines identiques utilisant la politique SPT. Il y a deux joueurs : le joueur 1 est responsable de deux tâches de longueur $1 - \epsilon$ et de $1 + \epsilon$, le joueur 2 d’une tâche de longueur 1.

Définition 25 (ϵ -équilibre de Nash) Un ϵ -équilibre de Nash pur est un profil pur $s = (s_1, \dots, s_n)$ tel que

$$\forall i \quad \forall s'_i \in S_i \quad c_i(s_1, \dots, s_i, \dots, s_n) \leq \epsilon \times c_i(s_1, \dots, s'_i, \dots, s_n).$$

Nous avons constaté [43] que pour la politique SPT, il existe toujours des 2-équilibres de Nash purs : cela correspond tout simplement à ordonnancer les tâches au plus tôt en commençant par la plus courte jusqu'à la plus longue. De plus, l'instance représentée par la figure 5.4 ne possède pas de $(\frac{3}{2} - \epsilon)$ -équilibre de Nash : pour toute configuration, les deux joueurs peuvent diminuer leur coût de $(\frac{3}{2} - \epsilon)$.

Pour les politiques d'ordonnancement LPT et aléatoires, les jeux ne possèdent pas toujours d' ϵ -équilibre [43]. Ce travail est en cours. Pour l'instant pour deux machines, nous avons des algorithmes pour calculer la meilleure réponse en temps polynomial d'un joueur en fonction des stratégies des autres joueurs. C'est une première étape pour calculer le plus petit α tel qu'il existe un α -équilibre de Nash pour une instance donnée.

5.3 Évaluation de la qualité des équilibres de Nash

Les jeux précédents possèdent au moins un équilibre de Nash pur. En général, ils en possèdent plusieurs. Comment peut-on les comparer entre-eux ? Comment peut-on mesurer le prix du manque de coordination entre les joueurs ?

Par exemple, le jeu d'ordonnancement représenté par la figure 5.2 possède cinq tâches ($w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 4, w_5 = 6$) voulant s'exécuter sur les deux machines identiques. Il paraît difficile de les comparer :

	équilibre 5.2.(a)	équilibre 5.2.(b)	équilibre 5.2.(c)
coût moyen de joueurs	7,8	8	7,6
coût maximum de joueurs	9	8	10
différence des charges	2	8	4
⋮			

De plus, observons que tous les coûts des joueurs de l'équilibre 5.2.(b) sont plus petits que ceux de l'équilibre 5.2.(c). En fait l'équilibre 5.2.(c) n'est pas souhaitable pour tous les joueurs : changer de stratégies permet à tout le monde de faire diminuer son coût. C'est la notion de *Pareto optimalité*. Le jeu est dans un *état Pareto optimal* s'il est un état dans lequel aucun joueur ne peut diminuer son coût sans détériorer le coût d'au moins un autre.

Tout d'abord, plusieurs critères ont été définis pour cela en fonction de mesure d'équité entre les joueurs comme les indices de *Jain* [Jain et al., 1994] ou sur la distance entre l'équilibre de Nash et un état Pareto optimal [Legrand and Touati, 2007b]. L'index de *Jain* est un paramètre qui mesure les différents inégalités en terme de coût entre les joueurs tout en tenant compte de l'impact de leur existence.

Ces notions ont été principalement utilisées pour des jeux d'accès aux médium de communication dans les réseaux sans fils [Coucheney et al., 2009]. On pourra se référer à [Legrand and Touati, 2007a] pour un survol de ces deux notions. Un avantage de ces mesures est qu'elles se basent uniquement sur la fonction de coût des joueurs sans définir un coût global du système.

De plus, le système peut avoir plusieurs situations Pareto optimales, mais avec des performances sociales différentes.

Compte tenu de la description d'un système, l'évaluation des équilibres paraît difficile. Tout d'abord une mesure de qualité de performance (ou *coût social*) dans le jeu doit être définie. Par exemple, cela peut se quantifier en fonction des ressources utilisées par les joueurs : comme la bande passante, le temps maximum d'acheminement d'une information pour les jeux liés aux réseaux, ou la date moyenne de fin d'exécution des tâches, de la charge maximale des machines pour les jeux d'allocations de tâches.

Une fois le coût social exprimé (coût d'un état), il existe plusieurs mesures pour évaluer les équilibres. Une des mesures la plus étudiée est le prix de l'anarchie ou le prix de la non-coopération. Cette mesure indique la détérioration due au fait que les joueurs agissent dans leur propre intérêt.

Définition 26 (prix de l'anarchie) *Le prix de l'anarchie (noté PoA) d'un jeu J est le rapport entre le coût social du pire équilibre de Nash sur le coût social optimum.*

$$PoA(J) = \max_{E \in NE(J)} \frac{W(E)}{OPT(J)}$$

avec

- $W(E)$ représente le coût social de l'état E du jeu ;
- $OPT(E)$ représente le coût optimal parmi toutes les configurations du jeu ;
- $NE(J)$ est l'ensemble des équilibres de Nash.

Une autre mesure correspond au *prix de la stabilité* [Anshelevich et al., 2004], [Schulz and Moses, 2003], défini comme le rapport entre le coût social du meilleur équilibre de Nash sur le coût social optimum.

Il y a une multitude de travaux sur ces deux prix. Les deux paragraphes suivants correspondent à un résumé partiel des différents résultats sur les jeux de congestion et les jeux d'allocation de tâches.

Jeux de congestion : [Awerbuch et al., 2006], [Christodoulou and Koutsoupias, 2005a] prouvent que pour les jeux de congestion, le prix de l'anarchie pur est $5/2$ pour les fonctions de coût linéaires et $d^{\Theta(d)}$ pour des fonctions de coût étant des polynômes de degré maximum d . Cette dernière borne a été précisément calculée dans [Aland et al., 2006]. Dans le contexte des jeux pondérés, le prix de l'anarchie mixte est $(3 + \sqrt{5}/2)$ pour les fonctions de coût linéaires et $d^{\Theta(d)}$ pour des fonctions de coût étant des polynômes de degré maximum d .

Dans les jeux de congestion à fonctions de coût linéaires, le prix de la stabilité est $\Theta(\sqrt{n})$ avec n le nombre de joueurs [Christodoulou and Koutsoupias, 2005b].

De plus des travaux récents étudient le prix de l'anarchie et le prix de la stabilité en tenant en compte uniquement des équilibres avec certaines propriétés comme la Pareto optimalité : voir par exemple [Bilò et al., 2008, Chien and Sinclair, 2009].

Jeux d'ordonnement et jeux d'allocations de tâches : La table 5.6 représente un résumé partiel de différents travaux sur le prix de l'anarchie en considérant que le coût social est le coût maximum parmi tous les joueurs. Des résultats sur les prix de la stabilité ont été établis [Agussurja and Lau, 2007], [Angel et al., 2006].

De plus, il existe des algorithmes d'approximation qui calculent des équilibres de Nash purs dont le coût social est proche à r de l'optimal : voir [Feldmann et al., 2003] pour un survol.

Le tableau suivant est inspiré de [Dürr and Thang, 2009] :

	politique SPT	politique LPT	politique aléatoire
machines identiques	$2 - \frac{1}{m}$ [Graham, 1966] [Immorlica et al., 2009]	$\frac{4}{3} - \frac{1}{3m}$ [Christodoulou et al., 2004] [Graham, 1969]	$2 - \frac{2}{m+1}$ [Finn and Horowitz, 1979] [Schuurman and Vredeveld, 2007]
machines uniformément reliées	$\Theta(\log m)$ [Aspnes et al., 1997] [Immorlica et al., 2009]	$1.52 \leq PoA \leq 1.59$ [Dobson, 1984, Friesen, 1987] [Immorlica et al., 2009]	$\Theta(\frac{\log m}{\log \log \mathcal{O}})$ [Czumaj and Vöcking, 2002]
machine identiques à accès restreint	$\Theta(\log m)$ [Aspnes et al., 1997] [Immorlica et al., 2009]	$\Theta(\log m)$ [Azar et al., 1995] [Immorlica et al., 2009]	$\Theta(\frac{\log m}{\log \log m})$ [Awerbuch et al., 2006] [Gairing et al., 2004]

TAB. 5.6 – Prix de l'anarchie pour les jeux d'allocation de tâches

Chapitre 6

Des jeux répétés et dynamiques

Le chapitre 5, considère des jeux statiques dans le sens où ils sont joués une seule fois. La théorie des jeux se focalise essentiellement en effet sur des outils pour caractériser les équilibres en présence de joueurs rationnels lorsque le jeu est joué une unique fois. Elle vise à prédire ce qu'il faut jouer si chaque joueur est rationnel. Dans ces circonstances, il est en fait difficile d'interpréter la notion de stratégie mixte : pourquoi introduire des probabilités lors que le jeu est joué une seule fois ? Plusieurs discussions ont été faites sur ce sujet [Binmore, 1999], [Osbourne and Rubinstein, 1994].

Pour introduire des aspects dynamiques en théorie des jeux, deux grandes approches ont été développées.

La première consiste à considérer des jeux répétés. Nous présentons cette approche dans la première section.

La seconde consiste à étudier certains types de comportements des joueurs. En effet, il est naturel de supposer que la réponse de chacun des joueurs est donnée par certaines règles de comportement : le choix du joueur i est une certaine fonction $f_i(Q)$ du passé Q du jeu : f_i correspond à la fonction modélisant le comportement de i .

Si les f_i correspondent à prendre la meilleure réponse à la statistique du passé du jeu, on obtient la dynamique du joueur fictif, que nous présentons dans la section 6.2.

Si les f_i correspondent à prendre la meilleure réponse au choix de l'adversaire au temps précédent, on obtient la dynamique dite myope. La dynamique pavlovienne est une de ses variantes. Elle consiste à conserver sa stratégie tant que le coût par cette stratégie est inférieur à un certain seuil, et sinon à changer de stratégie pour une meilleure réponse. Nous présenterons des résultats sur la puissance de ce type de comportements dans la section 6.3.

Nos contributions personnelles relatives à ce chapitre concerne d'une part le calcul de stratégies comportementales pour obtenir des stratégies Pareto optimale dans le contexte du routage inter-domaine dans les jeux répétés [18]. Nous avons étudié le comportement des joueurs fictifs dans ce contexte [Boussaton, 2009]. D'autre part, une caractérisation partielle de la puissance des protocoles de population utilisant des dynamiques de jeu myopes [24], [Rabie, 2009].

Les travaux présentés dans ce chapitre ont été obtenus via le travail de l'étudiant en thèse Octave Boussaton, étudiant à l'université de Nancy I, et le co-encadrement de stagiaires de l'Ecole Normale supérieure de Lyon.

6.1 Jeux répétés

Considérons le problème du dilemme du prisonnier avec la matrice de coût suivante :

		joueur 2	
		D	C
joueur 1	D	(3,3)	(0,4)
	C	(4,0)	(1,1)

Le seul équilibre de Nash pur dans ce jeu statique est (D,D). Par contre, si le jeu est répété plusieurs fois, on aimerait pouvoir dire que l'intérêt de chacun des joueurs serait de choisir C à chaque étape puisque leur coût serait plus petit que si tous les deux choisissent D. Cela impliquerait une certaine coopération entre les deux joueurs. Plusieurs questions se posent alors. Par exemple est-ce un équilibre de Nash dans un jeu répété? Existe-t-il d'autres équilibres de Nash?

Mais avant de répondre partiellement à ces questions, nous allons retranscrire ces définitions dans le cadre des jeux répétés.

Tout d'abord, nous allons rappeler les notations des jeux statiques définies dans le chapitre 5 :

- $[n] = \{1, \dots, n\}$ est l'ensemble des joueurs
- \mathcal{S}_i (resp. K_i) est l'ensemble des stratégies pures (resp. mixtes) du joueur i
- $\mathcal{K} = \prod_{i=1}^n \mathcal{K}_i$ est l'espace des stratégies mixtes.
- $\mathcal{S} = \prod_{i=1}^n \mathcal{S}_i$ est l'espace des stratégies pures.
- c_i est la fonction de coût du joueur i .

Répéter (itérer) k fois un jeu, revient à étendre l'espace des stratégies en \mathcal{S}_i^k pour chacun des joueurs i . En effet, le joueur i choisit son action $\mathbf{q}_i(t) \in \mathcal{S}_i$ au temps t pour $t = 1, 2, \dots, k$. Le jeu répété k fois est donc un jeu sur le nouvel espace de stratégies $\prod_{i=1}^n \mathcal{S}_i^k$.

À l'étape t , le profil de ce jeu est noté par $\mathbf{q}(t)$. Conformément à [Binmore, 1999], pour éviter les confusions, nous nommerons *actions* les choix $\mathbf{q}_i(t)$ de chacun des joueurs à un instant donné, et *stratégies de comportement* les suites $Q_i = \mathbf{q}_i(1), \dots, \mathbf{q}_i(k)$, c'est-à-dire les stratégies pour ce jeu global (répété).

Maintenant, nous allons décrire comment représenter les répétitions du jeu, en particulier, en parlant d'histoire du jeu.

Définition 27 L'ensemble H_t des histoires de longueur t est $H_t = \mathcal{S}^t$. L'ensemble de toutes les histoires est $H = \cup_{t>0} H_t$.

Maintenant, nous allons introduire la notion de comportement d'un joueur qui correspond à ce qu'il va jouer à un instant donné en fonction de l'histoire du jeu (et surtout par rapport à sa connaissance de son passé).

Définition 28 Une stratégie de comportement du joueur i est une application f_i de H dans \mathcal{K}_i .

Répéter un jeu un nombre fini et fixé de fois (et connu par les joueurs) n'apporte rien de plus par rapport aux comportements des joueurs. En effet, considérons l'exemple du dilemme des prisonniers. À la dernière étape du jeu répété, la stratégie D domine la stratégie C, les deux joueurs jouent donc la stratégie D. Ensuite pour l'avant-dernière

étape, ils savent que la dernière étape sera (D, D). Donc, ils joueront la stratégie D. Un simple raisonnement de récurrence en arrière montre que chacun des joueurs a toujours intérêt à jouer D aux temps $t, t-1, \dots, 1$: voir [Binmore, 1999, page 349].

Il est plus intéressant de répéter le jeu un nombre infini de fois. Cette fois, la stratégie de coopération systématique, c'est-à-dire pour le dilemme des prisonniers jouer toujours C , devient un équilibre de Nash pour le jeu répété : en d'autres termes, coopérer devient intéressant pour chacun.

De façon similaire, si l'on suppose que l'on répète le jeu un nombre fini, mais aléatoire, de fois le jeu, avec à chaque étape une probabilité δ que le jeu continue encore une étape de plus, et $1 - \delta$ que ce soit la dernière étape, alors la probabilité que l'on joue la i ème étape vaut δ^i . En moyenne, le gain de chacun des joueurs est donc donné par l'équation 6.1 : voir [Binmore, 1999].

$$\text{cout}(H_k) = \sum_{t=1}^k \delta^{t-1} \text{cout}_i(\mathbf{q}(t)) \quad (6.1)$$

où

- $H_k = (\mathbf{q}(1), \dots, \mathbf{q}(k))$ est l'histoire du jeu.
- cout_i est la fonction de coût du joueur i .

Cette façon de compter peut aussi s'interpréter comme un coût actualisé. Le paramètre δ correspond à un *taux d'actualisation* : pour un joueur, le coût δx aujourd'hui est équivalent au coût x demain. En général, le coût d'un joueur est comptabilisé avec un le taux d'actualisation $\delta > 0$ (que l'on peut prendre égal à 1, si k est infini ; on choisit $\delta < 1$ si k est fini).

Définissons deux types de jeux infinis :

Définition 29 (jeu infiniment répété et jeu escompté) *Le jeu escompté au taux d'actualisation $\delta \in]0, 1]$ est le jeu répété une infinité de fois et dont le coût final du joueur i est $\sum_{t \geq 1} (1 - \delta) \delta^{t-1} \text{cout}_i(\mathbf{q}(t))$.*

Le jeu infiniment répété est le jeu répété une infinité de fois et dont le coût final du joueur i est $\lim_{k \rightarrow \infty} \sum_{t=1}^k \text{cout}_i(\mathbf{q}(t))$ si cette limite existe.

Le problème est que la stratégie de coopération systématique n'est pas en fait le seul équilibre de Nash. Jouer la stratégie D systématiquement reste aussi un équilibre de Nash du jeu répété.

Avant d'étudier les propriétés des équilibres de Nash, considérons *l'espace des coûts réalisables* : c'est un polytope convexe représentant l'ensemble des coûts réalisables dans le jeu répété. En tant que convexe, il s'écrit $\text{conv}(\mathcal{K})$ et rappelons que \mathcal{K} correspond à l'espace des stratégies mixtes de tous les joueurs : voir 6.1 pour le jeu du dilemme des prisonniers.

Puis nous allons définir l'espace des coûts pour des stratégies coopératives pour des joueurs rationnels. En fait, coopération est lié à punition. Cela vient du fait qu'il faut inciter les joueurs rationnels à rester en coopération. Mais comment punir ? Tout simplement en jouant la situation la plus défavorable pour le joueur i (voir équation 6.2).

Définition 30 *Pour chaque joueur i , le niveau de punition du joueur i est*

$$v_i = \max_{Q_{-i} \in \prod_{j \neq i} \mathcal{K}_j} \min_{\mathbf{q}_i \in \mathcal{K}_i} \text{cout}_i(\mathbf{q}_i, Q_{-i}) \quad (6.2)$$

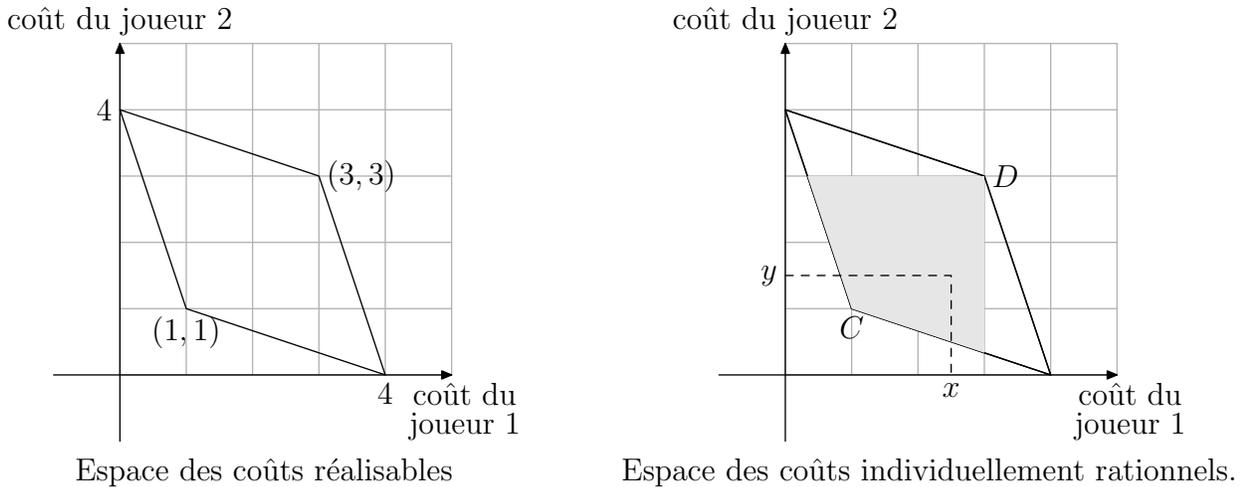


FIG. 6.1 – espace des coûts pour le problème du dilemme des prisonniers

Définissons un espace des coûts où les joueurs rationnels peuvent être sanctionnés si ils ne coopèrent pas.

Définition 31 *L'ensemble E des coûts individuellement rationnels est*

$$E = \text{conv}(\mathcal{K}) \cap \{(cost_i)_{i \in [n]} : cost_i \leq v_i \forall i \in [n]\}$$

En fait (voir par ex. [Binmore, 1999]) :

Théorème 24 (Le théorème de tout le monde (Folk Theorem)) *L'ensemble des coûts d'équilibre pour le jeu infiniment répété et escompté est l'ensemble des coûts individuellement rationnels.*

Intuitivement, les stratégies dans le jeu répété infini correspondent en fait aux stratégies mixtes dans le jeu statique. Considérons une stratégie mixte appartenant à E . On peut construire une stratégie de comportement correspondant à une simulation de la stratégie mixte dans le jeu statique et si un joueur i dévie, il est sanctionné.

Certains équilibres construits sont fortement critiquables en tant que solution rationnelle car ils reposent sur des punitions non-crédibles : étant donné qu'un joueur a dévié, rien assure que ses adversaires auront intérêt à punir pour toujours le cas échéant. Pour remédier à cette remarque la notion d'équilibre en sous-jeu parfait (ESJP) a été introduite par *R. J. Aumann, L. S. Shapley et A. Rubinstein* en 1976. De façon informelle, un profil de stratégies σ est un équilibre d'un sous-jeu parfait si pour toute histoire h de H , à partir de tout instant t , la suite des σ à partir t reste un équilibre. Cela permet d'oublier le passé et surtout de modéliser une sanction d'une déviation de façon limitée dans le temps. Dans ce contexte, le théorème *de tout le monde* reste valide. En effet si un joueur dévie en date t , il est puni jusqu'à la date t^2 , après quoi les joueurs oublient la déviation et se remettent à jouer la suite prévue.

Dans [18], nous nous sommes penchées sur le problème de construire un équilibre de Nash dans un jeu infiniment répété pour le routage inter-domaine. Dans un réseau, un seul sommet veut faire transiter ses informations vers un sommet destination en empruntant le chemin de plus petit coût. Chaque sommet possède un coût de transit et une fonction

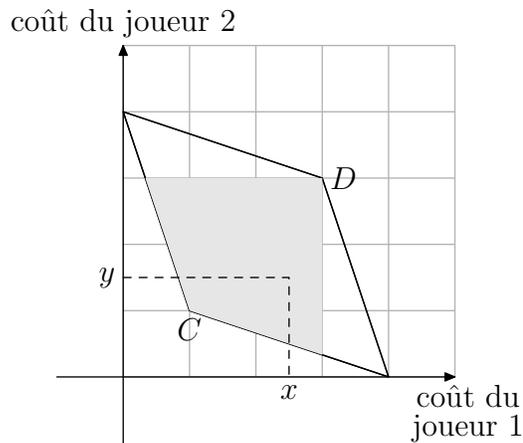


FIG. 6.2 – Pour tout point (x, y) de la zone grisée, il y a une paire q_1, q_2 de stratégies qui garantit le gain x pour le joueur 1, y pour le joueur 2, et tel que (q_1, q_2) soit un équilibre de Nash pour le dilemme des prisonniers itéré. En particulier pour le point C qui correspond à la coopération systématique, et pour le point D qui correspond à la défection systématique.

de prix pour transiter les données. Cette dernière peut fluctuer pour attirer le trafic. Et l'objectif est de maximiser la différence moyenne entre le prix de transit et le coût de faire transiter les messages. Dans ce contexte, pour certaines instances, nous avons trouvé un algorithme polynomial qui permet de construire une stratégie coopérative correspondant à un équilibre de Nash Pareto optimal.

Observons qu'en général, le nombre d'équilibres de Nash du jeu répété n'est pas dénombrable puisque toute stratégie mixte dans le jeu statique dans l'espace E a son pendant dans le jeu répété. Dans ce contexte, un jeu répété possède une multitude d'équilibres de Nash.

Le problème du dilemme des prisonniers répété a été beaucoup étudié. En particulier, [Axelrod, 1984], a décrit le résultat de tournois de règles de comportements pour le dilemme des prisonniers itéré. Axelrod argumente dans [Axelrod, 1984] qu'il existe un meilleur comportement, nommée *TIT – FOR – TAT*. Le comportement *TIT – FOR – TAT* consiste à coopérer à la première étape, puis ensuite à faire la même chose que son adversaire aux temps ultérieurs.

6.2 Vers la dynamique des joueurs fictifs

Maintenant, nous allons présenter des approches étudiant des types de comportement. Tout d'abord, nous allons présenter l'approche des dynamiques des joueurs fictifs en étudiant si elle permet d'apprendre un équilibre de Nash. Ensuite, nous retranscrivons la notion des jeux répétés dans les systèmes informatiques répartis plus particulièrement dans les protocoles de population définis dans [Angluin et al., 2004] inspirés par les réseaux de capteurs.

Dans cette section, nous allons introduire l'apprentissage d'équilibre de Nash que nous développerons plus en détail dans le chapitre 7. Le jeu est répété à chaque étape. En pratique, le joueur i est confronté au problème suivant à chaque instant t : quelle action jouée au temps t étant donnée l'histoire du jeu ?

Il est naturel de supposer que la réponse de chacun est donnée par certaines règles de comportement : pour chaque joueur i , $\mathbf{q}_i(t) = f(Q)$, où f_i est une fonction qui détermine le comportement de i en fonction du passé Q .

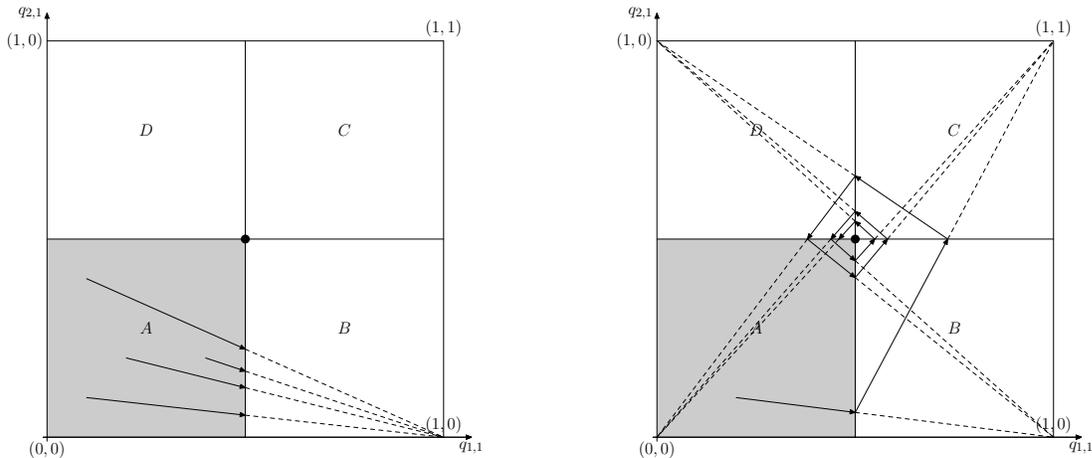
Nous allons étudier la dynamique des joueurs fictifs qui permet dans certains cas de garantir que le système converge vers un équilibre de Nash.

La notion de joueur fictif a été introduite dans [Brown, 1951]. Le joueur est un joueur *fictif* si le joueur a le comportement suivant : en fonction du passé, et de la statistique des stratégies de ses adversaires, le joueur jouera une meilleure réponse. A la première répétition du jeu, il se comportera comme un joueur myope, ensuite il constituera ses propres croyances sur la stratégie des ses adversaires. Cette croyance est basée sur les statistiques des stratégies jouées par les adversaires. Plus formellement, le processus de prédiction des joueurs dépend des actions passées : pour prédire la stratégie du joueur au temps t , il estimera que chaque joueur adverse jouera l'action i avec probabilité $n_i/(t-1)$ au temps t si ce dernier a joué l'action i n_i fois au temps $1, 2, \dots, t-1$.

Pour simplifier l'étude, en suivant le raisonnement de [Binmore, 1999] nous allons prendre l'exemple d'un jeu à deux joueurs symétrique ayant la forme stratégique suivante :

		joueur 2	
		1	2
joueur 1	1	(3,1)	(0,3)
	2	(1,2)	(2,0)

Si le joueur 2 a utilisé n_i fois l'action numéro i entre le temps 1 et $t-1$, alors le joueur 1 estimera que le joueur 2 jouera au temps t l'action i avec probabilité $q_{2,i}(t) = n_i/(t-1)$. De façon analogue, le joueur 2 évaluera la probabilité $q_{1,i}(t)$ que le joueur 1 joue l'action i .



direction de la dynamique dans la zone A exemple de comportement de la dynamique

FIG. 6.3 – Convergence vers un équilibre mixte dans un jeu à 2 joueurs

Pour étudier la dynamique du système, il suffit de passer d'un temps discret à un temps continu (voir [Binmore, 1999] pour plus de détails). Le système peut être décrit par le couple $(q_{1,1}, q_{2,1})$ avec $q_{i,1}$ correspondant à la probabilité que le joueur i joue la stratégie 1.

Une analyse simple sur les meilleures réponses pour chaque joueur (voir [Binmore, 1999]) montre que aussi longtemps que $(x_2(t), y_2(t))$ reste dans la zone A de la partie gauche de

la figure 6.3, le joueur 1 voudra utiliser la stratégie pure 2, et le joueur 2 la stratégie pure 1.

Ici, la dynamique $(q_{1,1}, q_{2,1})$ restera donc dans cette zone jusqu'à l'instant $t + \tau$ pour $\tau > 0$ suffisamment petit. Puisqu'on connaît le choix du joueur 2 entre le temps t et $t + \tau$, on peut donc évaluer $y_2(t + \tau)$ comme

$$q_{2,1}(t + \tau) = \frac{tq_{2,1}(t)}{t + \tau}. \quad (6.3)$$

Par réécriture, on a $\frac{q_{2,1}(t+\tau)-q_{2,1}(t)}{\tau} = -q_{2,1}(t)$. En faisant tendre τ vers 0, on obtient

$$q'_{2,1}(t) = \frac{q_{2,1}(t)}{t}.$$

De façon similaire, on obtient $q'_{1,1}(t) = \frac{1-q_{1,1}(t)}{t}$.

Les points qui satisfont ces deux équations sont sur le segment entre $(q_{1,1}(t), q_{2,1}(t))$ et le point $(1, 0)$. Sur les zones B , C , et D , une étude similaire peut être réalisée. La figure 6.3 montre comment la dynamique se comporte dans les quatre différentes zones. Une simple analyse sur les distances entre la dynamique et le point d'équilibre à chaque entrée de la dynamique dans une zone permet de constater que elle converge donc vers l'équilibre de Nash mixte du jeu.

Dans [Boussaton, 2009], nous avons analysé le comportement de la dynamique des joueurs fictifs de cette manière sur le problème de routage inter-domaine dans des jeux à deux joueurs suite au modèle proposé dans les travaux de [18]. Malheureusement, nous n'avons pas pu étendre le résultat de convergence au problème à trois joueurs.

En effet, la convergence n'a pas lieu sur tous les jeux : on peut facilement considérer des jeux où les trajectoires ne convergent pas, ou avec des cycles limites [Binmore, 1999], [Monderer and Sela, 1996], [Shapley, 1964]. Par exemple, pour le jeu généralisé "papier"- "ciseau"- "caillou" la dynamique du joueur fictif ne converge pas : voir par exemple [Binmore, 1999], [Shapley, 1964].

Dans [Robinson, 1951], il a été prouvé que la dynamique du joueur fictif converge vers un équilibre de Nash dans les jeux à somme nulle. Ce résultat a été aussi prouvé pour les jeux 2×2 [Miyasawa, 1961] non dégénérés c'est-à-dire tels qu'aucun joueur a des utilités identiques dans sa matrice. La condition de non dégénérescence est nécessaire [Monderer and Sela, 1996]. Ce résultat a été étendu pour les jeux non dégénérés à deux joueurs ayant 3 stratégies [Shapley, 1964] tout en donnant un exemple où la dynamique du joueur fictif ne converge pas sans cette hypothèse. Ce résultat a été étendu à des jeux à deux joueurs ayant n stratégies [Berger, 2003].

Beaucoup de travaux ont été consacrés à savoir quand le comportement de la dynamique du joueur fictif converge ou pas voir [Fudenberg and Levine, 1996] pour un survol. Il est à noter que :

Théorème 25 *La dynamique des joueurs fictifs converge vers des équilibres de Nash pour les jeux de potentiel exacts [Monderer and S., 1996] et pour les jeux de potentiel ordinal [Berger, 2007].*

Ces dynamiques du joueur fictif nécessitent une vision globale du jeu : chaque joueur doit connaître sa meilleure réponse, sa matrice de coût en fonction des stratégies de ses adversaires. L'hypothèse de connaître la matrice complète de coût est forte puisque cela implique que le joueur a une grande connaissance du système.

6.3 Vers les protocoles de population

La puissance de calcul de réseaux d'agents mobiles, anonymes, ayant des ressources limitées a été étudié récemment. En particulier, Angluin et al. ont proposé un modèle de calcul réparti dans [Angluin et al., 2004]. Le modèle a été proposé à l'origine pour modéliser les calculs réalisés par les réseaux de capteurs dans lesquels les agents sont passivement mobiles. Le premier exemple cité dans [Angluin et al., 2004] correspond à des capteurs attachés à des oiseaux et programmés pour vérifier certaines propriétés : par exemple comme déterminer si plus de 5 % de la population satisfait une propriété (avoir de la fièvre dans l'exemple de [Angluin et al., 2004]).

6.3.1 Description d'un protocole de population

Dans ce modèle, un *protocole de population* correspond à un ensemble fini d'agents ayant un nombre fini d'états. Les agents interagissent par paire choisie par un adversaire. Chaque interaction a pour conséquence une actualisation des deux agents selon une règle de transition δ . Comme les agents sont anonymes (agents non distinguables), une configuration d'un système à un instant donné est caractérisée par le nombre n_i d'individus étant dans l'état i , pour $1 \leq i \leq k$.

Plus formellement, un *protocole de population* est défini par un uplet $(Q, \Sigma, \iota, \omega, \delta)$. Les agents possèdent un ensemble fini d'états internes Q . Σ correspond à un ensemble de symboles d'entrée. Les fonctions ι, ω correspondent à des fonctions d'encodage des entrées et de décodage des sorties : $\iota : \Sigma \rightarrow Q$ est la fonction qui traduit les symboles d'entrée en états, et $\omega : Q \rightarrow \{0, 1\}$ est la fonction qui décode la sortie associé à un état. La relation $\delta \subseteq Q^4$ décrit comment les paires d'agents interagissent, en fait elle peut être vu comme une liste d'interactions possibles. Par abus de notation, une interaction peut être décrite par la notation $(q_1, q_2) \rightarrow (q'_1, q'_2)$, ou éventuellement par la notation $q_1 q_2 \rightarrow q'_1 q'_2$.

L'exécution du protocole correspond à une séquence d'interactions entre deux agents à partir d'une configuration initiale. À chaque étape, une interaction se déroule entre une paire d'agents. À l'issue d'une interaction entre u et v , les états de ces deux agents évoluent en fonction des règles de transition : l'agent u (resp. v) devient dans l'état u' (resp. v') si $(u, v, u', v') \in \delta$.

Notons par la suite $C \rightarrow C'$ si C et C' sont deux configurations telles que C' peut être obtenu par C par une interaction entre paire d'agents de C .

Une exécution est *équitable* si pour toutes les configurations C , si à partir de la configuration C , il existe une interaction permettant d'atteindre la configuration C' en une étape (c'est-à-dire $C \rightarrow C'$) alors pour toute dérivation $C_0 C_1 \dots$, avec $C_i \rightarrow C_{i+1}$ pour tout i , si C apparaît infiniment souvent, alors C' aussi.

À tout moment, pendant l'exécution d'un protocole de population, l'état q de chaque agent détermine la sortie par la valeur $\omega(q)$ à cet instant : une configuration C s'interprète par 1 (respectivement 0) si tous ses agents sont dans des états tels que $\omega(q) = 1$ (resp. 0).

La majorité des travaux sur ces modèles et leurs variantes portent sur la caractérisation des prédicats calculables $\psi : \mathbb{N}^m \rightarrow \{0, 1\}$.

On dit que le protocole *reconnait* le prédicat $\psi : \mathbb{N}^m \rightarrow \{0, 1\}$ si pour tout uplet (n_1, \dots, n_m) (configuration initiale), toute exécution débutant par le codage de cet uplet mène à des configurations dont les interprétations sont égales à 1 lorsque $\psi(n_1, \dots, n_m) = 1$, et aussi mène à des configurations dont les interprétations sont égales à 0 lorsque

$\psi(n_1, \dots, n_m) = 0$.

En particulier, dans [Angluin et al., 2004], [Angluin et al., 2006], la puissance de ce modèle a été caractérisée :

Théorème 26 ([Angluin et al., 2006]) – *Tout prédicat $\psi : \mathbb{N}^m \rightarrow \{0, 1\}$ définissable en arithmétique de Presburger est calculé par un protocole de population.*
 – *Réciproquement tout prédicat $\psi : \mathbb{N}^m \rightarrow \{0, 1\}$ calculé est définissable en arithmétique de Presburger.*

Comme les ensembles définissables en arithmétique de Presburger coïncident avec les ensembles semi-linéaires sur les entiers, le prédicat “5% d’agents dans l’état 1” est définissable en arithmétique de Presburger et par conséquent il existe un protocole pour décider s’il existe 5% d’agents dans l’état 1.

Pour plus d’informations sur les résultats sur ces modèles et ses variantes, nous renvoyons notre lecteur à [Angluin et al., 2007], [Aspnes and Ruppert, 2007].

6.3.2 Des jeux dans les protocoles de population

Nous avons vu que les protocoles de population correspondent à des interactions entre paires d’agents. Il est naturel de considérer que chacune de ces interactions est le fruit d’un jeu.

Nous nous sommes intéressées au comportement *pavlovien* et avons cherché à caractériser la puissance des protocoles correspondant.

Nous notons $BR_{\neq q}$ une meilleure réponse parmi toutes les stratégies privées de la stratégie q .

Définition 32 (Protocole de population associé à un jeu) *Considérons un jeu à deux joueurs ayant M_1 et M_2 comme matrices de coût avec $\mathcal{S}_1 = \mathcal{S}_2$. Soit Δ un réel que l’on nommera seuil.*

Le protocole associé à ce jeu correspond au protocole de population dont l’ensemble des états Q est l’ensemble des stratégies ($Q = \mathcal{S}_1 = \mathcal{S}_2$) et dont les règles de transition δ sont définies de la façon suivante :

$$(q_1, q_2, q'_1, q'_2) \in \delta$$

où

- $q'_1 = q_1$ quand $M_1(q_1, q_2) \leq \Delta$
- $q'_1 \in BR_{\neq q_1}(q_2)$ quand $M_{q_1, q_2} > \Delta$

et

- $q'_2 = q_2$ quand $M_2(q_2, q_1) \leq \Delta$
- $q'_2 \in BR_{\neq q_2}(q_1)$ quand $M_{q_2, q_1} > \Delta$,

Définition 33 (Protocole de population pavlovien) *Un protocole de population est pavlovien si on peut le construire à partir d’un jeu à deux joueurs et d’un seuil Δ*

Remarquons que lorsque le jeu n’est pas symétrique, il faut distinguer les deux entités : celui qui a initié l’interaction (correspondant au joueur 1) et celui qui subit cette interaction (correspondant au joueur 2) .

Il existe des protocoles pavloviens qui calcule le *OU* (resp. *ET*) logique sur les bits d'entrée. Il suffit simplement de considérer des protocoles à deux états qui calculent ces opérations : tout protocole à deux états est pavlovien.

Par exemple, voici un protocole calculant le *OU* logique :

$$\left\{ \begin{array}{l} 01 \rightarrow 11 \\ 10 \rightarrow 11 \\ 00 \rightarrow 00 \\ 11 \rightarrow 11 \end{array} \right. \text{ et les matrices de coût } M_1 = M_2 \text{ avec } \Delta = 0 :$$

		Adversaire	
		0	1
Joueur	0	0	+1
	1	+1	0

Par exemple pour la règle $01 \rightarrow 11$, lorsque un joueur jouant 0 rencontre un autre jouant 1, alors celui jouant 0 a un coût positif. Comme il a un comportement pavlovien et que son coût est supérieur au seuil, pour la prochaine interaction, il changera de stratégie en jouant 1.

Une remarque tout simple, est que tous les protocoles de population ne sont pas pavloviens [24]. En effet, il suffit de considérer un protocole de population symétrique à trois états et de construire un ensemble de règles afin de faire une permutation entre les états. Cette propriété empêche de construire une matrice de jeu.

Par la suite, nous noterons les prédicats de la façon suivante [*formule logique*], où une *formule logique* du type $\sum_i a_i \sigma_i \geq b_i$ (respectivement : $\sum_i a_i \sigma_i \equiv b_i \pmod{c_i}$) est vraie si la somme des nombres de symboles σ_i multipliés par a_i est supérieure à b_i (resp. congrue à b_i modulo c_i) : les protocoles calculant le *OU* (resp. *ET*) logique correspondent aussi à des prédicats de comptage d'agents : $[x.0 \geq 1]$ (resp. $[x.1 = 0]$), où x est l'entrée. Il suffit simplement changer les notations pour définir des protocoles de comptage $[x.\sigma \geq 1]$, et $[x.\sigma = 0]$ pour toutes les lettres σ de l'alphabet Σ .

Malheureusement, avec les définitions classiques des protocoles de population, le protocole ci-dessous ne calcule pas un *XOR* logique sur les bits en entrée et donc pas le prédicat de comptage équivalent $[x.1 \equiv 1 \pmod{2}]$.

$$\left\{ \begin{array}{l} 01 \rightarrow 01 \\ 10 \rightarrow 10 \\ 00 \rightarrow 00 \\ 11 \rightarrow 00 \end{array} \right. \quad (6.4)$$

Ce protocole permet de calculer le *XOR* logique dans le sens faible dans lequel au bout d'un certain temps, soit tous les agents sont dans l'état 0 si le *XOR* logique des bits d'entrée est 0. Sinon, seul un seul agent sera dans l'état 1 mais pas tous les agents.

Protocoles associés à un jeu symétrique. Nous conjecturons qu'il n'existe pas de protocoles pavlovien symétrique qui calcule le *XOR* logique (i.e. le prédicat de comptage équivalent à $[x.1 \equiv 1 \pmod{2}]$). De plus, dans [24], une caractérisation partielle de la puissance des calculs des protocoles pavloviens a été étudiée :

Théorème 27 Soit x l'entrée du protocole,

- le prédicat $[x.\sigma \geq k]$, qui est vrai quand le nombre d'occurrences du symbole σ en entrée est plus grand que k avec $k \leq 4$ ou k une puissance de 2.
- le prédicat $[x.\sigma \geq x.\sigma']$, avec σ, σ' deux symboles en entrée,

est calculable par un protocole pavlovien associé à un jeu symétrique.

On peut remarquer que le prédicat $[x.\sigma \geq x.\sigma']$ correspond au problème de la majorité qui détermine si le symbole σ apparaît plus que le symbole σ' en entrée.

Le problème d'élection d'un chef peut être résolu par un protocole pavlovien. Il suffit simplement de rendre le protocole classique non-symétrique par un protocole symétrique. Dans le protocole symétrique deux états L_1 et L_2 représentent le fait qu'un agent peut être chef. Voici le protocole résolvant un problème d'élection avec une taille de la population supérieure à 3 :

$$\left\{ \begin{array}{l} L_1L_2 \rightarrow L_1N \\ L_1N \rightarrow NL_2 \\ L_2N \rightarrow NL_1 \\ NN \rightarrow NN \\ L_2L_1 \rightarrow NL_1 \\ NL_1 \rightarrow L_2N \\ NL_2 \rightarrow L_1N \\ L_1L_1 \rightarrow L_2L_2 \\ L_2L_2 \rightarrow L_1L_1 \end{array} \right. \text{ avec sa matrice de jeu } (\Delta = 0) :$$

		Adversaire		
		L_1	L_2	N
Joueur	L_1	+3	0	+3
	L_2	+1	+3	+3
	N	+2	+3	0

Protocoles asymétriques. Dans [Rabie, 2009], la puissance de calcul des protocoles pavloviens asymétriques a été étudiée. Elle semble plus grande :

Théorème 28 Soit $x = (x_1, \dots, x_z)$ l'entrée du protocole,

- le prédicat $[x \cdot \sigma \geq k]$, avec $\sigma = (\sigma_1, \dots, \sigma_z)$, et n'importe quel entier k ;
 - le prédicat $[x \equiv r \pmod{m}]$ avec n'importe quels entiers r et m ;
 - le prédicat $[x.\sigma \geq x.\sigma']$, avec σ, σ' deux symboles en entrée,
- est calculable par un protocole pavlovien asymétrique.

Pour que la puissance de calcul corresponde à l'arithmétique de Presburger, il faudrait construire un protocole calculant des combinaisons linéaires booléennes. Nous conjecturons qu'il n'y en existe pas. Dans ce modèle, il semble difficile d'obtenir des preuves d'impossibilité. Par contre, avec l'ajout de l'hypothèse que les entités peuvent exécuter en parallèle des protocoles pavloviens, alors, on peut prouver que les protocoles pavloviens reconnaissent tout prédicat de l'arithmétique de Presburger.

Des premiers pas ont été amorcé pour les protocoles à population infinie [8], [Aupy, 2009].

Chapitre 7

Apprentissage des équilibres de Nash

Ce chapitre se consacre à des algorithmes d'apprentissage d'équilibres de Nash purs et à leur temps de convergence. La section 7.1 rappelle quelques résultats sur les dynamiques de meilleure réponse. Ensuite, la section 7.2 décrit un algorithme générique (en fait deux algorithmes) très général qui couvre ceux de la section 7.1. Elle présente des propriétés de convergence faible vers des équations différentielles. En paramétrant cet algorithme générique, on peut garantir que ces équations différentielles correspondent à des exemples classiques de dynamiques dans la théorie évolutionnaire des jeux. Cela est présenté dans la section 7.3. Nous discutons des liens qui existent entre le temps de convergence de l'algorithme et celui de l'équation différentielle ordinaire associée dans la section 7.4. Finalement, nous discutons de la dynamique de réplication sur les jeux de potentiel ordinal et exacts et nous discutons de leur temps de convergence dans la dernière section.

Les résultats discutés dans ce chapitre sont le fruit d'une collaboration avec Olivier Bournez, et correspondent essentiellement à [39] et [40].

Dans de nombreuses situations, certains participants peuvent tirer quelques avantages de l'utilisation de certaines ressources communes selon leur propre intérêt au détriment de l'intérêt collectif. Par exemple, les applications pair-à-pair avec des phénomènes tels que free-riding, fournissent un exemple des comportements problématiques dans des systèmes répartis à grande échelle. Le routage inter-domaine dans le réseau Internet, où certains intérêts économiques peuvent fournir des incitations pour altérer les performances globales, est un autre exemple.

Dans tous ces contextes, les participants adaptent souvent leur stratégie en fonction de leur connaissance locale du système par de petits ajustements afin d'améliorer leur propre profit. L'impact de chaque joueur sur le réseau est faible. Toutefois, comme le nombre d'acteurs est grand, une évolution globale du système peut se produire.

Ici, nous nous intéressons à comprendre sous quelles conditions le système peut converger vers des situations rationnelles, c'est-à-dire, vers des équilibres de Nash. Nous nous focalisons sur des dynamiques qui réalisent de petits ajustements stochastiques. Nous supposons ces ajustements complètement répartis puisque tous les joueurs participant aux jeux ont souvent une vue locale du système, sans vision globale du jeu.

Autrement dit, nous regardons le système évolué. A chaque instant, les joueurs choisissent les actions qu'ils vont réaliser. En fonction de l'état global du système, ils reçoivent une récompense, que l'on peut voir comme un coût. En fonction de leur passé, ils changent alors leur stratégie. Nous nous intéressons à l'évolution globale du système, et à comprendre sous quelles conditions on peut garantir que cette évolution converge vers des

équilibres de Nash.

7.1 Dynamique de meilleure réponse

Considérons tout d’abord les jeux de potentiel ordinal définis dans le chapitre 5. Rappelons que ces jeux possèdent toujours au moins un équilibre de Nash pur. La preuve de [Rosenthal, 1973] (qui correspond aussi au théorème 18 et à la discussion qui suit) peut se voir comme un algorithme d’apprentissage d’équilibres. En effet, elle se base sur le fait que si un joueur améliore son profit, alors la fonction de potentiel globale décroît aussi. La suite de profils correspondant à une suite de meilleures réponses individuelles correspond alors à une suite décroissante de la fonction de potentiel. Comme cette suite a un nombre fini de valeurs, une telle suite de profils doit être finie et donc doit atteindre un équilibre de Nash pur [Rosenthal, 1973]. Cette preuve de l’existence d’équilibres de Nash purs peut se transformer en la preuve que la dynamique de meilleure réponse converge vers un équilibre de Nash pur : si les joueurs jouent chacun leur tour et changent de ressources en améliorant systématiquement leur profit quand ils le peuvent, alors on aura convergence vers un équilibre de Nash pur.

Plus formellement :

Algorithme 0 (générique) de la dynamique de meilleure réponse :

1. A l’initialisation,
chaque joueur i a une stratégie pure.
2. A chaque étape t ,
 - Un joueur i est sélectionné en fonction d’une règle \mathcal{R}
 - Le joueur i choisit la (ou une s’il y en a plusieurs) stratégie qui minimise son coût (choix de la meilleure réponse).

La règle \mathcal{R} peut correspondre à

- la règle FIFO : “chaque joueur joue à son tour” ;
- la règle ALÉATOIRE : “le joueur est sélectionné aléatoirement uniformément” ;
- la règle MAX : “le joueur sélectionné est celui qui améliora le plus son coût” ;
- la règle MIN : “le joueur sélectionné est celui qui améliora le moins son coût” .

Plusieurs travaux sur les jeux de potentiel ordinal (comme les travaux [Goldberg, 2004], [Even-Dar et al., 2007], [Immorlica et al., 2009] pour les plus récents) ont calculé le temps de convergence selon la règle \mathcal{R} . Le calcul du temps de convergence se base alors sur la fonction de potentiel, en minorant à chaque mouvement la décroissante de la fonction de potentiel :

Théorème 29 ([Even-Dar et al., 2007]) *Pour un jeu d’ordonnancement avec des machines identiques, l’algorithme atteint un ϵ -équilibre de Nash pur en au plus $O(W^2/\epsilon)$ étapes quelle que soit la politique \mathcal{R} de choix des joueurs, où W la somme des poids de toutes tâches.*

Ces bornes peuvent être améliorées selon la politique \mathcal{R} .

Théorème 30 ([Even-Dar et al., 2007]) *Un équilibre de Nash pur est atteint*

- en au plus $n(n+1)/2$ étapes avec la règle FIFO
 - en au plus n étapes avec la règle MAX
 - en au plus $(\frac{n}{K})^K / (2(K!))$ étapes avec la règle MIN
- avec n joueurs et $K = m - 1$.

Contrairement à ce qui se passe pour les jeux d'ordonnement, les jeux de congestion, qui sont aussi des jeux de potentiel exact, possèdent des équilibres de Nash purs difficiles à calculer : le problème est PLS-complet [Fabrikant et al., 2004].

De ce fait, il est naturel de chercher à se restreindre à des instances avec de bonnes propriétés : [Chien and Sinclair, 2007] propose la condition du α -saut :

Définition 34 *Une instance du jeu de congestion respecte la condition du α -saut, avec $\alpha \geq 1$, si la fonction de coût C_e sur chaque ressource e satisfait $C_e(k+1) \leq \alpha \times C_e(k)$ pour tout $k \geq 1$.*

Pour les jeux symétriques de congestion, (lorsque les joueurs ont même ensemble de stratégies), calculer un équilibre de Nash est PLS-complet [Fabrikant et al., 2004] et même si les jeux respectent la condition du 2-saut [Chien and Sinclair, 2007]. De ces résultats, la dynamique de meilleure réponse peut converger vers l'équilibre en un temps exponentiel. C'est pour cette raison que la dynamique de meilleure réponse est légèrement modifiée dans [Chien and Sinclair, 2007] en considérant que les changements d'un joueur ne se font plus en allant vers la meilleure réponse mais vers une ϵ -amélioration : voir [Chien and Sinclair, 2007] pour la définition formelle.

On a alors le résultat suivant :

Théorème 31 ([Chien and Sinclair, 2007]) *Dans les jeux symétriques de congestion avec n joueurs satisfaisant la condition du α -saut, la dynamique de ϵ -amélioration évoquée ci-dessus converge vers un équilibre en $\lceil n\alpha\epsilon^{-1}\log(nC) \rceil$ où C est une borne supérieure des coûts de tous les joueurs.*

Si tous les joueurs choisissent leurs stratégies en parallèle, plutôt que chacun leur tour, on peut considérer l'algorithme suivant, qui à notre connaissance, a été étudié uniquement pour les jeux d'ordonnement :

1. A l'initialisation,
 - chaque joueur i a une stratégie pure.
2. A chaque étape t , pour chaque joueur i , faire en parallèle,
 - Soit r_i la ressource courante de la tâche i .
 - Choisir une autre ressource de façon uniforme.
 - Si $C_{r_i}(t) \geq C_j(t) + (1 + \epsilon)w_i$ alors déplacer la tâche i vers la ressource j avec la probabilité $\epsilon(1 - \frac{C_j(t)}{C_{r_i}(t)})$.

Ce algorithme converge vers un équilibre en $O(\log \log m + n^4)$ étapes pour les tâches unitaires [Berenbrink et al., 2006], et en $O(mn\Delta^3(\epsilon^{-2}))$ étapes pour les tâches ayant des longueurs arbitraires [Berenbrink and Schulte, 2007] avec m machines et n tâches en moyenne. Cette preuve de convergence utilise les techniques de martingale reprises à la fin du chapitre.

Maintenant, nous allons définir des classes d'algorithmes d'apprentissages qui permettront de retomber de façon générique sur les dynamiques usuelles de la théorie des jeux évolutionnaires comme par exemple celle de réplication, celle de *logit*...

7.2 Classes d'algorithmes d'apprentissage.

Dans cette section, nous considérons des algorithmes génériques d'apprentissage, c'est-à-dire si l'on préfère, des familles d'algorithmes d'apprentissage.

Nous allons raisonner sur des familles d'algorithmes plutôt que sur un algorithme très précis, car il se trouve que l'étude de tous les algorithmes de la famille peut se faire uniformément, comme nous allons le voir.

Plus précisément, nous allons nous focaliser sur les familles qui correspondent à des algorithmes que l'on obtient en instanciant l'algorithme 1 ou l'algorithme 2 dans ce qui suit, pour une certaine fonction F , un paramètre b , et éventuellement une fonction σ .

Dans ceux-ci :

- le paramètre b vise à être un réel positif proche de 0 ;
- la fonction F est donnée par ses composantes, c'est-à-dire est donnée par les fonctions $F_i^b(r_i(t), s_i(t), q_i(t))$, que l'on veut aussi générique que possible : on impose seulement que le vecteur $q_i(t)$ reste toujours un vecteur de probabilité : c'est-à-dire, que $q_{i,\ell}(t) \in [0, 1]$ et $\sum_{\ell} q_{i,\ell}(t) = 1$ soient conservés.
- La fonction σ est le plus souvent l'identité, mais elle peut en différer pour obtenir certains comportements : dans le cas général, la fonction σ est donnée par ses composantes, les fonctions σ_i , que l'on impose seulement de satisfaire à $\sum_{\ell \in \mathcal{S}_i} \sigma_{i,\ell}(q_i) = 1$;

Les deux familles ont pour point commun qu'à chaque étape, tous les joueurs choisissent simultanément leur action en fonction de leur stratégie mixte courante et tel que plusieurs (voir algorithme 1) ou un joueur (voir algorithme 2) réactualise(nt) à chaque étape sa stratégie mixte.

Algorithme 1 : version synchrone

1. A l'initialisation (à l'étape 0),
 $q_i(0) \in \mathcal{K}_i$ est un vecteur arbitraire de probabilité pour tout i .
2. Pour chaque étape t ,
 - Chaque joueur i sélectionne une stratégie $s_i(t) \in \mathcal{S}_i$ selon la distribution de probabilité $\sigma_i(q_i(t))$.
 - Ces différents choix permettent d'obtenir un coût $r_i(t)$ pour chaque joueur i .
 - Chaque joueur i réactualise $q_i(t)$ de la façon suivante :

$$q_i(t+1) = q_i(t) + bF_i^b(r_i(t), s_i(t), q_i(t)). \quad (7.1)$$

Algorithme 2 : version asynchrone

1. A l'initialisation (à l'étape 0),
 $q_i(0) \in \mathcal{K}_i$ est un vecteur arbitraire de probabilité pour tout i .
2. Pour chaque étape t ,
 - Chaque joueur i sélectionne une stratégie $s_i(t) \in \mathcal{S}_i$ selon la distribution de probabilité $\sigma_i(q_i(t))$.
 - Ces différents choix permettent d'obtenir un coût $r_i(t)$ pour chaque joueur i .
 - Un joueur $i(t)$ est sélectionné de façon aléatoire :
 le joueur $i = i(t)$ est sélectionné avec la probabilité p_i ($\sum_{i=1}^n p_i = 1$).
 - Chaque joueur i réactualise son vecteur $q_i(t)$ selon la règle suivante :

$$q_i(t+1) = q_i(t) + \begin{cases} bF_i^b(r_i(t), s_i(t), q_i(t)) & \text{si } i = i(t) \\ 0 & \text{sinon} \end{cases}$$

Remarque : Ces deux algorithmes sont complètement répartis : à chaque étape, le joueur i a seulement besoin de connaître son coût r_i et sa stratégie mixte courante q_i pour réactualiser sa propre stratégie mixte courante q_i .

La dynamique de chaque instance de ces algorithmes est donnée par l'évolution du profil du jeu $Q(t) = (q_1(t), \dots, q_n(t)) \in \mathcal{K}$, à chaque étape t . Ce qui nous intéresse est le comportement asymptotique de $Q(t)$ et sa convergence possible vers des équilibres de Nash.

Nous notons $\mathbb{E}[Y | Q]$, l'espérance de la variable aléatoire Y lorsque les joueurs jouent selon le profil de stratégies mixtes donné par Q : c'est-à-dire lorsque le joueur j choisit l'action $\ell \in \mathcal{S}_j$ avec la probabilité $q_{j,\ell}$.

Plus généralement, nous noterons $\mathbb{E}_\sigma[Y | Q]$ l'espérance de la variable aléatoire Y lorsque les joueurs jouent selon le profil de stratégies mixtes donné par $\sigma(Q)$: c'est-à-dire lorsque le joueur j choisit l'action $\ell \in \mathcal{S}_j$ avec la probabilité $\sigma_{j,\ell}(q_{j,\ell})$.

Nous faisons l'hypothèse que la moyenne $\mathbb{E}_\sigma[F_i^b(r_i(t), s_i(t), q_i(t)) | Q(t)]$ est toujours définie, que $G_i(Q) = \lim_{b \rightarrow 0} \mathbb{E}_\sigma[F_i^b(r_i(t), s_i(t), q_i(t)) | Q]$ existe et que $G_i(Q)$ est une fonction continue de Q .

Quelque soit l'algorithme utilisé, il est facile de constater que $Q(t)$ est une chaîne de Markov homogène.

Définissons $\Delta Q(t)$ comme $\Delta Q(t) = Q(t+1) - Q(t)$, et $\Delta q_i(t)$ comme $q_i(t+1) - q_i(t)$.

Nous pouvons écrire

$$\mathbb{E}[\Delta q_i(t) | Q(t)] = b\tilde{p}_i \mathbb{E}_\sigma[F_i^b(r_i(t), s_i(t), q_i(t)) | Q(t)], \quad (7.2)$$

où \tilde{p}_i est 1 pour une instance de l'algorithme 1, et p_i pour une instance de l'algorithme 2.

Posons $G_i(Q) = \lim_{b \rightarrow 0} \tilde{p}_i \mathbb{E}_\sigma[F_i^b(r_i(t), s_i(t), q_i(t)) | Q(t)]$.

De façon informelle, $Q(t)$ converge vers la solution de l'équation différentielle ordinaire suivante :

$$\begin{cases} \frac{dX}{dt} & = G(X) \\ X(0) & = Q(0) \end{cases} \quad (7.3)$$

On peut s'y attendre, en raison du raisonnement informel suivant : supposons qu'on puisse remplacer $\mathbb{E}[\Delta q_i(t) | Q(t)]$ par $\Delta q_i(t)$ dans l'équation (7.2). Avec le changement

de variable $t \leftarrow tb$, on obtient

$$q_i(t+b) - q_i(t) = b\tilde{p}_i \mathbb{E}_\sigma [F_i^b(r_i(t), s_i(t), q_i(t)) | Q(t)].$$

En approximant $q_i(t+b) - q_i(t)$ par $b \frac{dq_i}{dt}(t)$ pour b petit, et le membre droit par $bG_i(Q)$, on s'attend à ce que le système se comporte comme l'équation différentielle (7.3) lorsque b est proche de 0.

Pour le prouver formellement, il faut introduire l'interpolation affine par morceaux (en anglais *piecewise-linear*) de $Q(t)$ notée $Q^b(\cdot)$ définie par

$$Q^b(t) = Q(\lfloor t/b \rfloor) + (t/b - \lfloor t/b \rfloor)(Q(\lfloor t/b \rfloor + 1) - Q(\lfloor t/b \rfloor)). \quad (7.4)$$

La fonction $Q^b(\cdot)$ appartient à l'espace de toutes les fonctions de \mathbb{R} dans \mathcal{K} qui sont continues droite et pourvues de limites gauche (fonctions *cad-lag*).

On peut alors considérer la suite $\{Q^b(\cdot) : b > 0\}$ et s'intéresser à la limite de cette suite notée $Q(\cdot)$ quand $b \rightarrow 0$.

Rappelons qu'une famille $(Y_t)_{t \in \mathbb{R}}$ de variables aléatoires converge *faiblement* vers une variable aléatoire Y , si $\mathbb{E}[h(X_t)]$ converge vers $\mathbb{E}[h(Y)]$ pour chaque fonction h bornée et continue.

On a alors le résultat formel suivant :

Théorème 32 *La suite des processus d'interpolation $\{Q^b(\cdot)\}$ converge faiblement, vers $Q(\cdot)$ quand $b \rightarrow 0$, vers la solution (unique et déterministe) de l'équation différentielle ordinaire (7.3).*

Ce théorème se prouve en utilisant des méthodes de preuve de convergence faible : formellement le théorème précédent se déduit de la preuve de [Stroock and Varadhan, 1979, Théorème 11.2.3].

En fait, nous voyons maintenant que le paramètre b correspond au pas de discrétisation de la dynamique continue donnée par (7.3).

Le résultat précédent est valable quelle que soit l'instance des familles d'algorithmes 1 et 2. Nous allons instancier dans la section 7.3 ces classes d'algorithmes pour obtenir différentes dynamiques continues, connues pour avoir leurs équilibres reliés aux équilibres de Nash des jeux.

7.3 Exemples de dynamiques

Nous allons instancier ces deux algorithmes sur trois types de dynamiques : des dynamiques de réplication de multi-population, *logit* et de la meilleure réponse seuillée.

7.3.1 Dynamique de réplication

Considérons le premier exemple en posant que

- les fonctions σ_i sont la fonction identité ;
- les fonctions $F_i^b(r_i(t), s_i(t), q_i(t))$ sont définis de la façon suivante :

$$F_i^b(r_i(t), s_i(t), q_i(t)) = \gamma(r_i(t))(e_{s_i(t)} - q_i(t)) \quad (7.5)$$

où $\gamma : \mathbb{R} \rightarrow [0, 1]$ est une fonction affine décroissante. En supposant que tous les coûts sont positifs, et par linéarité de la fonction d'espérance, nous pouvons noter que tous les coûts sont bornés par une constante M , et nous pouvons prendre $\gamma(x) = \frac{M-x}{M}$ par exemple.

Rappelons que $e_{s_i(t)}$ correspond au vecteur unitaire avec l'élément $s_i(t)$ égal à 1.

Maintenant, nous allons définir vers quelles équations différentielles ordinaires, les deux algorithmes convergent alors faiblement.

En supposant qu'à l'étape t , le joueur i choisit la stratégie $s_i(t)$ et s'il réactualise sa stratégie courante :

$$\Delta q_{i,\ell}(t) = q_{i,\ell}(t+1) - q_{i,\ell}(t) = \begin{cases} -b\gamma(r_i(t))q_{i,\ell}(t) & \text{si } s_i(t) \neq \ell \\ -b\gamma(r_i(t))q_{i,\ell}(t) + b(\gamma(r_i(t))) & \text{si } s_i(t) = \ell. \end{cases}$$

Alors, nous avons

$$\begin{aligned} G_{i,\ell}(Q) &= \lim_{b \rightarrow 0} \frac{1}{b} \tilde{p}_i \mathbb{E} [\Delta q_{i,\ell}(t) \mid Q(t)] \\ &= q_{i,\ell} \tilde{p}_i (\mathbb{E} [\gamma(r_i(t)) \mid Q(t), s_i(t) = \ell] - \mathbb{E} [\gamma(r_i(t)) \mid Q(t)]). \end{aligned}$$

En d'autres termes, selon le théorème 32, il y a convergence faible vers les solutions de l'équation (7.3), qui correspondent ici aux solutions de l'équation différentielle ordinaire (en supposant que γ est une fonction décroissante dont le domaine d'image est entre 0 et 1) :

$$\frac{dq_{i,\ell}}{dt} = -q_{i,\ell} \tilde{p}_i (c_i(e_\ell, Q_{-i}) - c_i(q_i, Q_{-i})), \quad (7.6)$$

Autrement dit, l'algorithme stochastique se comporte comme une équation différentielle ordinaire correspondant à une dynamique de réplication (de multi-population).

Les propriétés de telles dynamiques sont bien connus : voir les théorèmes *de tout le monde* de la théorie des jeux évolutionnaires.

Proposition 1 *Toutes les assertions suivantes sont vraies pour les solutions de l'équation (7.6) : (i) Tous les équilibres de Nash sont des points stationnaires. (ii) Tous les équilibres de Nash stricts sont asymptotiquement stables. (iii) Tous les points stationnaires stables sont des équilibres de Nash.*

7.3.2 Dynamique de *logit*

Pour obtenir notre second exemple, inspiré de [Cominetti et al., 2008], il suffit de définir les fonctions σ_i de la façon suivante :

$$\sigma_{i\ell}(q_i) = \frac{\exp(q_{i,\ell}/\kappa)}{\sum_j \exp(q_{i,j}/\kappa)}, \quad (7.7)$$

c'est-à-dire une dynamique "*logit*", où κ est une constante positive. Lorsque κ tend vers 0, cette dynamique converge vers la dynamique de la meilleure réponse, tandis que lorsque κ tend vers l'infini, cette dynamique se comporte comme un choix uniforme [Hofbauer and Sigmund, 2003].

Considérons $F_i^b(r_i(t), s_i(t), q_i(t))$ défini comme suit

$$F_i^b(r_i(t), s_i(t), q_i(t)) = (\gamma(r_i(t)) - q_{i,s_i(t)})e_{s_i(t)} \quad (7.8)$$

Dans ce cas, nous avons :

$$\begin{aligned} G_{i,\ell}(Q) &= \lim_{b \rightarrow 0} \frac{1}{b} \tilde{p}_i \mathbb{E}_\sigma [\Delta q_{i,\ell}(t) \mid Q(t)] \\ &= \sigma_{i,\ell}(q_{i,\ell}(t)) \tilde{p}_i (\mathbb{E}_\sigma [\gamma(r_i(t)) \mid Q(t), s_i(t) = \ell] - q_{i,\ell}). \end{aligned}$$

En d'autres termes, selon le théorème 32, il y a convergence faible vers les solutions de l'équation (7.3), qui correspondent ici aux solutions de l'équation différentielle ordinaire :

$$\frac{dq_{i,\ell}}{dt} = \sigma_{i,\ell}(q_{i,\ell}(t)) (\mathbb{E}_\sigma [\gamma(r_i(t)) \mid Q(t), s_i(t) = \ell] - q_{i,\ell}). \quad (7.9)$$

Les points limites de ces équations différentielles ordinaires correspondent aux équilibres de Nash d'un jeu dont les coûts sont perturbés par un terme additif tendant vers 0 lorsque κ tend vers 0 :

Proposition 2 ([Hofbauer and Sigmund, 2003],[Cominetti et al., 2008]) *Les points stationnaires des dynamiques (7.9) correspondent aux équilibres de Nash des jeux dont les utilités sont données par $v_i(Q) = c_i(Q) - \kappa \sum_\ell q_{i,\ell} \log q_{i,\ell}$.*

Une fois de plus, pour des jeux en général, nous obtenons que la limite pour $b \rightarrow 0$ des dynamiques des algorithmes stochastiques est une équation différentielle ordinaire dont les points limites stables (si ils existent) sont des équilibres de Nash lorsque $t \rightarrow \infty$. S'il y a convergence de l'équation différentielle ordinaire alors, on peut espérer que ces algorithmes stochastiques apprennent des équilibres.

7.3.3 Dynamique de la meilleure réponse seuillée

Pour notre troisième exemple, nous prenons des fonctions σ_i comme la fonction identité, et les fonctions $F_i^b(r_i(t), s_i(t), q_i(t))$ comme définies de la façon suivante :

1. choisir une autre stratégie pure $e_j(t)$ de façon uniforme : la stratégie j est choisie avec une probabilité $\frac{1}{m_i}$.
2. jouer la stratégie $e_j(t)$. Cela permet au joueur i d'obtenir un nouveau coût $r_i(t)$.
3. considérer alors

$$F_i^b(r_i(t), s_i(t), q_i(t)) = \nu(r_i(t) - r_j(t))(e_j - q_i(t)),$$

où ν est une fonction croissante qui retourne 0 pour des arguments négatifs et 1 pour des arguments plus grand que $\epsilon > 0$.

Cela mène à des dynamiques proches des dynamiques considérées par les auteurs de [Berenbrink and Schulte, 2007]. Aussi longtemps que l'état courant $Q(t)$ n'est pas un ϵ -équilibre de Nash, certains joueurs chercheront à améliorer leur réponse. Ainsi, les points stationnaires doivent correspondre à des ϵ -équilibres de Nash.

7.4 Propriétés de convergence.

7.4.1 Bornes sur la vitesse de convergence

Le défaut des résultats de convergence faible dans l'esprit du théorème 32 est qu'ils ne nous donnent aucun renseignement sur la vitesse de convergence. En effet, il n'est pas possible de déduire, étant donné $\epsilon > 0$, quel est le temps T , pour que les états de notre système soient à une distance inférieure à ϵ de l'équation différentielle ordinaire et quel est l'ordre de grandeur de b pour avoir une telle propriété.

Il est en fait possible d'obtenir des bornes sur la vitesse de convergence.

Observons pour cela que la dynamique (7.1) peut s'écrire

$$Q(t+1) = Q(t) + bG(Q) + bD(Q) + bR(Q) \quad (7.10)$$

où

$$D(Q) = \mathbb{E}_\sigma [F_i^b(r_i(t), s_i(t), q_i(t)) | Q(t)] - G_i(Q)$$

peut être vu comme une perturbation déterministe et

$$R(Q) = F_i^b(r_i(t), s_i(t), q_i(t)) - \mathbb{E}_\sigma [F_i^b(r_i(t), s_i(t), q_i(t)) | Q(t)]$$

comme une perturbation aléatoire.

Par hypothèses, $D(Q)$ est en $\mathcal{O}(b)$ et $\mathbb{E}[R(Q) | Q] = 0$.

En l'absence de ces perturbations, la dynamique (7.10) se comporte comme un schéma de discrétisation d'Euler $q_i(t+1) = q_i(t) + bG_i(Q)$ pour résoudre l'équation différentielle ordinaire (7.3).

L'idée est de généraliser les résultats connus sur les bornes sur les erreurs commises par les méthodes d'Euler pour résoudre les équations différentielles en prenant en compte les perturbations.

Cela donne les résultats suivants dont les preuves sont inspirées par les constructions de [Benaïm, 1999].

Théorème 33 ([40]) *Supposons que l'équation différentielle ordinaire converge vers un équilibre de Nash Q^* pour certaines conditions initiales $Q(0)$. Soit $T(\epsilon)$ le temps de nécessaire pour converger vers des points à distance ϵ de Q^* .*

Alors pour tout $\epsilon > 0$, l'algorithme stochastique partant de $Q(0)$ va converger avec une forte probabilité vers un état à distance 2ϵ de Q^ . Cela se produira en un nombre d'étapes de l'ordre de*

$$\frac{1}{\epsilon} T(\epsilon) \exp(\Lambda T(\epsilon)),$$

si b est de l'ordre de $\epsilon \exp(-\Lambda T(\epsilon))$, et si Λ est une constante telle que G est Λ -Lipschtizienne.

Autrement dit, le temps requis par l'algorithme stochastique pour converger est formellement relié au temps de convergence de l'équation différentielle ordinaire correspondante.

En d'autres termes,

Corollaire 33.1 ([40]) *Borner le temps de convergence de l'algorithme stochastique correspond exactement à borner le temps de convergence de l'équation différentielle ordinaire associée (7.3).*

7.4.2 Exemple : le jeu symétrique à deux joueurs

Dans cette section, nous considérons les dynamiques décrites dans la section 7.3.1, qui tendent vers une l'équation différentielle ordinaire de la forme (7.6). Nous faisons leur étude exhaustive sur les jeux génériques à deux joueurs symétriques.

Les dynamiques de réplication ne changent pas même si les modifications de la matrice de coût décrites dans la section 5.1.3 sont réalisées. Les dynamiques (7.6) peuvent ainsi être formulées de la façon suivante :

$$\begin{aligned} \frac{dq_{1,1}}{dt} &= (a_1 q_{2,1} - a_2 q_{2,2}) q_{1,1} q_{1,2} & \frac{dq_{2,1}}{dt} &= (a_1 q_{1,1} - a_2 q_{1,2}) q_{2,1} q_{2,2} \\ \frac{dq_{1,2}}{dt} &= -\frac{dq_{1,1}}{dt} & \frac{dq_{2,2}}{dt} &= -\frac{dq_{2,1}}{dt} \end{aligned}$$

Dans le chapitre 5, nous avons rappelé la classification de [Weibull, 1995], des jeux symétriques à deux joueurs ayant deux stratégies. Les comportements peuvent se ranger en quatre catégories.

Catégorie I avec $a_1 a_2 < 0$, $a_1 < 0$ et $a_2 > 0$. Nous allons étudier le comportement de la trajectoire de la dynamique de réplication. La figure 7.1 représente le portrait de phase¹ dans le jeu où $a_1 = -1$ et $a_2 = 1$.

En constatant que $q_{2,1} + q_{2,2} = 1$, il est facile d'observer que $(a_1 q_{2,1} - a_2 q_{2,2}) < -\delta$ où $\delta = \min(-a_1, a_2)$. Comme $\frac{dq_{1,1}}{dt} = (a_1 q_{2,1} - a_2 q_{2,2}) q_{1,1} (1 - q_{1,1})$, nous avons $\frac{dq_{1,1}}{dt} < \delta(1 - q_{1,1}) q_{1,1}$ où $\delta = \min(-a_1, a_2)$. Par conséquent, nous obtenons

- $q_{1,1}(t) < e^{-\delta t} q_{1,1}(0)$ dès que $q_{1,1}(0) \neq 0$,
- $q_{2,1}(t) < e^{-\delta t} q_{2,1}(0)$ dès que $q_{2,1}(0) \neq 0$ (de façon symétrique).

Cela implique que toute solution de l'équation différentielle ordinaire du système va converger vers l'unique équilibre de Nash pur dont le profil est $q_{1,1} = q_{2,1} = 0$. Par conséquent, le temps de convergence $T(\epsilon)$ de l'équation différentielle pour atteindre cet équilibre est de l'ordre de $\frac{1}{\delta} \ln(\frac{1}{\epsilon})$.

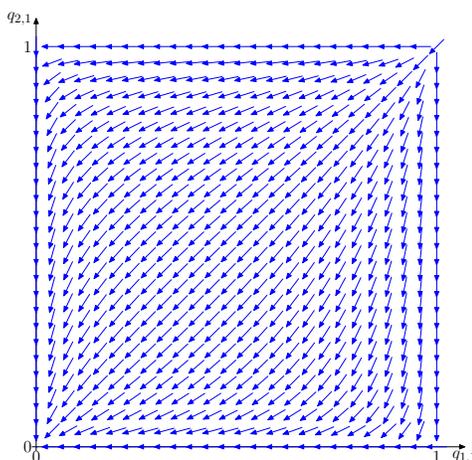


FIG. 7.1 – Le portrait de phase de l'équation de réplication pour un jeu de catégorie I

¹Un portrait de phase est une représentation géométrique d'un système dynamique : la direction de la trajectoire du système est représenté pour chaque point.

Catégorie IV avec $a_1 a_2 < 0$, $a_1 > 0$ et $a_2 < 0$: Le raisonnement et les résultats sont identiques à ceux de la catégorie I. Des calculs similaires permettent de constater que $\frac{dq_{1,2}}{dt} < \delta(1 - q_{1,2})q_{1,2}$, et que $\frac{dq_{2,2}}{dt} < \delta(1 - q_{2,2})q_{2,2}$ où $\delta = \min(a_1, -a_2)$. Le temps de convergence $T(\epsilon)$ vers l'unique équilibre de Nash pur est de l'ordre de $\frac{1}{\delta} \ln(\frac{1}{\epsilon})$.

Catégorie II avec $a_1 a_2 > 0$ et a_1, a_2 les deux sont positifs : La figure 7.2(a) représente le portrait de phase dans le jeu où $a_1 = 1$ et $a_2 = 2$. Pour étudier le comportement de la dynamique, nous posons $\lambda = \frac{a_2}{a_1 + a_2} - q_{2,1}$ et $\gamma = \frac{a_1}{a_1 + a_2} - q_{1,1}$.

Par calcul, observons $\frac{dq_{1,1}}{dt} = -\lambda(a_1 + a_2)q_{1,1}(1 - q_{1,1})$. Si $\lambda > 0$, alors $\frac{dq_{1,1}}{dt} < -(a_1 + a_2)\lambda q_{1,1} < 0$. Si $\lambda < 0$, alors $\frac{dq_{1,1}}{dt} > -(a_1 + a_2)\lambda q_{1,1} > 0$, ce qui entraîne $\frac{d(1 - q_{1,1})}{dt} < (a_1 + a_2)\lambda(1 - q_{1,1}) < 0$.

De façon symétrique, nous obtenons

$$- \frac{dq_{2,1}}{dt} < -(a_1 + a_2)\gamma q_{2,1} < 0 \text{ si } \gamma > 0$$

$$- \frac{d(1 - q_{2,1})}{dt} < (a_1 + a_2)\gamma(1 - q_{2,1}) < 0 \text{ si } \gamma < 0$$

Par conséquence, lorsque $\lambda > 0$, $q_{1,1}$ décroît, tandis que pour $\lambda < 0$, $q_{1,1}$ augmente. De façon symétrique pour $q_{2,1}$. L'équilibre de Nash mixte dont le profil correspond à $q_{1,1} = \frac{a_2}{a_1 + a_2}$ et à $q_{2,1} = \frac{a_1}{a_1 + a_2}$, est un point de selle. A l'exception de ce point, quelque soit le point initial, la dynamique converge vers un de ces deux équilibres de Nash pur (dont les profils sont $q_{1,1} = q_{2,1} = 0$ et $q_{1,1} = q_{2,1} = 1$).

L'espace peut se diviser en quatre régions dont chaque région a pour coin l'équilibre de Nash mixte.

Pour les deux régions diagonales, en fonction des calculs précédents, les dynamiques convergent vers un équilibre de Nash pur sans quitter sa région avec la vitesse de l'ordre de $\exp(-\delta t)$, pour $\delta = \min(|\lambda|, |\gamma|)$. Par conséquence, le temps $T(\epsilon)$ de convergence est de l'ordre de $\frac{1}{\delta} \ln(\frac{1}{\epsilon})$.

Pour les deux autres régions, la dynamique quitte sa région en temps de l'ordre de $\frac{1}{\delta} \ln(\frac{1}{\epsilon'})$ avec $\delta = \min(|\lambda|, |\gamma|)$ et avec ϵ' la distance entre ce point et l'équilibre de Nash mixte.

Catégorie III où $a_1 a_2 > 0$ et où a_1, a_2 sont tous les deux négatifs. La figure 7.2(b) représente le portrait de phase dans le jeu où $a_1 = -1$ et $a_2 = -2$.

Rappelons qu'il existe un unique équilibre de Nash mixte et deux équilibres de Nash purs. L'équilibre de Nash mixte a le profil suivant : $q_{1,1} = \frac{a_2}{a_1 + a_2}$, $q_{2,1} = \frac{a_1}{a_1 + a_2}$. Le premier équilibre de Nash pur a le profil suivant : $q_{1,1} = 1$, $q_{2,1} = 0$. Le second équilibre de Nash pur a le profil suivant : $q_{1,1} = 0$, $q_{2,1} = 1$.

Les dynamiques peuvent être étudiées comme pour celle de la catégorie II en partageant l'espace en quatre régions. La différence est maintenant que les régions diagonales sont quittées en un temps fini et que dans les régions anti-diagonales, les dynamiques convergent vers les équilibres de Nash purs.

La discussion précédente permet d'affirmer la convergence vers un point à distance plus petite que ϵ d'un équilibre de Nash en un temps polynomial en $\frac{1}{\epsilon}$, si b est de l'ordre de $\frac{1}{\epsilon}$.

Pour les jeux généraux, on peut espérer des informations sur la convergence en étudiant les valeurs propres du jacobien en les équilibres des dynamiques en s'inspirant de la théorie classique des systèmes dynamiques [Hirsch et al., 2003].

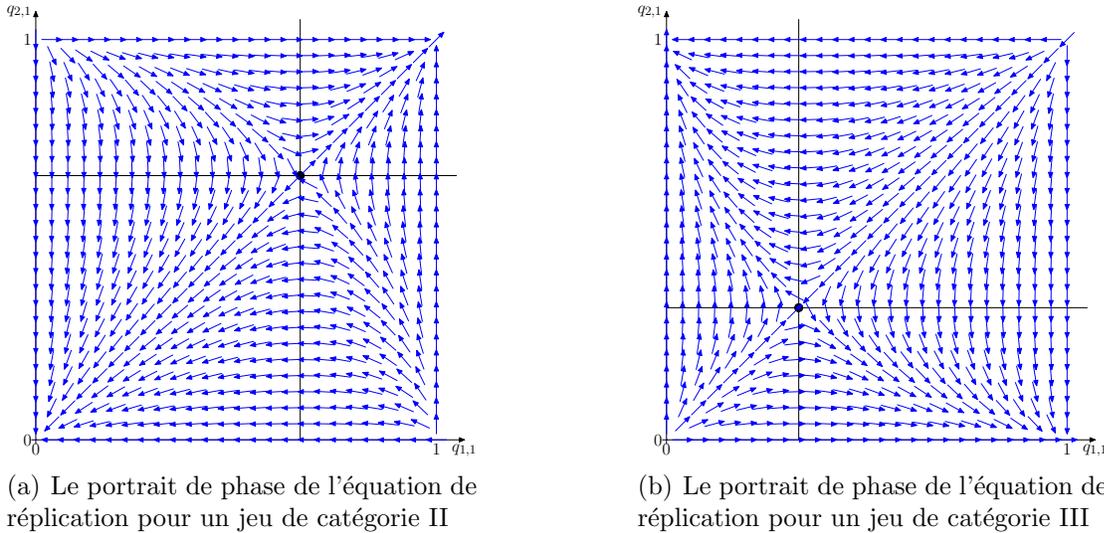


FIG. 7.2 – Le portrait de phase de l'équation de répliation pour un jeu de catégorie III. Le point correspond à l'équilibre de Nash mixte.

7.5 Algorithmes d'apprentissage pour les jeux de potentiel ordinal

Nous allons maintenant étudier la convergence pour les jeux de potentiel ordinal en utilisant l'algorithme 2.

On se focalise sur des dynamiques de *réplication perturbée* de la forme suivante :

- les fonctions σ_i sont la fonction identité ;
- les fonctions $F_i^b(r_i(t), s_i(t), q_i(t))$ sont définies de la façon suivante :

$$F_i^b(r_i(t), s_i(t), q_i(t)) = \mathcal{O}(b) + \begin{cases} \gamma(r_i(t))(e_{s_i(t)} - q_i(t)) & \text{avec probabilité } \alpha \\ b(e_{s_j} - q_i(t)) & \text{avec probabilité } 1 - \alpha, \\ & \text{où } j \in \{1, \dots, m_i\} \\ & \text{est choisit uniformément,} \end{cases} \quad (7.11)$$

où

- le paramètre α est une constante telle que $0 < \alpha < 1$;
- $\gamma : \mathbb{R} \rightarrow [0, 1]$ est une fonction *décroissante*. Les même remarques sur la fonction γ que celles faites lorsque nous avons discuté de la dynamique de répliation s'appliquent dans ce contexte.

Le terme $\mathcal{O}(b)$ désigne une perturbation qui reste de l'ordre du paramètre b , lorsque b tend vers 0. Cette perturbation peut être aléatoire (stochastique).

En utilisant les mêmes arguments que précédemment (par le calcul de $\Delta q_{i,\ell}(t)$ et de $G_i(Q) = \lim_{b \rightarrow 0} \frac{1}{bp_i} \mathbb{E}[\Delta q_{i,\ell}(t) | Q(t)]$), nous pouvons affirmer que de telles dynamiques correspondent aux dynamiques de répliation de multi-population : en d'autres termes, selon le théorème 32, il y a convergence faible vers les solutions de l'équation (7.3), qui correspondent ici aux solutions de l'équation différentielle ordinaire (7.6).

Nous insistons sur le fait que le raisonnement à venir ne peut pas s'appliquer (au moins directement) sur l'algorithme 1. En effet, la preuve du temps de convergence basée sur les arguments de martingale, ne fonctionne pas ici lorsque les joueurs réactualisent leur

stratégies mixtes simultanément.

7.5.1 Caractérisation via des fonctions de Lyapunov

Définition 35 (Jeu de Lyapunov [39]) *Un jeu est un jeu de Lyapunov s'il possède une fonction de Lyapunov, c'est-à-dire s'il existe une fonction $F : \mathcal{K} \rightarrow \mathbb{R}$ positive ou nulle, de classe \mathcal{C}^1 , telle que pour tout i, ℓ et Q , quand $G(Q) \neq 0$,*

$$\sum_{i,\ell} p_i \frac{\partial F}{\partial q_{i,\ell}}(Q) G_{i,\ell}(Q) < 0, \quad (7.12)$$

où $G_{i,\ell}$ correspond à la dynamique (7.6) sur l'espace des stratégies mixtes \mathcal{K} .

La classe des jeux de Lyapunov contient celle des jeux de potentiel ordinal et par conséquence celle des jeux de potentiel exact. De plus, on dira qu'une fonction de Lyapunov $F : \mathcal{K} \rightarrow \mathbb{R}$ est *multiaffine*, si et seulement si elle peut être définie comme un polynôme en toutes ses variables, qui est de degré 1 en chaque variable, et tel que chacun de ces monomes ne contient aucun facteur de la forme $q_{i,\ell} q_{i,\ell'}$.

Théorème 34 ([39]) *Un jeu de potentiel ordinal est un jeu de Lyapunov. De plus, les fonctions de Lyapunov sont multiaffines.*

Plus précisément, si la fonction ϕ est la fonction de potentiel du jeu de potentiel ordinal, alors la moyenne

$$F(Q) = \mathbb{E}[\phi(Q)] = \mathbb{E}[\phi(Q) \mid \text{les joueurs sélectionnent les stratégies pures selon } Q]$$

est une fonction de Lyapunov.

Définition 36 (Jeu de potentiel [Sandholm, 2001]) *Un jeu est un jeu de potentiel continu s'il existe une fonction de classe \mathcal{C}^1 $F : \mathcal{K} \rightarrow \mathbb{R}$ telle que pour tout i, ℓ et Q ,*

$$\frac{\partial F}{\partial q_{i,\ell}}(Q) = u_i(e_\ell, Q). \quad (7.13)$$

Proposition 3 ([39]) *Un jeu de potentiel continu est un jeu de Lyapunov. De plus, les fonctions de Lyapunov sont multiaffines.*

Rappelons que les jeux de potentiel exacts sont définis à la page 45.

Proposition 4 ([39]) *Un jeu de potentiel exact ayant ϕ comme fonction de potentiel est un jeu de potentiel continu avec $F(Q) = \mathbb{E}[\phi(Q)]$ comme fonction de potentiel. Et en restreignant F aux fonctions de classe \mathcal{C}^2 , le jeu de potentiel continu est aussi un jeu de potentiel exact.*

Un jeu de Lyapunov peut à priori avoir une fonction de potentiel non multiaffine, et par conséquence il peut exister des jeux de Lyapunov qui ne sont pas des jeux de potentiel ordinal.

L'intérêt de fonctions de Lyapunov est qu'elles fournissent un outil pour prouver des propriétés de convergence. Rappelons que l'ensemble limite $\omega(Q_0)$ du point Q_0 est l'ensemble des points d'accumulation des trajectoires partant du point Q_0 : en considérant une trajectoire partant de Q_0 , cet ensemble contient Q^* avec $Q^* = \lim_{n \rightarrow \infty} Q(t_n)$, pour toute suite croissante $(t_n)_{n \geq 0} \in \mathbb{R}$.

Proposition 5 ([39]) *Pour tout jeu de Lyapunov, les solutions de l'équation différentielle ordinaire (7.3) ont leur ensemble des points limites non vide, compact, composé uniquement de points stationnaires de cette dynamique. Et sur ces points limites, F est constant.*

Observons que toutes les classes précédentes sont des jeux de Lyapunov selon les dynamiques (7.6). Et par conséquence,

Corollaire 34.1 ([39]) *Dans un jeu de Lyapunov, quelque soit la condition initiale, les solutions de l'équation différentielles ordinaires (7.6) vont converger. Les points limites stables sont des équilibres de Nash.*

7.5.2 Techniques de Martingale

Il est en fait possible de dire plus de choses sur le temps de convergence pour les types de jeux précédemment cités (ordinaire, de potentiel exact, de potentiel continu, d'ordonnancement, de congestion, d'allocation de tâches) par l'utilisation de techniques de martingales.

En effet, il est possible dans ce cas de travailler directement sur l'algorithme stochastique en évitant le passage à l'équation différentielle ordinaire (7.3) et à la double limite $b \rightarrow 0, t \rightarrow \infty$.

Le lemme suivant est fondamental, et à la base du raisonnement.

Lemme 1 *Lorsque F est une fonction de Lyapunov multiaffine,*

$$\mathbb{E}[\Delta F(Q(t+1)) | Q(t)] = \sum_{i=1}^n \sum_{\ell=1}^{m_i} \frac{\partial F}{\partial q_{i,\ell}}(Q(t)) \mathbb{E}[\Delta q_{i,\ell} | Q(t)], \quad (7.14)$$

où $\Delta F(t) = F(Q(t+1)) - F(Q(t))$.

Lorsque l'on considère une dynamique de réplication en utilisant le fait que $G_i(Q) = \lim_{b \rightarrow 0} \tilde{F}_i^b(Q)$ le coté droit de l'équation (7.14) est

$$b \sum_{i=1}^n \sum_{\ell=1}^{m_i} p_i \frac{\partial F}{\partial q_{i,\ell}}(Q) G_{i,\ell}(Q) + \mathcal{O}(b^2). \quad (7.15)$$

L'espérance de cette quantité est négative par l'équation (7.12) lorsque $G(Q) \neq 0$ et b suffisamment petit.

Lorsque b est suffisamment petit, $(F(Q(t)))_t$ sera une super-martingale jusqu'à attendre le point où (7.15) est proche de 0.

De plus, pour les dynamiques de réplication, l'équation (7.15) peut se réécrire

$$-b \frac{1}{4} \sum_i p_i \sum_{\ell \neq \ell'} q_{i,\ell} q_{i,\ell'} (u_i(e_\ell, Q_{-i}) - u_i(e_{\ell'}, Q_{-i}))^2 + \mathcal{O}(b^2).$$

Lorsque l'on considère les dynamiques stochastiques perturbées en utilisant des arguments de la théorie des martingales, on peut obtenir des résultats sur la stabilité : nous notons $L(\mu)$ pour le sous ensemble des états Q tel que $F(Q) \leq \mu$.

Proposition 6 ([39]) *Soit $\lambda > 1$. Soit $Q(t_0)$ un état. Soit b un réel suffisamment petit pour que (7.14) soit négatif ou nul en dehors de $L(F(Q(0)))$. Alors $Q(t)$ sera tel que $Q(t) \in L(\lambda F(Q(t_0)))$ à jamais après un temps $t \geq t_0$ avec une probabilité plus grande que $1 - \frac{1}{\lambda}$.*

Remarquons que pour un jeu de Lyapunov, avec une fonction F de Lyapunov multiaffine, les points Q^* réalisant un minimum local F^* de F sur le compact \mathcal{K} doivent correspondre aux équilibres de Nash.

Une dynamique est dite *perturbée*, si pour tout $Q \in K$, pour n'importe quel voisinage V de Q dans sa fermeture, la probabilité que $Q(t+1) \in V$ quand $Q(t) = Q$ est positive.

Si la dynamique est perturbée, alors la chaîne de Markov associée est ergodique. Ce implique que tout voisinage sera visité avec une probabilité positive.

Ainsi, si dans un voisinage d'un tel point, nous pouvons appliquer la proposition précédente, on obtiendrait que presque sûrement, après un certain temps, $Q(t)$ sera proche d'un équilibre de Nash pour toujours avec une forte probabilité. Le défaut de cette approche est le fait que nous donnons aucune borne sur le temps pour atteindre un tel voisinage.

Il est en fait possible d'obtenir des bornes sur le temps espéré de convergence, en utilisant les arguments précédents (F est une super-martingale) :

Théorème 35 ([39]) *Considérons un jeu de Lyapunov avec une fonction F de Lyapunov multiaffine. Supposons $b = \mathcal{O}(\epsilon)$. Quel que soit l'état initial de l'algorithme, l'algorithme atteindra presque sûrement un ϵ -équilibre de Nash. De plus, le temps espéré $T(\epsilon)$ est majoré par*

$$T(\epsilon) \leq \mathcal{O}\left(\frac{F(Q(0))}{\epsilon}\right).$$

Nous croyons que ces bornes sont ténues pour des jeux de potentiel ordinal génériques. En effet, dans ce type de jeu, il n'est pas nécessaire d'avoir une relation entre les variations des utilités et de celle de la fonction de potentiel : seul le signe de la variation doit être identique.

Nous pensons qu'il existe de meilleures bornes pour les jeux de congestion sous certaines hypothèses, comme par exemple la condition du α -saut. Nous travaillons actuellement sur ce point.

Chapitre 8

Conclusion

Ce document correspond à une présentation de mes travaux récents autour de la théorie des graphes, de la complexité, et de la théorie algorithmique des jeux.

Ma recherche est essentiellement motivée par les applications liées à l’algorithmique pour les réseaux sous l’angle de l’optimisation des ressources des réseaux en garantissant certaines performances. Pour chaque problème, après sa modélisation sous forme de problèmes liés aux graphes, nous cherchons à comprendre s’il est possible d’obtenir des garanties sur l’optimisation des ressources. Ceci correspond à des travaux liés à l’algorithmique en général, centralisée ou distribuée, à la théorie des graphes et à des problèmes de complexité. Depuis quelques temps, nous nous intéressons aussi à prendre en compte la dimension économique dans le sens où les acteurs du système peuvent avoir leur propre intérêt/coût.

Le chapitre 2 illustre par exemple ce dernier point : nous nous sommes intéressées au problème de la construction d’arbre de plus court chemin en présence de partenaires économiques ([10], [42]). Cette étude est une extension d’un problème de routage inter-domaine. Lorsque deux types de coûts concurrents existent, le problème devient NP-complet. En revanche pour des instances ayant une certaine propriété (ne contenant pas de roues avec conflits), l’algorithme réparti glouton calcule un tel arbre et en plus de façon auto-stabilisante.

Tout au long du document, nous avons discuté de quelques suites possibles aux différents travaux de recherche décrits.

Nous reprenons ici quelques éléments de discussions. Cette discussion peut se décliner autour de deux principaux axes

1. Compromis et approximations ;
2. Modèles et algorithmes en présence de partenaires économiques.

Compromis et approximations.

Depuis plusieurs années, un aspect de mon travail concerne l’étude de la frontière entre la NP-complétude et la polynomialité pour différents problèmes liés aux réseaux et à l’algorithmique sur les graphes.

Le chapitre 3 se consacre à nos résultats relatifs à des algorithmes d’approximation. Ces résultats portent sur la numérotation de sommets sur les graphes d’intervalle, en liens avec la bio-informatique [27] et sur la gestion des ressources dans le contexte de la redistribution de données [14] et dans l’ordonnancement de messages dans les réseaux

optiques (essentiellement [7], [20], [21], [33]). Dans ces travaux, nous avons caractérisé pour quels types d'instances le problème devient polynomial.

Le chapitre 4 correspond à une discussion sur les colorations de graphes non-classiques comme la b -coloration [6] et la f -coloration [34]. Tout au long de ce chapitre, nous étudions la frontière entre la polynomialité et la NP-complétude pour ces problèmes. Cela inclue par exemple de décider si un graphe possède une telle coloration s'il possède des propriétés sur l'ensemble de ces types de coloration. Cette étude se fait de façon systématique en fonction de la structure des instances (graphe biparti ou pas, avec un degré maximal borné ou non, connaissance ou pas de l'existence d'une telle coloration). Cette étude s'est étendue vers l'inapproximabilité des problèmes d'optimisation correspondant.

Il ressort fortement de tous ces travaux que la recherche de solutions exactes est souvent impossible. Il nous semble donc vital de poursuivre nos investigations actuelles sur des outils qui permettent de contourner cet obstacle en s'orientant vers la recherche de compromis ou d'approximations.

Comprendre la frontière entre la polynomialité et l'intractabilité permet de classer les problèmes et aide à construire des méthodes adaptées pour les résoudre ou les approximer, ou à comprendre les paramètres pertinents. Un exemple connu est de considérer les graphes à largeur d'arbre bornée. En effet, de nombreux problèmes NP-complets sur les graphes généraux s'avèrent polynomiaux lorsque l'on se restreint à des largeurs d'arbres bornées : par exemple lorsque les problèmes peuvent se transcrire en logique monadique du second ordre. Le même type de phénomène se produit pour des types de graphes classiques comme les graphes d'intervalles : de nombreux problèmes NP-complets deviennent polynomiaux, même si des problèmes connus et intensivement étudiés comme la largeur de coupe ne sont pas encore résolus.

Même si un problème est polynomial, l'algorithme calculant la solution optimale peut être coûteux en termes d'opérations en pratique. La résolution par des algorithmes simples adaptés est nécessaire. Pour illustrer ces dires, les entités peuvent être limitées en termes de puissance de calcul (modèle des protocoles de populations) ou les entités peuvent avoir une ressource critique limitée (par exemple l'énergie pour les noeuds dans les réseaux sans fils, le temps pour les réseaux optiques). Avoir un algorithme simple et implémentable implique souvent une dégradation de la qualité de la solution. Cela implique de trouver un algorithme d'approximation correspondant à un compromis entre la simplicité et la qualité de la solution obtenue.

Poursuivre nos investigations sur l'utilisation de la théorie de l'approximation nous paraît nécessaire pour l'optimisation des ressources des réseaux de télécommunications en vue de l'émergence des nouveaux services gourmands en qualité de service.

Cela peut se faire autour de nombreux problèmes. Par exemple, de manière anecdotique, le mécanisme de transport de données MPLS (*Multiprotocol Label Switching* en anglais) correspond à une technique utilisant la commutation de paquets. Cette technique qui garantit une réservation de la bande passante met au goût du jour des travaux de ma thèse qu'il faut étendre à cette problématique particulière.

Plus généralement, comme il n'existe pas dans ce contexte de recette universelle pour résoudre tous les problèmes, nous proposons de continuer à travailler sur les problèmes d'optimisation dans ce contexte au fil des rencontres scientifiques ou via des contrats. Par exemple, nous sommes impliquées dans le contrat européen ETICS qui débute. Notre part du travail correspond à la réservation de ressources à deux niveaux basée sur des techniques de résolution du problème du sac-à-dos multidimensionnel. En outre, chaque

niveau du réseau correspond à des entités ayant leur propre intérêt et ce qui complique la modélisation puisqu'il faut introduire la notion de concurrence.

Modèles et algorithmes en présence de partenaires économiques

L'accroissement de la taille des réseaux actuels, fait que le comportement global émergeant des réseaux n'est plus réellement celui prévu par l'algorithmique classique. En effet, les différents partenaires impliqués dans les réseaux sont des sociétés commerciales en nombre croissant qui ont des intérêts économiques propres et divergents. Aussi, il est de moins en moins réaliste de supposer que chacun agit uniquement dans l'intérêt des performances globales de l'ensemble, et l'algorithmique classique répartie, qui présuppose généralement que chacun effectue la tâche qui lui incombe dans l'intérêt général, est de plus en plus mise en défaut.

La prise en compte de la concurrence des acteurs dans un système introduit des difficultés supplémentaires : en particulier, sur la modélisation des systèmes, la construction de solutions satisfaisantes, et sur l'interprétation des solutions. Par exemple, modéliser un jeu correspond souvent à quantifier l'intérêt d'un joueur par le biais d'une fonction d'utilité ou de coût et aussi celle du coût social du système. Cette modélisation est souvent assez techniquement difficile à rendre réaliste. Comment quantifier précisément les préférences lorsqu'elles peuvent être subjectives ?

Le chapitre 5 est un survol des jeux principaux étudiés dans la littérature informatique comme par exemple les jeux basés sur le placement de tâches, ou les jeux de congestion.

A ce jour, les différents travaux en théorie algorithmique des jeux se focalisent principalement sur :

1. la caractérisation des équilibres de Nash purs (ou stables) s'ils en existent ;
2. le calcul de la qualité des équilibres par rapport au coût social (prix de l'anarchie, de la stabilité) ;
3. la construction de tels équilibres de Nash ;
4. l'évaluation des équilibres ainsi calculés en considérant le coût social.

Les deux derniers points peuvent se réaliser de manière soit centralisée ou soit répartie. Une solution centralisée implique souvent un algorithme calculant un équilibre de Nash ayant une vision complète du jeu. L'approche répartie fait souvent l'hypothèse que chaque joueur choisit dynamiquement sa propre stratégie par apprentissage, en supposant que le jeu est réitéré/répété.

En fait, il existe plusieurs façons d'introduire des aspects dynamiques en théorie des jeux. Les chapitres 6 et 7 illustrent deux approches.

Le chapitre 6 porte sur l'étude la répétition d'un même jeu, avec des joueurs qui évoluent selon certains comportements. Nos contributions personnelles relatives à ce chapitre concerne d'une part le calcul de stratégies comportementales pour obtenir des stratégies Pareto optimales dans le contexte du routage inter-domaine [18]. Nous avons étudié le comportement des joueurs fictifs dans ce contexte. D'autre part, nous avons obtenu une caractérisation partielle de la puissance des protocoles de population utilisant des dynamiques de jeux myopes [24].

Le chapitre 7 porte sur l'apprentissage d'équilibres de Nash, selon une classe de dynamiques stochastiques d'apprentissage particulière : le choix stochastique du joueur i

dépend d'une fonction des comportements des autres joueurs et du passé du joueur i . Des algorithmes simples d'apprentissage sur des jeux de potentiel ordinal ont été décrits. Les principaux résultats de convergence d'équilibres purs se basent essentiellement sur la décroissance (en moyenne) de la fonction de potentiel.

Notre objectif est de comprendre ce qu'il est possible de garantir sur des algorithmes en présence de partenaires aux intérêts propres.

Pour cela, il faut, dans le contexte de l'algorithmique répartie :

1. comprendre les notions d'équilibres ou de stabilités pertinentes ;
2. comprendre s'il existe de tels équilibres ;
3. comprendre s'il existe des algorithmes qui les construisent de façon répartie ;
4. comprendre si ces équilibres ainsi construits peuvent être évalués par rapport à un coût social ;
5. comprendre les limites des modèles et méthodes associés.

Nous pensons intimement, que toute l'algorithmique répartie est à repenser selon ces aspects.

La théorie des jeux est un outil naturel. Il faut toutefois avoir conscience que les solutions de la théorie des jeux sont généralement pas directement utilisables. En fait, les développements historiques de la théorie des jeux en mathématiques et économie, ont souvent oublié la nécessité de solutions constructives, efficaces et implémentables. C'est pour cette raison que l'on distingue maintenant souvent la théorie des jeux de la théorie algorithmique des jeux. Par exemple, en théorie des jeux, le théorème de Nash affirme que tout jeu fini possède un équilibre de Nash. Ce théorème garantit l'existence mais n'est pas constructif, et ne dit pas comment ils peuvent se calculer. Des travaux récents de la théorie algorithmique des jeux discutent de la difficulté de les calculer.

Une orientation possible de nos travaux est de transformer (si c'est possible) les problèmes classiques de l'algorithmique répartie (construction d'arbre couvrant de poids minimal, construction de couplages, construction d'ensemble stable dans un graphe,...) en prenant en compte ces aspects.

Par exemple, en poursuivant les explorations sur la construction d'arbres de plus court chemin en liens avec le chapitre 2. Dans le chapitre 2, nous avons extrait une famille d'instances qui possèdent des équilibres de Nash purs. La preuve de ce résultat semble posséder des fortes similitudes avec certaines preuves dont l'esprit est de définir un ordre sur les différentes configurations du système et de construire (à partir d'un algorithme) une suite de configurations décroissante en fonction cet ordre. Si un tel ordre peut exister pour la construction répartie d'arbres, cela impliquerait que ce jeu correspond à un jeu de potentiel ordinal. Ce qui impliquerait qu'il existe d'autres algorithmes répartis (comme par exemple ceux décrit dans les chapitres 6 et 7) convergents vers des équilibres de Nash purs.

Il est à noter qu'en algorithmique classique, comparer deux algorithmes peut se faire en évaluant le nombre d'opérations, de messages, ... En théorie algorithmique des jeux, apprendre un équilibre de façon répartie permet de prédire si un système va converger vers une situation stable. Savoir s'il existe plusieurs algorithmes répartis qui le permettent est déjà une information en soit pour comprendre mieux le système et pour pouvoir classifier les comportements des acteurs en fonction de la convergence ou non du système.

Un autre axe est de comprendre quels sont les algorithmes d'apprentissage d'équilibres de Nash mixtes et les propriétés des jeux pour lesquels ils convergent. A ma connaissance,

la majorité des algorithmes d'apprentissage (à l'exception des joueurs fictifs) en théorie algorithmique des jeux calculent en fait uniquement des équilibres de Nash purs. En particulier, la majorité de ce type de travaux porte sur des jeux où il en existe toujours au moins un équilibre pur. Des questions naturelles sont de comprendre s'il existe des algorithmes d'apprentissage répartis permettant de calculer un équilibre de Nash mixte et si dans ce contexte, calculer un tel équilibre a du sens. Par exemple, dans le problème de la construction d'arbre de plus court chemin du chapitre 2, la notion d'équilibre de Nash mixte n'est pas réellement pertinente puisqu'à chaque étape, le chemin change et cela implique un routage instable.

Un autre axe est de comprendre si les résultats sont robustes au fait que les acteurs puissent ne pas avoir le même comportement. Dans les différents types d'algorithmes d'apprentissage d'équilibres, bien que les acteurs aient des intérêts différents, on suppose qu'ils utilisent tous le même algorithme de comportement. Uniquement sous ces hypothèses, le système est prouvé converger vers un équilibre. Cette hypothèse est forte dans le sens où maintenant beaucoup d'acteurs peuvent intervenir dans un système en pratique. Il est iréaliste de supposer que tout le monde exécute le même programme (aie le même comportement). Il peut aussi plus généralement être naturel de supposer que les acteurs puissent avoir un comportement byzantin (dans le sens imprévisible ou nuisible). Il faut comprendre quels sont les comportements comptatibles pour atteindre une situation stable, quelle est la robustesse des algorithmes en présence d'acteurs byzantins.

Annexe A

Problème d'optimisation locale

Cette annexe se consacre particulièrement à introduire les notions de base de la complexité portant sur les problèmes de construction de solutions.

Considérons la problème de décision : étant donnée une instance x décider s'il existe une solution y tel que la relation polynomiale $R(x, y)$. Le problème de recherche associé à ce problème est le suivant : étant donnée une instance x , trouver une solution y tel que $R(x, y)$ si elle existe, répondre **non** sinon.

De plus, même s'il existe toujours une solution liée pour le problème de la décision, la calculer reste néanmoins compliqué.

Une relation polynomiale R est *totale* si pour n'importe quel x , il existe un y tel que $R(x, y)$. Cela implique que le problème de décision pour une relation polynomiale totale n'a que des instances positives. Malgré cette connaissance, il est parfois difficile de construire une telle solution.

La *classe TFNP* est la classe des problèmes de recherche associés à un problème de décision caractérisés par une relation totale.

Nous allons considérer deux sous-classes de TFNP : la classe *PLS* et la classe *PPAD*.

A.1 La classe PLS

Définition 37 (La classe PLS) *Un problème d'optimisation Π dans la classe PLS (due à [Johnson et al., 1988]) (en anglais Polynomial Local Search), est caractérisé par l'ensemble de ses instances. Pour chaque instance x ,*

- *l'ensemble des solutions $SOL(x)$ est reconnaissable en temps polynomial;*
- *chaque solution $y \in SOL_\pi(x)$ a sa valeur $m_\Pi(x, y)$ calculable en temps polynomial;*
- *il est possible de trouver en temps polynomial une solution $y_0 \in SOL_\pi(x)$;*
- *chaque solution $y \in SOL_\pi(x)$ a un voisinage $V_x(y)$ de solutions possibles qui est reconnaissable en temps polynomial;*
- *pour chaque solution $y \in SOL_\pi(x)$, savoir si y est un optimum local ou produire une solution du voisinage de y peut se calculer en temps polynomial.*

Pour une instance donnée x , l'objectif est de trouver un optimum local par rapport à un voisinage fixé.

Un exemple de problème PLS-complet donné dans ce document (page 46) est de construire un équilibre de Nash dans un jeu de congestion [Johnson et al., 1988]. En effet, cela revient à calculer un minimum local pour la fonction de potentiel associée à ce jeu.

A.2 La classe PPAD

La classe PPAD défini dans [Papadimitriou, 1994] est une sous-classe de TFNP de problèmes qui sont réductibles en temps polynomial au problème *END-OF-THE-LINE* :

- étant donné deux circuits S et P avec n bits en entrée et en sortie tels que $P(0^n) = 0^n \neq S(0^n)$
- déterminer une entrée $x \in \{0, 1\}^n$ tel que $P(S(x)) \neq x$ ou $S(P(x)) \neq x \neq 0^n$.

De façon duale, le problème *END-OF-THE-LINE* crée un graphe dirigé avec $\{0, 1\}^n$ comme ensemble de sommets et ayant un arc allant x vers y pour tout couple de sommets si $y = S(x)$ et $P(y) = x$. Ce graphe a au plus un degré sortant ou entrant de 1 pour chaque sommet. De plus il a au moins une ou plusieurs sources, puisque le sommet 0^n est une source. Il a donc aussi un puits. Ici, on recherche soit une source autre que 0^n ou alors un puits.

Il est à noter que ce graphe a un nombre exponentiel de sommets. Le graphe G est défini par une fonction polynomiale qui calcule soit le prédécesseur ou le successeur de n'importe quel sommet du graphe.

Cette classe de problèmes de recherche est utilisée pour évaluer la complexité de calcul d'un équilibre de Nash (voir page 42).

Index

- équilibre de Nash
 - mixte, 41
- action, 54
- algorithme
 - approximable, 23
- Algorithme
 - auto-stabilisant, 16
 - faiblement auto-stabilisant, 18
- approximation
 - à un facteur près, 22
- classe
 - APX, 23
 - NPO, 22
 - PLS, 87
 - TFNP, 87
- coloration, 31
 - a -coloration, 32
 - b -coloration, 33
 - f -coloration, 37
- ϵ -équilibre de Nash
 - pur, 50
- équilibre de Nash
 - pur, 19
- état
 - Pareto optimal, 50
- Fonction
 - d'utilité, 18
 - de coût, 18
- fonction de potentiel
 - ordinal, 44
- forme
 - normal du jeu, 19
- graphe
 - b -continu, 36
 - d'intervalle, 24
 - f -continu, 38
- histoire, 54
 - de longueur t , 54
- Jeu, 18
 - symétrique, 42
- jeu
 - d'ordonnancement, 46
 - à accès restreint, 46
 - de congestion, 45
 - pondérés, 46
 - de Lyapunov, 77
 - escompté, 55
 - infiniment répété, 55
- jeu de potentiel, 45
 - exact, 45
 - ordinal, 44
- joueur
 - fictif, 58
- meilleure réponse, 40
- nombre
 - a -chromatique, 32
 - chromatique, 31
 - f -achromatique, 37
 - f -chromatique, 37
- numérotation sommet
 - reo, 24
- politique
 - aléatoire, 48
 - LPT, 48
 - SPT, 48
- prix
 - de l'anarchie, 51
 - de la stabilité, 51
- Problème d'optimisation, 21
- Profil
 - pur, 18
- profil
 - de stratégies, 40

- réduction
 - préservation d'intervalle, 23
- sommet
 - b -chromatique, 33
- spectre
 - b -spectre, 34
 - f -spectre, 38
- stratégie
 - de comportement, 54
 - mixte, 39
- Stratégie
 - pure, 18
- support d'une stratégie mixte, 40
- taux d'actualisation, 55

Bibliographie

- [Adar and Huberman, 2000] Adar, E. and Huberman, B. A. (2000). Free riding on gnutella. *First Monday*, 5(10). http://firstmonday.org/issues/issue5_10/adar/index.html.
- [Afrati et al., 2002] Afrati, F., Aslanidis, T., Bampis, E., and Milis, I. (2002). Scheduling in switching networks with set-up delays. In *AlgoTel 2002*, Mèze, France.
- [Afrati et al., 2005] Afrati, F. N., Aslanidis, T., Bampis, E., and Milis, I. (2005). Scheduling in switching networks with set-up delays. *J. Comb. Optim.*, 9(1) :49–57.
- [Agussurja and Lau, 2007] Agussurja, L. and Lau, H. C. (2007). The price of stability in selfish scheduling games. In *IAT '07 : Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 305–311, Washington, DC, USA. IEEE Computer Society.
- [Ahronovitz et al., 2006] Ahronovitz, E., König, J.-C., and Saad, C. (2006). A distributed method for dynamic resolution of bgp oscillations. In *20th International Parallel and Distributed Processing Symposium (IPDPS 2006)*.
- [Aland et al., 2006] Aland, S., Dumrauf, D., Gairing, M., Monien, B., and Florian, S. (2006). Exact price of anarchy for polynomial congestion games. In *Annual Symposium on Theoretical Aspects of Computer Science (STAC)*, volume 3884 of *Lecture Notes in Computer Science*, pages 218–229. Springer.
- [Angel et al., 2006] Angel, E., Bampis, E., and Pascual, F. (2006). The price of approximate stability for scheduling selfish tasks on two links. In *Parallel Processing, 12th International Euro-Par Conference (Euro-Par)*, volume 4128 of *Lecture Notes in Computer Science*, pages 157–166. Springer.
- [Angluin et al., 2004] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M. J., and Peralta, R. (2004). Computation in networks of passively mobile finite-state sensors. In *Twenty-Third ACM Symposium on Principles of Distributed Computing*, pages 290–299. ACM Press.
- [Angluin et al., 2006] Angluin, D., Aspnes, J., and Eisenstat, D. (2006). Stably computable predicates are semilinear. In *PODC '06 : Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA. ACM Press.
- [Angluin et al., 2007] Angluin, D., Aspnes, J., Eisenstat, D., and Ruppert, E. (2007). The computational power of population protocols. *Distributed Computing*, 20(4) :279–304.
- [Anshelevich et al., 2004] Anshelevich, Dasgupta, Kleinberg, Tardos, Wexler, and Roughgarden (2004). The price of stability for network design with fair cost allocation. In *FOCS : IEEE Symposium on Foundations of Computer Science (FOCS)*.

- [Aspnes et al., 1997] Aspnes, J., Azar, Y., Fiat, A., Plotkin, S. A., and Waarts, O. (1997). On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3) :486–504.
- [Aspnes and Ruppert, 2007] Aspnes, J. and Ruppert, E. (2007). An introduction to population protocols. In *Bulletin of the EATCS*, volume 93, pages 106–125.
- [Aupy, 2009] Aupy, G. (2009). A propos des protocoles infinis. Master’s thesis, Ecole Normale Supérieure de Lyon.
- [Ausiello et al., 1999] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Spaccamela, A. M., and Protasi, M. (1999). *Complexity and Approximation*. Springer Verlag.
- [Awerbuch et al., 2006] Awerbuch, B., Azar, Y., Richter, Y., and Tsur, D. (2006). Tradeoffs in worst-case equilibria. *Theoretical Computer Science*, 361(2-3) :200–209.
- [Axelrod, 1984] Axelrod, R. M. (1984). *The Evolution of Cooperation*. Basic Books.
- [Azar et al., 1995] Azar, Y., Naor, J. S., and Rom, R. (1995). The competitiveness of on-line assignments. *J. Algorithms*, 18(2) :221–237.
- [Benaïm, 1999] Benaïm, M. (1999). Dynamics of stochastic approximation algorithms. *Séminaire de Probabilités XXXIII. Lecture Notes in Math*, 1709 :1–68.
- [Berenbrink et al., 2006] Berenbrink, P., Friedetzky, T., Goldberg, L. A., Goldberg, P., Hu, Z., and Martin, R. (2006). Distributed selfish load balancing. In *SODA ’06 : Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 354–363, New York, NY, USA. ACM.
- [Berenbrink and Schulte, 2007] Berenbrink, P. and Schulte, O. (2007). Evolutionary equilibrium in bayesian routing games : Specialization and niche formation. In *ESA*, pages 29–40.
- [Berge, 1987] Berge, C. (1987). *Graphes*. Gauthier Villars.
- [Berger, 2003] Berger, U. (2003). Fictitious play in 2xn games. Game Theory and Information 0303009, EconWPA.
- [Berger, 2007] Berger, U. (2007). Brown’s original fictitious play. *Journal of Economic Theory*.
- [Bilò et al., 2008] Bilò, V., Fanelli, A., Flammini, M., and Moscardelli, L. (2008). Graphical congestion games with linear latencies. In *SPAA ’08 : Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures*, pages 194–196, New York, NY, USA. ACM.
- [Binmore, 1999] Binmore, K. (1999). *Jeux et Théorie des jeux*. DeBoeck Université, Paris-Bruxelles. Translated from ‘Fun and Games : a text on game theory’ by Francis Bismans and Eulalia Damaso.
- [Blum, 1994] Blum, A. (1994). New approximation algorithms for graph coloring. *J. ACM*, 41(3) :470–516.
- [Bodlaender, 1989] Bodlaender, H. L. (1989). Achromatic number is np-complete for cographs and interval graphs. *Information Processing Letters*, 31(3) :135–138.
- [Bongiovanni et al., 1981] Bongiovanni, G., Coppersmith, D., and Wong, C. K. (1981). An Optimum Time Slot Assignment Algorithm for an SS/TDMA System with Variable Number of Transponders. *IEEE Trans. Comm.*, 29(5) :721–726.

- [Bonomo et al., 2009] Bonomo, F., Duràn, G., Maffray, F., Marenco, J., and Valencia-Pabon, M. (2009). On the b-coloring of cographs and p_4 -sparse graphs. *Graphs and Combinatorics*, 25(2) :153–167.
- [Boussaton, 2009] Boussaton, O. (2009). *Routage et théorie des jeux*. PhD thesis, Université de Nancy I. En rédaction.
- [Brooks, 1941] Brooks, R. L. (1941). On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37(02) :194–197.
- [Brown, 1951] Brown, G. W. (1951). Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*.
- [Brucker, 2001] Brucker, P. (2001). *Scheduling Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Cairnie and Keith, 1997] Cairnie, N. and Keith, E. (1997). Some results on the achromatic number. *Journal of Graph Theory*, 26 :129–136.
- [Chekuri et al., 2007] Chekuri, C., Mydlarz, M., and Shepherd, F. B. (2007). Multicommodity demand flow in a tree and packing integer programs. *ACM Trans. Algorithms*, 3(3) :27.
- [Chen and Mok, 2004] Chen, D. and Mok, A. (2004). *Handbook of scheduling : algorithms, models, and performance analysis*, chapter The Pinwheel : A real-Times Scheduling Problem. CRC Press.
- [Chen and Deng, 2005] Chen, X. and Deng, X. (2005). Settling the complexity of 2-player nash-equilibrium. *Electronic Colloquium on Computational Complexity*, 140.
- [Chien and Sinclair, 2007] Chien, S. and Sinclair, A. (2007). Convergence to approximate nash equilibria in congestion games. In *SODA*, pages 169–178.
- [Chien and Sinclair, 2009] Chien, S. and Sinclair, A. (2009). Strong and pareto price of anarchy in congestion games. In *Automata, Languages and Programming, 36th International Colloquium, (ICALP)*, volume 5555 of *Lecture Notes in Computer Science*, pages 279–291.
- [Choi et al., 1996] Choi, H., Choi, H.-A., and Azizoglu, M. (1996). Efficient Scheduling of Transmissions in Optical Broadcast Networks. *IEEE/ACM Trans. Net.*, 4(6) :913–920.
- [Christodoulou and Koutsoupias, 2005a] Christodoulou, G. and Koutsoupias, E. (2005a). On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Annual European Symposium (ESA)*, volume 3669 of *Lecture Notes in Computer Science*, pages 59–70. Springer.
- [Christodoulou and Koutsoupias, 2005b] Christodoulou, G. and Koutsoupias, E. (2005b). The price of anarchy of finite congestion games. In *STOC '05 : Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 67–73, New York, NY, USA. ACM.
- [Christodoulou et al., 2004] Christodoulou, G., Koutsoupias, E., and Nanavati, A. (2004). Coordination mechanisms. In *Automata, Languages and Programming : 31st International Colloquium*, volume 3142 of *Lecture Notes in Computer Science*, pages 345–357. Springer.
- [Chung, 1988] Chung, F. R. K. (1988). Selected topics in graph theory. In *Selected Topics in graph theory*, pages 151–168. Academic Press.

- [Cominetti et al., 2008] Cominetti, R., Melo, E., and Sorin, S. (2008). A payoff-based learning procedure and its application to traffic games. *Games and Economic Behavior*.
- [Conitzer and Sandholm, 2008] Conitzer, V. and Sandholm, T. (2008). New complexity results about nash equilibria. *Games and Economic Behavior*.
- [Cormen et al., 2001] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms, Second Edition*. The MIT Press.
- [Corteel et al., 2005] Corteel, S., Valencia-Pabon, M., and Vera, J. C. (2005). On approximating the b-chromatic number. *Discrete Applied Mathematics*, 146(1) :106–110.
- [Coucheney et al., 2009] Coucheney, P., Touati, C., and Gaujal, B. (2009). Fair and efficient user-network association algorithm for multi-technology wireless networks. In *Proc. of the 28th conference on Computer Communications miniconference (INFOCOM)*.
- [Crescenzi et al., 2001] Crescenzi, P., Xiaotie, D., and Papadimitriou, C. H. (2001). On Approximating a Scheduling Problem. *Journal of Combinatorial Optimization*, 5 :287–297.
- [Czumaj and Vöcking, 2002] Czumaj, A. and Vöcking, B. (2002). Tight bounds for worst-case equilibria. In *Proceedings of the 13th Annual ACM-SIAM Symposium On Discrete Mathematics (SODA-02)*, pages 413–420, New York. ACM Press.
- [Dasgupta et al., 2006] Dasgupta, A., Ghosh, S., and Tixeuil, S. (2006). Selfish stabilization. In *8th International Symposium Stabilization, Safety, and Security of Distributed Systems (SSS)*, volume 4280 of *Lecture Notes in Computer Science*, pages 231–243. Springer.
- [Desprez et al., 1998] Desprez, F., Dongarra, J., Petitet, A., Randriamaro, C., and Robert, Y. (1998). Scheduling Block-Cyclic Array Redistribution. *IEEE TPDS*, 9(2) :192–205.
- [Díaz et al., 2002] Díaz, J., Petit, J., and Serna, M. J. (2002). A survey of graph layout problems. *ACM Computing Survey*, 34(3) :313–356.
- [Díaz et al., 2001] Díaz, J., Petit, J., Serna, M. J., and Trevisan, L. (2001). Approximating layout problems on random graphs. *Discrete Mathematics*, 235(1–3) :245–253.
- [Dijkstra, 1971] Dijkstra, E. W. (1971). *A short introduction to the art of programming*.
- [Dobson, 1984] Dobson, G. (1984). Scheduling independent tasks on uniform processors. *SIAM Journal on Computing*, 13(4) :705–716.
- [Dunbar et al., 2000] Dunbar, J. E., Hedetniemi, S. M., T., H. S., Jacobs, D. P., Knisely, D. J., Laskar, R., and Rall, D. F. (2000). Fall colorings of graphs. *J. of Combin. Math. and Combin. Comput.*, 33 :257–273.
- [Dürr and Thang, 2009] Dürr, C. and Thang, N. K. (2009). Non-clairvoyant scheduling games. In *2nd International Symposium on Algorithmic Game Theory (SAGT 2009)*, Lecture Notes in Computer Science. Springer.
- [Erlebach et al., 2003] Erlebach, T., Pagourtzis, A., Potika, K., and Stefanakos, S. (2003). Resource allocation problems in multifiber wdm tree networks. In *Graph-Theoretic Concepts in Computer Science, 29th International Workshop*, volume 2880 of *Lecture Notes in Computer Science*, pages 218–229.

- [Even et al., 1976] Even, S., Itai, A., and Shamir, A. (1976). On the complexity of timetable and multicommodity flow problem. *SIAM J. Comput.*, 5 :691–703.
- [Even and Shiloach, 1975] Even, S. and Shiloach, Y. (1975). Np-completeness of several arrangement problems. Technical report, Computer Science Dept., The Technion, Haifa, Israel.
- [Even-Dar et al., 2003] Even-Dar, E., Kesselman, A., and Mansour, Y. (2003). Convergence time to Nash equilibria. *30th International Conference on Automata, Languages and Programming (ICALP)*, pages 502–513.
- [Even-Dar et al., 2007] Even-Dar, E., Kesselman, A., and Mansour, Y. (2007). Convergence time to Nash equilibrium in load balancing. *ACM Transactions on Algorithms*, 3(3).
- [Fabrikant et al., 2004] Fabrikant, A., Papadimitriou, C., and Talwar, K. (2004). The complexity of pure nash equilibria. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 604–612. ACM Press.
- [Faik, 2005] Faik, T. (2005). *La b-continuité des b-colorations : complexité, propriétés structurelles et algorithmes*. PhD thesis, Université Paris XI.
- [Farach-Colton et al., 2004] Farach-Colton, M., Huang, Y., and Woolford, J. L. L. (2004). Discovering temporal relations in molecular pathways using protein-protein interactions. In *RECOMB '04 : Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 150–156, New York, NY, USA. ACM.
- [Feige, 2000] Feige, U. (2000). Approximating the bandwidth via volume respecting embeddings. *J. Comput. Syst. Sci.*, 60(3) :510–539.
- [Feige and Kilian, 1998] Feige, U. and Kilian, J. (1998). Zero knowledge and the chromatic number. *J. Comput. Syst. Sci.*, 57(2) :187–199.
- [Feldmann et al., 2003] Feldmann, Gairing, Lucking, Monien, and Rode (2003). Selfish routing in non-cooperative networks : A survey. In *MFCS : Symposium on Mathematical Foundations of Computer Science*.
- [Finn and Horowitz, 1979] Finn, G. and Horowitz, E. (1979). A linear time approximation algorithm for multiprocessor scheduling. *BIT Numerical Mathematics*.
- [Fotakis et al., 2005] Fotakis, D., Kontogiannis, S., and Spirakis, P. (2005). Selfish unsplittable flows. *Theoretical Computer Science*, 348(2) :226–239.
- [Fotakis et al., 2002] Fotakis, D., Spyros Kontogiannis, E., Mavronicolas, M., and Spirakis, P. (2002). The Structure and Complexity of Nash Equilibria for a Selfish Routing Game. *Automata, Languages and Programming : 29th International Colloquium, ICALP 2002, Málaga, Spain, July 8-13, 2002 : Proceedings*.
- [Friesen, 1987] Friesen, D. K. (1987). Tighter bounds for lpt scheduling on uniform processors. *SIAM J. Comput.*, 16(3) :554–560.
- [Fudenberg and Levine, 1996] Fudenberg, D. and Levine, D. K. (1996). *The Theory of Learning in Games*. MIT Press. available at <http://ideas.repec.org/p/cla/levarc/624.html>.
- [Gairing et al., 2004] Gairing, M., Lücking, T., Mavronicolas, M., and Monien, B. (2004). Computing nash equilibria for scheduling on restricted parallel links. In *Proceedings of*

- the thirty-sixth annual ACM symposium on Theory of computing*, pages 613–622. ACM Press.
- [Garey and Johnson, 1979] Garey, M. and Johnson, D. (1979). *Computers and Intractability : A Guide to the theory of NP-Completeness*. W.H Freeman and co.
- [Gilboa and Zemel, 1989] Gilboa, I. and Zemel, E. (1989). Nash and correlated equilibria : Some complexity results,. *Games and Economic Behavior*.
- [Goldberg and Klipker, 1976] Goldberg, M. A. and Klipker, I. A. (1976). Minimal placing of trees on a line. Technical report, Technical report, Physico-Technical Institute of Low Temperatures, Academy of Sciences of Ukrainian SSR, USSR.
- [Goldberg, 2004] Goldberg, P. W. (2004). Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In *PODC '04 : Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 131–140, New York, NY, USA. ACM.
- [Gopal and Wong, 1985] Gopal, I. and Wong, C. (1985). Minimizing the Number of Switching in an SS/TDMA System. *IEEE Trans. Comm.*
- [Gopal et al., 1982] Gopal, I. S., Bongiovanni, G., Bonuccelli, M. A., Tang, D. T., and Wong, C. K. (1982). An Optimal Switching Algorithm for Multibeam Satellite Systems with Variable Bandwidth Beams. *IEEE Trans. Comm.*, COM-30(11) :2475–2481.
- [Graham, 1966] Graham, R. L. (1966). Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*.
- [Graham, 1969] Graham, R. L. (1969). Bounds on multiprocessing timing anomalies. *SIAM Journal Applied Mathematics*, 17 :416–429.
- [Griffin et al., 2002] Griffin, T., Shepherd, F. B., and Wilfong, G. T. (2002). The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10(2) :232–243.
- [Griffin and Wilfong, 2000] Griffin, T. and Wilfong, G. T. (2000). A safe path vector protocol. In *INFOCOM*, pages 490–499.
- [Harary, 1983] Harary, F. (1983). Maximum versus minimum invariants for graphs. *Journal of graph theory*.
- [Harary et al., 1967] Harary, F., Hedetniemi, S. T., and Prins, G. C. E. (1967). An interpolation theorem for graphical homomorphisms. *Portugaliae Mathematica*, 26 :453–462.
- [Håstad, 1997] Håstad, J. (1997). Clique is hard to approximate within $n^{1-\epsilon}$. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(38).
- [Heggernes et al., 2008] Heggernes, P., Lokshtanov, D., Mihai, R., and Papadopoulos, C. (2008). Cutwidth of split graphs, threshold graphs, and proper interval graphs. In *34rd International Workshop of Graph-Theoretic Concepts in Computer Science*.
- [Hirsch et al., 1989] Hirsch, M. D., Papadimitriou, C. H., and Vavasis, S. A. (1989). Exponential lower bounds for finding brouwer fix points. *J. Complexity*, 5(4) :379–416.
- [Hirsch et al., 2003] Hirsch, M. W., Smale, S., and Devaney, R. (2003). *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press.
- [Hofbauer and Sigmund, 2003] Hofbauer, J. and Sigmund, K. (2003). Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 4 :479–519.
- [Huitema, 2001] Huitema, C. (2001). *Routing in the Internet*. Prentice Hall.

- [Ibrahim and Matta, 2004] Ibrahim, S. Y. and Matta, I. (2004). A randomized solution to bgp divergence. In *in Proceedings of the 2nd IASTED International Conference on Communication and Computer Networks (CCN'04)*.
- [Immorlica et al., 2009] Immorlica, N., Li, L. E., Mirrokni, V. S., and Schulz, A. S. (2009). Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410(17) :1589–1598.
- [Irving and Manlove, 1999] Irving, R. W. and Manlove, D. (1999). The b-chromatic number of a graph. *Discrete Applied Mathematics*, 91(1–3) :127–141.
- [Jain et al., 1994] Jain, R. K., Chiu, d. M. W., and Hawe, W. R. (1994). A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC-TR-301, DEC Research Report.
- [Javadi and Omoomi, 2009] Javadi, R. and Omoomi, B. (2009). On b -coloring of the kneser graphs. *Discrete Mathematics*, 209 :4399–4408.
- [Johnson et al., 1988] Johnson, D., Papadimitriou, C., and Yannakakis, M. (1988). How easy is local search? *Journal of Computer System Science*, 37(1) :79–100.
- [Joulie, 2007] Joulie, A. (2007). *Séparation de graphes et d'hypergraphes pour l'étude des réseaux métaboliques : application à Lactococcus Lactis*. PhD thesis, Université de Versailles-Saint Quentin.
- [Kleitman and Vohra, 1990] Kleitman, D. J. and Vohra, R. V. (1990). Computing the bandwidth of interval graphs. *SIAM J. Discrete Math*, 3(3) :373–375.
- [Kortsarz and Krauthgamer, 2001] Kortsarz, G. and Krauthgamer, R. (2001). On approximating the achromatic number. In *SODA '01 : Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 309–318, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- [Kouider and Zaker, 2006] Kouider, M. and Zaker, M. (2006). Bounds for the b-chromatic number of some families of graphs. *Discrete Mathematics*, 306(7) :617–623.
- [Koutsoupias and Papadimitriou, 1999] Koutsoupias, E. and Papadimitriou, C. (1999). Worst-case equilibria. In *Symposium on Theoretical Computer Science (STACS'99)*, pages 404–413, Trier, Germany.
- [Kratochvíl et al., 2002] Kratochvíl, J., Tuza, Z., and Voigt, M. (2002). On the b-chromatic number of graphs. In Kucera, L., editor, *Graph-Theoretic Concepts in Computer Science, 28th International Workshop (WG)*, volume 2573 of *Lecture Notes in Computer Science*, pages 310–320. Springer.
- [Kratsch and Stewart, 2002] Kratsch, D. and Stewart, L. (2002). Approximating bandwidth by mixing layouts of interval graphs. *SIAM J. Discrete Math.*, 15(4) :435–449.
- [Krysta and Loryś, 2006] Krysta, P. and Loryś, K. (2006). Efficient approximation algorithms for the achromatic number. *Theor. Comput. Sci.*, 361(2) :150–171.
- [Laskar and Lyle, 2009] Laskar, R. and Lyle, J. (2009). Fall colouring of bipartite graphs and cartesian products of graphs. *Discrete Appl. Math.*, 157(2) :330–338.
- [Legrand and Touati, 2007a] Legrand, A. and Touati, C. (2007a). How to measure efficiency? In *Proceedings of the 1st International Workshop on Game theory for Communication networks (Game-Comm'07)*, ACM International Conference Proceeding Series, page 8. ACM.

- [Legrand and Touati, 2007b] Legrand, A. and Touati, C. (2007b). Non-cooperative scheduling of multiple bag-of-task applications. In *International Conference on Computer Communications (INFOCOM)*, pages 427–435. IEEE.
- [Leighton and Rao, 1988] Leighton, F. T. and Rao, S. (1988). An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *29th Annual Symposium on Foundations of Computer Science FOCS*, pages 422–431.
- [Lemke and Howson, 1964] Lemke, C. E. and Howson, J. J. T. (1964). Equilibrium points of bimatrix games. *SIAM Journal on Applied Mathematics*.
- [Miyasawa, 1961] Miyasawa, K. (1961). *On the convergence of the learning process in a 2x2*. Econometric Research Program, Princeton University.
- [Monderer and S., 1996] Monderer, D. and S., S. L. (1996). Fictitious play property for games with identical interests. *Journal of Economic Theory*.
- [Monderer and Sela, 1996] Monderer, D. and Sela, A. (1996). A 2×2 game without the fictitious play property. *Games and Economic Behavior*.
- [Monderer and Shapley, 1996] Monderer, D. and Shapley, L. (1996). Potential games. *Games and economic behavior*, 14(1) :124–143.
- [Monien, 1986] Monien, B. (1986). The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. *SIAM J. Algebraic Discrete Methods*.
- [Monien and Sudborough, 1986] Monien, B. and Sudborough, I. H. (1986). Min cut is np-complete for edge weighted trees. In *International Colloquium on Automata, Languages and Programming on Automata, languages and programming*, pages 265–274, New York, NY, USA. Springer-Verlag New York, Inc.
- [Nash, 1950] Nash, J. F. (1950). Equilibrium points in n -person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36 :48–49.
- [Nisan et al., 2007] Nisan, N., Roughgarden, T., Tardos, É., and Vazirani, V. V. (2007). *Algorithmic Game Theory*. Cambridge University Press.
- [Osbourne and Rubinstein, 1994] Osbourne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.
- [Papadimitriou, 1994] Papadimitriou, C. H. (1994). On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*.
- [Paschos, 2003] Paschos, V. T. (2003). Polynomial approximation and graph-coloring. *Computing*, 70(1) :41–86.
- [Pieris and G.H., 1994] Pieris, G. R. and G.H., S. (1994). Scheduling Transmission in WDM Broadcast-and-Select Networks. *IEEE/ACM Transaction on Networking*, 2(2).
- [project of the 5th PCRD, 2005] project of the 5th PCRD, E. (2005). Data and voice integration over dwdm (david).
- [Rabie, 2009] Rabie, M. (2009). Protocoles de population et théorie des jeux. Master’s thesis, Ecole Normale Supérieure de Lyon.
- [Robinson, 1951] Robinson, J. (1951). An iterative method of solving a game. *Annals of Mathematics*.

- [Rosenthal, 1973] Rosenthal, R. (1973). A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1) :65–67.
- [Roughgarden, 2007] Roughgarden, T. (2007). *Algorithmic Game Theory*, chapter Routing Games. Cambridge University Press.
- [Rouskas and Sivaraman, 1996] Rouskas, N. and Sivaraman, V. (1996). On the Design of Optimal TDM Schedules for Broadcast WDM Networks with Arbitrary Transceiver Tuning Latencies. In *IEEE INFOCOM'96*, pages 1217–1224.
- [Rousseau, 2006] Rousseau, S. (2006). *Algorithmique du routage dans les réseaux optiques*. PhD thesis, Université de Versailles-Saint Quentin.
- [Sandholm, 2001] Sandholm, W. (2001). Potential games with continuous player sets. *Journal of Economic Theory*, 97(1) :81–108.
- [Savani and von Stengel, 2004] Savani, R. and von Stengel, B. (2004). Exponentially many steps for finding a nash equilibrium in a bimatrix game. In *Symposium on Foundations of Computer Science (FOCS)*, pages 258–267. IEEE Computer Society.
- [Schulz and Moses, 2003] Schulz, A. S. and Moses, N. S. (2003). On the performance of user equilibria in traffic networks. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 86–87, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- [Schuurman and Vredeveld, 2007] Schuurman, P. and Vredeveld, T. (2007). Performance guarantees of local search for multiprocessor scheduling. *INFORMS J. on Computing*, 19(1) :52–63.
- [Shapley, 1964] Shapley, L. S. (1964). *Advances in Game Theory*, chapter Some topics in two-person games. Univ. Press, Princeton.
- [Shiloach, 1979] Shiloach, Y. (1979). A minimum linear arrangement algorithm for undirected tree. *SIAM Journal on Computing*, 8(1) :15–32.
- [Sprague, 1994] Sprague, A. P. (1994). An $o(n \log n)$ algorithm for bandwidth of interval graphs. *SIAM J. Discrete Math.*, 7(2) :213–220.
- [Stroock and Varadhan, 1979] Stroock, D. and Varadhan, S. (1979). *Multidimensional Diffusion Processes*. Springer.
- [Tel, 2000] Tel, G. (2000). *Introduction to Distributed Algorithms*. Cambridge university press.
- [Unger, 1998] Unger, W. (1998). The complexity of the approximation of the bandwidth problem. In *Proceedings of the Thirty-ninth Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 82–91.
- [Uriel Feige and Lee, 2007] Uriel Feige, U. and Lee, J. R. (2007). An improved approximation ratio for the minimum linear arrangement problem. *Inf. Process. Lett.*, 101(1) :26–29.
- [Wagner, 2005] Wagner, F. (2005). *Redistribution de données à travers un réseau à haut débit*. PhD thesis, Université Henri Poincaré - Nancy I.
- [Weibull, 1995] Weibull, J. W. (1995). *Evolutionary Game Theory*. The MIT Press.
- [Yannakakis, 1985] Yannakakis, M. (1985). A polynomial algorithm for the min-cut linear arrangement of trees. *Journal of ACM*, 32(4) :950–988.

- [Yannakakis and Gavril, 1980] Yannakakis, M. and Gavril, F. (1980). Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3) :364–372.
- [Zuckerman, 2007] Zuckerman, D. (2007). Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1) :103–128.

Bibliographie Personnelle

Articles publiés dans des revues internationales avec comité de lecture

- [1] Lali Barrière, Johanne Cohen, and Margarita Mitjana. Gossiping in chordal rings under the line model. *Theoretical Computer Science*, 264(1) :53–64, 2001.
- [2] Dominique Barth, Pascal Berthomé, and Johanne Cohen. The eulerian stretch of a digraph and the ending guarantee of a convergence routing. *Journal of Interconnection Networks (JOIN)*, 5(2) :93–109, jun 2004.
- [3] Dominique Barth, Olivier Bournez, Octave Boussaton, and Johanne Cohen. Distributed learning of equilibria in a routing game. In *Parallel Processing Letters (PPL)*, pages 189–204. 2009.
- [4] Dominique Barth, Johanne Cohen, Alain Denise, and Romain Rivière. Shuffling biological sequences with motif constraints. *Journal of Discrete Algorithms*, 2007.
- [5] Dominique Barth, Johanne Cohen, and Corentin Durbach. Multicast tree allocation algorithms for distributed interactive simulation. *International Journal of High Performance Computing and Networking - IJHPCN.*, 4(3/4) :137–151, 2006.
- [6] Dominique Barth, Johanne Cohen, and Touafik Faik. On the b-continuity property of graphs. *Discrete Applied Mathematics*, 155(12) :1761–1768, 2007.
- [7] Dominique Barth, Johanne Cohen, Lynda Gastal, Thierry Mautor, and Sétephane Rousseau. Algorithmic study of complexity of two quality-of-service packet models in an optical slotted ring network. *Journal of Optical Networking*, 5(11) :780–789, 2006.
- [8] Olivier Bournez, Philippe Chassaing, Johanne Cohen, Lucas Gerin, and Xavier Koe-gler. On the convergence of population protocols when population goes to infinity. *Applied Mathematics and Computation*, 2009.
- [9] Thierry Chich, Johanne Cohen, and Pierre Fraignaud. Unslotted deflection routing : a practical and efficient protocol for multi-hop optical networks. *IEEE/ACM Transaction on Networking*, 9(1) :47–58, 2001.
- [10] Johanne Cohen, Anurag Dasgupta, Sukumar Ghosh, and Sébastien Tixeuil. An exercise in selfish stabilization. *ACM Transactions of Adaptive Autonomous Systems (TAAS)*, 3(4), 2008.
- [11] Johanne Cohen, Pierre Fraignaud, and Cyril Gavoille. Recognizing knödel graphs. *Discrete Mathematics*, 250 :41–62, Mar 2002.

- [12] Johanne Cohen, Pierre Fraignaud, Jean-Claude König, and André Raspaud. Optimized broadcasting and multicasting protocols in cut-through routed networks. *IEEE Transactions on Parallel and Distributed Systems*, 9(8) :788–802, 1998.
- [13] Johanne Cohen, Pierre Fraignaud, and Margarita Mitjana. Polynomial time algorithms for minimum-time broadcast in tree. *Theory of Computing Systems*, 35(6) :641–665, 2002.
- [14] Johanne Cohen, Emmanuel Jeannot, Nicolas Padoy, and Frédéric Wagner. Message scheduling for parallel data redistribution between clusters. *IEEE Transactions on Parallel and Distributed Systems.*, 17(10) :1136–1150, 2006.

Articles publiés dans des conférences internationales avec comité de lecture

- [15] Lali Barrière, Johanne Cohen, and Margarita Mitjana. Gossiping in chordal rings under the line model. In *23rd Symposium on Mathematical Foundations of Computer Science*, pages 37–47, 1998.
- [16] Dominique Barth, Olivier Bournez, Octave Bousaton, and Johanne Cohen. Distributed learning of wardrop equilibria. In Springer-Verlag, editor, *7th International Conference on Unconventional Computation (UC 2008)*, volume 5204 of *Lecture Notes in Computer Science*, pages 19–32, 2008.
- [17] Dominique Barth, Johanne Cohen, Alain Denise, and Romain Rivière. Shuffling biological sequencess with motifs constraints. In *Algorithms and Computational Methods for Biochemical and Evolutionary Networks (Compbionet 04)*. KCL publications, 2004.
- [18] Dominique Barth, Johanne Cohen, Loubna Echabbi, and Chahinez Hamlaoui. transit price negotiation : a combined game theoretic and distributed algorithmic approach. In *International Conference on Network Control and Optimization (EuroFGI NET-COOP'2007)*, 2007.
- [19] Dominique Barth, Johanne Cohen, Paraskevi Fragopoulou, and Gérard Hébuterne. Wavelengths assignment on a ring all-optical metropolitan area network. In *3rd Workshop on Approximation and Randomization Algorithms in Communication Networks - ARACNE'2002, Rome, Italy*, September 2002.
- [20] Dominique Barth, Johanne Cohen, Lynda Gastal, Thierry Mautor, and Stephane Rousseau. Comparison of fixed size and variable size packet models in an optical ring network : Algorithms and performances. In *Photonics in Switching (PS'2003)*, pages 89–91, 2003.
- [21] Dominique Barth, Johanne Cohen, Lynda Gastal, Thierry Mautor, and Stephane Rousseau. Fixed size and variable size packet models in an optical ring network : Complexity and simulations. In Cevdet Aykanat, Tugrul Dayar, and Ibrahim Korpeoglu, editors, *19th International Symposium on Computer and Information Sciences (ISCIS)*, volume 3280 of *Lecture Notes in Computer Science*, pages 238–246. Springer, Oct 2004.
- [22] Pascal Berthomé, Johanne Cohen, and Afonso Ferreira. Embedding tori in partitioned optical passive star networks. In *4th International Colloquium on Structural*

- Information and Communication Complexity (SIRROCO)*, pages 40–52. Carleton Scientific press, 1997.
- [23] Vincent Bouchitté, Johanne Cohen, and Eric Fleury. Optimal deadlock-free path-based multicast algorithms in meshes. In *5th International Colloquium on Structural Information and Communication Complexity (SIRROCO)*, Amalfi, Italy, June 1998. Carleton Scientific press.
- [24] Olivier Bournez, Jérémie Chalopin, and Johanne Cohen. Playing with population protocols. In *International Workshop on The Complexity of Simple Programs*, 2008.
- [25] Olivier Bournez, Philippe Chassaing, Johanne Cohen, Lucas Gerin, and Xavier Koe-gler. On the convergence of a population protocol when population goes to infinity. In *Physics and Computations, Workshop of Unconventional Computation 2008, UC 2008*, Vienna, Austria, August 25-28 2008.
- [26] Johanne Cohen. Broadcasting, multicasting and gossiping in trees under the all-port line model. In *10th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 164–171. ACM press, 1998.
- [27] Johanne Cohen, Fedor Fomin, Pinar Heggernes, Dieter Kratsch, and Gregory Kucherov. Optimal linear arrangement of interval graphs. In Springer-Verlag, editor, *31st International Symposium on Mathematical Foundations of Computer Science - MFCS 2006*, Lecture Notes in Computer Science, pages 267–279, 2006.
- [28] Johanne Cohen, Pierre Fraignaud, and Cyril Gavoille. Recognizing bipartite incident-graphs of circulant digraphs. In *25-th Workshop on Graph-Theoretic Concepts in Computer Science (WG '99)*, volume 1665 of *LNCS*, pages 215–227, 1999.
- [29] Johanne Cohen, Pierre Fraignaud, Jean-Claude König, and André Raspaud. Optimized broadcasting and multicasting in cut-through routed networks. In *7th International Parallel Processing Symposium (IPPS'97)*, pages 734–738, 1997.
- [30] Johanne Cohen, Pierre Fraignaud, and Margarita Mitjana. Minimal contention-free matrices with application to multicasting. In *DIMACS Workshop on Robust Communication Networks*, 1999.
- [31] Johanne Cohen, Pierre Fraignaud, and Margarita Mitjana. Scheduling calls for multicasting in tree-networks. In *10th ACM-SIAM Symp. on Discrete Algorithms (SODA '99)*, pages 881–882, 1999.
- [32] Johanne Cohen, Emmanuel Jeannot, and Nicolas Padoy. Messages scheduling for data redistribution between clusters. In *Algorithms, models and tools for parallel computing on heterogeneous networks (HeteroPar'03) workshop of SIAM PPAM 2003*, volume 3019 of *LNCS*, pages 896–906, Czestochowa, Poland, sep 2003.

Articles publiés dans des conférences françaises avec comité de lecture

- [33] Henry Amet, Johanne Cohen, Freddy Deppner, Marie-Claude Portmann, and Stéphane Rousseau. Un problème d'ordonnancement de messages : Partie 1 modélisations. In *congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'05)*, pages 54–55, 2005.

Articles cités soumis à des revues internationales avec comité de lecture

- [34] Dominique Barth, Johanne Cohen, and Taoufik Faik. Non approximability and non-continuity of the fall coloring graph problem. Soumis à Discrete Applied Mathematics, 2009.

Articles cités soumis à des conférences internationales avec comité de lecture

- [35] Johanne Cohen, Anurag Dasgupta, Sukumar Ghosh, and Sébastien Tixeuil. Selfish stabilization in maximum flow tree. Technical Report, 2009.

Rapports de recherche cités

- [36] Dominique Barth, Johanne Cohen, and Touafik Faik. Complexity of determining the b-continuity property of graphs. Technical Report RR-2003/37, PRiM, Université Versailles, 2003.
- [37] Dominique Barth, Johanne Cohen, and Touafik Faik. Non approximability and non-continuity of the fall coloring problem. Technical Report 1402, LRI, 2005.
- [38] Dominique Barth, Johanne Cohen, and Antoine Joulie. About complexity and tractability of the minimum α -vertex graph separator problem. Technical Report 3, PRiM, Université Versailles, 2009.
- [39] Olivier Bournez and Johanne Cohen. Learning equilibria in games by stochastic distributed algorithms. Technical Report abs/0907.1916, CoRR, 2009.
- [40] Olivier Bournez and Johanne Cohen. Stochastic learning of equilibria in games : The ordinary differential equation method. Technical report, PRiM, Université Versailles, 2009.
- [41] Johanne Cohen. On the fall coloring for graph having a degree less than 3. Technical Report, 2009.
- [42] Johanne Cohen and Sylvie Delaët. Stability of routes in interdomain routing. Submitted, July 2009.
- [43] Johanne Cohen and Fanny Pascual. Nash equilibria in allocation task game. Technical Report, 2009.