



HAL
open science

Problèmes d'optimisation dans les graphes paramétrés

Bastien Sérée

► **To cite this version:**

Bastien Sérée. Problèmes d'optimisation dans les graphes paramétrés. Mathématique discrète [cs.DM].
École centrale de Nantes, 2022. Français. NNT : 2022ECDN0066 . tel-04056794

HAL Id: tel-04056794

<https://theses.hal.science/tel-04056794>

Submitted on 3 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Bastien SÉRÉE

Problèmes d'optimisation dans les graphes paramétrés

Thèse présentée et soutenue à l'École Centrale de Nantes, le 15 décembre 2022
Unité de recherche : UMR6004, Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Étienne ANDRÉ Professeur des universités, Université Sorbonne Paris Nord
Pierre-Alain REYNIER Professeur des universités, Aix-Marseille Université

Composition du Jury :

Président :	Nicolas MARKEY	Directeur de recherche CNRS, CNRS Rennes
Examineurs :	Nathalie SZNAJDER	Maître de conférences, Sorbonne Université
	Stéphane LE ROUX	Maître de conférences, Ecole normale supérieure Paris Saclay
Dir. de thèse :	Didier LIME	Professeur des universités, École Centrale de Nantes
Co-encadrant :	Loïg JEZEQUEL	Maître de conférences, Nantes Université

REMERCIEMENTS

Je tiens tout d'abord à remercier Pierre-Alain Reynier et Étienne André pour avoir accepté de relire ma thèse et d'en être les rapporteurs, pour tous leurs retours détaillés qui m'ont permis d'améliorer ce manuscrit. Je tiens à remercier Nicolas Markey d'avoir accepté d'être président du jury et aussi Stéphane Le Roux et Nathalie Sznajder pour avoir accepté de faire partie de mon jury. Je remercie également tous les membres du jury d'avoir accepté d'assister à la présentation de mon travail, pour leurs questions et leurs conseils.

Je voudrais remercier tout particulièrement Didier Lime pour avoir été mon directeur de thèse et Loïg Jezequel qui a co-encadré ce travail. Ils ont toujours été disponible et à l'écoute et leur nombreux conseils m'ont beaucoup apporté. Ils m'ont permis de gagner en qualité et en efficacité dans ma manière de faire de la recherche et ont su être à l'écoute lorsque j'en avais besoin. Leurs nombreuses relectures et leurs retours précis et détaillés ont été très appréciables et cette thèse leur doit beaucoup.

Je remercie toute l'équipe STR du LS2N pour la bonne ambiance de travail, les petits conseils pratiques, la bonne humeur générale mais surtout pour le sentiment d'avoir réellement fait partie de cette équipe tout au long de mon doctorat.

Un grand merci à toutes les personnes avec qui j'ai extrêmement apprécié discuter et enseigner. Tout cela m'a permis de prendre confiance en moi, d'améliorer ma vision de la pédagogie et de me confirmer que je veux continuer à enseigner.

Je remercie également toute la team des doctorants du LS2N pour le soutien moral, l'ambiance sérieusement détendue et tous les petits moments qui permettent d'oublier le travail. Tout particulièrement un grand merci à Rémi et Samuel pour les pauses qui permettent de trouver des solutions et celles permettant réellement de faire une pause, sans oublier ces longs moments d'incompréhension devant des problèmes bien trop éloignés de nos connaissances que l'on s'infligeait les uns aux autres.

Je tiens aussi à remercier tous les gens de ma famille qui m'ont soutenu. Merci à Francine pour sa relecture attentive malgré son affection toute relative pour les mathématiques.

Je remercie mes chats pour leurs câlins et leur adorable bêtise, je ne remercie pas

Nuage pour son obsession à manger mes feuilles de brouillon.

Enfin, je remercie Mel, qui a su me soutenir, me supporter, m'encourager pendant toute cette thèse et surtout lors des derniers mois qui n'ont pas toujours été faciles. Je n'aurais pas pu finir mon doctorat sans tout ça. Merci d'être toi...

SOMMAIRE

1	Introduction	7
	Introduction	7
1.1	Motivations générales	7
1.1.1	Vérification formelle	8
1.1.2	Graphes	9
1.1.3	Énergies	10
1.1.4	Paramètres	12
1.2	Définitions	14
1.2.1	Graphes orientés à énergie paramétrée	14
1.2.2	Machine à compteurs	20
1.2.3	Problèmes	21
1.3	État de l'art	23
1.3.1	Plus courts chemins	23
1.3.2	Problèmes d'accessibilité paramétrée en dimension 1	23
1.3.3	Accessibilité avec contraintes sur l'énergie dans les graphes non paramétrés	26
1.3.4	Problèmes multi-dimensionnels	28
1.4	Contributions et plan	32
2	Arbres des plus courts chemins	35
2.1	Introduction	35
2.2	Définitions	36
2.2.1	Graphes paramétrés	36
2.2.2	Arbres sur des graphes paramétrés	38
2.2.3	Contraintes et zones associées aux arbres	43
2.3	Présentation de notre algorithme de calcul des énergies minimales	43
2.4	Présentation formelle de notre algorithme	47
2.4.1	L'algorithme	48

2.4.2	Preuve de l'algorithme	49
2.4.3	Complexité	53
2.5	Cas avec des paramètres à valeurs entières	59
2.6	Conclusion	60
3	Accessibilité multi-dimensionnelle à énergie paramétrée bornée	63
3.1	Introduction	63
3.2	Problème LU total avec paramètres à valeurs entières	64
3.3	Problème LU total avec paramètres à valeurs rationnelles	66
3.4	Conclusion	75
4	Jeux de parité à énergie paramétrée	77
4.1	Introduction	77
4.2	Définitions	78
4.3	Cas à un joueur	79
4.3.1	Définitions	79
4.3.2	Transformation d'un chemin en schéma d'itération	80
4.3.3	Transformation d'un chemin en schéma multiple d'itération	89
4.3.4	Chemins gagnants	97
4.3.5	Complexité	102
4.4	Cas à deux joueurs	104
4.4.1	Définitions	104
4.4.2	Stratégies particulières	107
4.4.3	Complexité	111
4.5	Conclusion	116
5	Conclusion	119
Conclusion		119
5.1	Principaux résultats	119
5.1.1	Perspectives et futurs travaux	120
Bibliographie		123

INTRODUCTION

1.1 Motivations générales

L'utilisation par les humains de moyens pour estimer des quantités remonte au début de leur Histoire. En effet ceux-ci ont éprouvé très tôt le besoin de calculer. Ainsi nous en retrouvons des traces précoces. Par exemple, une des plus anciennes sources écrites découverte à ce jour montre la présence de calculs à la fin du IV^e millénaire avant J.-C, en Mésopotamie. Ces données archéologiques se présentent sous la forme d'une tablette de compatibilité gravée dans l'argile¹. On observe une grande diversité des usages et des calculs deviennent nécessaires, des méthodes se développent dans de nombreux champs pour organiser les activités quotidiennes comme dans l'agriculture, les activités marchandes, les déplacements ou encore les activités militaires mais également dans le développement des sciences et des connaissances de façon générale telles que la géométrie, l'architecture, l'astronomie, etc. Au fil du temps, les besoins d'outils et de méthodes n'ont fait que croître pour répondre à des problèmes de plus en plus complexes. Les progrès technologiques ont notamment apporté des questionnements supplémentaires et une nécessité de pousser le développement de ces méthodes.

La complexification des problèmes à résoudre implique une charge supplémentaire pour l'être humain qui atteint certaines limites : erreurs de calcul, coût en temps très important, etc. Ainsi des méthodes ont vu le jour afin de rendre les calculs plus efficaces et plus fiables. Les algorithmes sont des méthodes de calcul qui ont été particulièrement développées et sophistiquées pour permettre une assistance à la résolution de ces problèmes complexes. Ce sont des suites d'opérations à suivre pas à pas, suites d'opérations qui peuvent ensuite être automatisées notamment avec l'aide d'outils informatiques, remplaçant ainsi le travail fastidieux et moins fiable des humains.

1. Musée du Louvre : <https://collections.louvre.fr/ark:/53355/c1010120676>

1.1.1 Vérification formelle

Une des techniques utilisées est celle des méthodes dites formelles. Les méthodes formelles sont des techniques permettant une rigueur mathématique et l'utilisation de la logique afin de s'assurer de la fiabilité du résultat. Il s'agit dans un premier temps d'abstraire le problème avec un modèle mathématique ce qui permet de l'isoler et de le considérer en lui-même, indépendamment d'éléments contextuels liés aux cas particuliers. Dans un second temps le problème ainsi abstrait est résolu. Il est ensuite possible, dans un dernier temps, d'appliquer la solution ainsi trouvée au problème concret.

Nous pouvons rapidement comprendre les avantages à utiliser un modèle. Cela permet de manipuler un objet purement mathématique. Il est ainsi possible de s'affranchir de la réalité qui implique de nombreuses difficultés dans le déroulement de résolutions de problèmes. En effet, le modèle permet de ne pas prendre en compte certaines spécificités relatives aux cas particuliers afin de généraliser le problème. Les algorithmes interviennent dans l'identification ou la vérification de propriétés mathématiques sur le modèle. Il est ensuite possible d'apporter la preuve mathématique que les solutions ainsi déterminées sont correctes (sur le modèle).

Cependant des méthodes mathématiques entraînent un surcoût lors des phases de développement qui sont alors plus longues. Ce défaut est néanmoins compensé par le niveau de fiabilité supérieur apporté par de telles méthodes qui peuvent rendre plus rapides les phases de validation mais qui sont surtout nécessaires dans de nombreux domaines qui ne peuvent pas se permettre une fiabilité moindre. En effet si lors d'une réunion par visioconférence la perte d'une image ne sera pas dommageable pour la qualité, la fluidité ou la compréhension de la conversation, dans des domaines tels que la sûreté nucléaire ou les systèmes embarqués il est indispensable d'avoir des algorithmes beaucoup plus fiables.

Nous pouvons donc dégager deux grands axes à cette méthode. D'une part les informations importantes et nécessaires à la résolution du problème sont identifiées puis traduites en un modèle adapté. D'autre part, le problème est résolu mathématiquement dans ce modèle. Tous ces éléments appartiennent à des domaines différents des mathématiques. Le travail de cette thèse s'inscrit dans ce dernier axe, celui des méthodes mathématiques utilisées pour résoudre des problèmes sur les modèles abstraits.

Ces méthodes, dites formelles [Woo+09], permettent d'avoir des outils mathématiques à disposition, applicables à tout problème une fois abstrait, qu'il soit initialement connu ou non. Une fois une situation modélisée, la méthode est applicable indépendamment de la nature du problème et fournit une solution clés en main sur le modèle et donc

des pistes privilégiées de solutions pour le problème réel, même s'il n'a pas encore été étudié spécifiquement. Nous avons choisi ici de nous intéresser à la famille d'abstractions mathématiques que sont les graphes.

1.1.2 Graphes

Prenons un exemple. Imaginons que nous souhaitions abstraire un réseau routier. Il est alors possible de symboliser les villes par des points et les routes par des traits. Deux points seront donc reliés par un trait si une route existe dans la réalité entre ces deux villes. Nous pouvons nous poser une question, un problème, plutôt simple pour commencer : puis-je aller de la ville A à la ville B par la route ? Ce qui revient à se demander s'il existe un réseau de routes suffisant pour relier A et B sans discontinuité. En effet, nous pourrions aisément imaginer des cas où deux villes A et B ne seraient pas reliées par des routes, si une des deux villes se trouve sur une île ou un autre continent par exemple. Ainsi sur notre modèle la question revient à savoir si en partant du point correspondant à la ville A, il est possible, en passant de trait en trait, d'aller jusqu'au point correspondant à la ville B. Ce type de modèle s'appelle un graphe. On notera qu'au sein des graphes ce que nous avons appelé des points et des traits se nomment respectivement des sommets et des arcs non orientés. La question posée revient quant à elle à se demander si A et B appartiennent à la même composante connexe de ce graphe. Ici le modèle permet de traiter le problème en excluant les données non nécessaires telles que la largeur des routes ou encore le nombre de voies.

Un des avantages majeurs de ces modélisations est, comme évoqué plus tôt, leur généralisation à des problèmes divers. Ainsi les outils qui permettent de répondre à ce type de problème permettent également de répondre à des problèmes appartenant à d'autres domaines. Prenons l'exemple du domaine des réseaux sociaux. Il est possible, à l'instar de notre exemple précédent, de modéliser les composantes d'un réseau social par des sommets et des arcs non orientés. Les sommets représenteront les personnes au lieu des villes et les arcs non orientés représenteront les relations entre des personnes au lieu des routes. Par exemple, un arc non orienté peut être considéré comme une relation « d'ami » sur Facebook, Snapchat ou encore Whatsapp, la particularité de ces réseaux sociaux étant le caractère symétrique des relations. Sur ces réseaux sociaux deux personnes doivent être « amies » pour pouvoir échanger des contenus dans les deux sens et étudier la structure des graphes peut aider à comprendre certains processus sociaux à l'œuvre comme par exemple l'étude des liens sociaux des organisations terroristes [CNM06]. Dans le cas des

réseaux sociaux la question que nous nous posons pourrait être de savoir quel utilisateur du réseau social pourra potentiellement voir le contenu partagé par un autre utilisateur. Par exemple si un utilisateur A partage une photo, est-ce que l'utilisateur B pourra la voir ? Au même titre que dans l'exemple du réseau routier cette question revient à se demander si A et B appartiennent à la même composante connexe de ce graphe. Ainsi le même algorithme appliqué à cette même modélisation permettra de répondre aussi bien aux deux questions : Peut-on se rendre de la ville A à la ville B par la route ? L'utilisateur B du réseau social pourra-t-il voir le contenu partagé par l'utilisateur A ? La même technique de résolution peut ainsi être appliquée à des problèmes différents.

Il est possible de complexifier ce modèle de graphe pour prendre en compte des subtilités supplémentaires en faisant non pas des arcs non orientés mais des arcs qui bénéficieront d'une orientation, ce qui revient à symboliser des relations possibles dans un seul sens plutôt que dans les deux. Ainsi dans notre exemple de réseau routier un arc allant d'un point A à un point B signifierait que la route est en sens unique, les usagers de la route pouvant se rendre de la ville A à la ville B par ce biais mais pas de la ville B à la ville A. Sur notre exemple des réseaux sociaux un arc allant d'un point A à un point B signifierait une relation asymétrique entre la personne A et la personne B. C'est le cas par exemple dans des réseaux sociaux tels que Twitter, Instagram ou TikTok où la personne A suit la personne B. La personne A reçoit alors du contenu de la part de la personne B sans que l'inverse ne soit vrai, B pouvant ne pas suivre A. On retrouve aussi ces modèles dans le fonctionnement des abonnements des plateformes telles que Twitch ou Youtube.

Ce modèle est un graphe orienté et les méthodes qui seront utilisées sur ce graphe seront les mêmes, indépendamment de la nature du problème réel modélisé, que ce soit une carte routière, un réseau social [NMD18] ou tout autre problème qui puisse être abstrait dans ce modèle.

1.1.3 Énergies

À l'instar de notre exemple de réseau routier nous pouvons aisément imaginer des problèmes plus complexes, impliquant davantage de variables. Prenons pour exemple le cas de la consommation énergétique d'un robot. Les différentes positions du robot, modélisées à nouveau par un graphe, sont notées A, B, C, etc. Dans ce contexte deux positions différentes du robot sont reliées par des actions et différentes actions ou séries d'actions peuvent permettre de passer de la position A à la position B (par des successions d'arcs différents). Une action pourrait être un déplacement par exemple. Il est également possible

de considérer que le robot effectue une action sans changer de position. Cette action pourrait être de se recharger par exemple. L'arc reliera alors le point A à ce même point A. La question serait alors de savoir quelles actions seraient les plus à même de réaliser une tâche donnée. Il est possible de complexifier de nouveau le modèle pour répondre à cette problématique. Cela serait représenté sur le graphe par des nombres que nous associerions aux arcs et qui correspondent dans notre exemple à l'énergie de l'action symbolisée par cet arc. Bien que ces nombres soient parfois appelés « coût » ou « poids » des arcs, nous avons choisi de les appeler « énergie » des arcs pour la suite. En effet le terme d'énergie correspond mieux aux problèmes que nous étudierons dans cette thèse. Dans notre exemple de robot ces nombres représentent l'énergie utilisée par le robot pour effectuer l'action considérée ce qui peut permettre de modéliser l'autonomie de la batterie du robot mais cela pourrait également correspondre à tout autre consommation d'énergie. Ces valeurs peuvent alors être positives, l'action apporte de l'énergie au robot, ou négatives, comme dans des cas où le robot utilise de l'énergie pour effectuer une action ou que l'arc symbolise de la déperdition d'énergie comme sur une batterie inactive. Par exemple, l'arc qui se verrait attribuer l'énergie $-12,7$ entre la position A et la position B indiquerait qu'il existe une action permettant de passer de la position A à la position B et qu'il faut consommer $12,7$ unités d'énergie pour l'effectuer. Une fois cette modélisation réalisée, elle permet de considérer plusieurs questions : quelles sont les actions optimales (qui consomment le moins d'énergie par exemple) pour mettre le robot dans une position donnée ou encore comment effectuer une série d'actions en conservant une énergie positive (ne pas tomber à court de batterie) ? Résoudre ce genre de problèmes devient possible dans ce modèle où nous associons des énergies aux arcs. Le modèle alors utilisé est appelé graphe orienté pondéré puisqu'il inclut les deux possibilités de complexification : l'orientation des relations entre les éléments modélisés et l'ajout d'énergie aux arcs. Le terme « pondéré » est ici employé pour des raisons historiques. En effet, nous avons vu que le terme « poids » est une autre appellation possible pour l'énergie associée à un arc. Ces graphes peuvent servir à répondre à des problèmes tels que ceux posés aux robots comme dans notre exemple mais ne s'y limitent pas. Ils peuvent également s'appliquer à de nombreux problèmes tels que des problèmes d'organisations de réseau de télécommunication ou encore à des problèmes de gestions de stock. L'abstraction mathématique constitue ainsi un moyen d'étudier des problèmes potentiellement inconnus uniquement à la condition qu'ils puissent être modélisés par un objet mathématique connu, ce qui permet de ne pas devoir commencer l'étude d'un tel nouveau problème en créant des méthodes et outils

spécifiques à partir de rien. Lorsque des méthodes pour résoudre ce nouveau problème sur le modèle existent, seule la modélisation du problème réel et l'application de la solution à ce problème sont à faire.

1.1.4 Paramètres

Pour prolonger notre exemple de robot, des séries d'actions peuvent être plus pertinentes que d'autres pour réaliser une tâche donnée en fonction de différentes variables comme certaines conditions environnementales ou des imprécisions dans l'estimation de l'énergie nécessaires pour effectuer une action par exemple. Des conditions environnementales changeantes (conditions climatiques par exemple) pourraient provoquer des changements dans la consommation d'énergie provoquée par une action en l'augmentant ou la diminuant. Les imprécisions pourraient quant à elles se manifester quand l'énergie nécessaire à une action ne peut pas être définie exactement. Par exemple considérons que nous voulons réaliser une tâche et qu'il existe deux séries d'actions permettant cela : effectuer la première série d'actions nécessite la consommation 55ua (unité arbitraire) d'énergie par le robot et effectuer la seconde série d'actions demande une consommation plus élevée de 58ua d'énergie. D'après une approche non paramétrée, la première série d'actions serait alors à privilégier pour réaliser la tâche car elle demande moins d'énergie que la seconde. Cependant nous pouvons imaginer qu'il y ait 5ua d'énergie de marge d'erreur dans l'estimation de la consommation d'énergie pour la première série ce qui implique que l'énergie totale de la série d'actions est comprise entre 50ua et 60ua. Imaginons que la seconde série d'actions ait une consommation d'énergie moins variable dont la marge d'erreur n'est que de 1ua d'énergie, ce qui implique que l'énergie totale nécessaire pour cette série d'actions est comprise entre 57ua et 59ua d'énergie. Il peut alors être plus intéressant de privilégier la seconde série d'actions selon les cas car elle est plus précise dans la certitude des résultats obtenus. En effet s'il est critique pour le robot de ne pas consommer 60ua d'énergie nous souhaiterons privilégier la seconde série d'actions même si nous nous attendons à ce qu'elle demande plus d'énergie que la première. Nous voyons ainsi que nous pouvons envisager des situations où il serait préférable d'effectuer une tâche en utilisant une certaine série d'actions, a priori moins efficace, mais avec peu de marge d'erreur que de l'effectuer avec une série d'actions, a priori plus efficace, mais avec davantage de marge d'erreur. Pour adapter notre modèle à ces considérations, il est ici nécessaire d'ajouter des paramètres aux énergies des arcs afin de modéliser les imprécisions des estimations de la consommation d'énergie. Par exemple un arc avec l'expression « $-12,7 + \lambda_1$ » entre

la position A et la position B indiquera qu'il existe une action reliant la position A à la position B et qu'il faut $12,7 + \lambda_1$ ua d'énergie pour l'effectuer. Dans ce type de situation, le concours de plusieurs variables complexifie la recherche du chemin le plus intéressant entre la position A et la position B. En effet le chemin le plus intéressant ne le sera pas de manière absolue mais dépendra de la valeur du paramètre λ_1 . Ainsi un chemin sera le plus intéressant uniquement pour certaines valeurs de paramètre. Le modèle que nous obtenons avec l'ajout de paramètres est appelé graphe orienté à énergie paramétrée et il permet de considérer plusieurs types de questions dont voici quelques exemples :

- Quelles sont les valeurs des paramètres telles qu'un chemin (série d'actions) du point A au point B soit minimal ? Cela signifie que nous cherchons les valeurs de paramètres qui permettent qu'une série d'actions donnée allant de A à B demande moins d'énergie que les autres séries d'actions envisageables reliant ces deux points ?
- Quels sont les chemins demandant le moins d'énergie pour aller du point A au point B en fonction des valeurs des paramètres ? Ici, contrairement au point précédent, nous cherchons, étant données des valeurs des paramètres, le chemin qui consomme le moins d'énergie pour aller de A à B.
- Est-il possible de trouver des valeurs des paramètres telles qu'un chemin du point A au point B ne dépasse pas une énergie donnée ? C'est-à-dire dans notre exemple, est-il possible de trouver des seuils d'imprécision tels qu'au moins un chemin entre la position A et la position B ne dépasse pas une certaine consommation d'énergie ?

Ainsi comme nous avons pu l'aborder, la modélisation mathématique est une étape dans la résolution de problèmes complexes, les systèmes de résolution ainsi obtenus seront fiables, généralisables et transposables. Cette méthode formelle peut prendre la forme de graphes à différents niveaux de complexité. Les graphes orientés à énergie paramétrée constituent la classe d'objets mathématiques que nous manipulerons dans cette thèse.

Nous nous intéresserons dans le cadre de cette thèse à la recherche de chemins dans les graphes paramétrés. Cela signifie que nous allons nous intéresser non seulement à la recherche de chemins dans des graphes mais également imposer que l'énergie de ces chemins vérifie certaines propriétés comme par exemple être minimale ou encore ne pas sortir d'un intervalle donné.

1.2 Définitions

Dans cette partie nous allons d'abord définir les principaux objets que nous considérerons, puis les problèmes sur lesquels nous nous concentrerons.

Pour tout $(a, b) \in \mathbb{Z}^2$, nous noterons par $\llbracket a, b \rrbracket$ l'ensemble de tous les entiers i tels que $a \leq i \leq b$.

1.2.1 Graphes orientés à énergie paramétrée

Les principaux objets qui seront considérés sont les graphes orientés à énergie paramétrée dont nous allons définir formellement la structure dans cette section ainsi que différentes notions sur les graphes dont nous aurons besoin.

Considérons \mathbb{A} un sous anneau des nombres réels car nous avons besoin d'un ensemble équipé de l'addition, de la multiplication ainsi que d'une relation d'ordre totale. Dans la pratique nous aurons $\mathbb{A} \in \mathbb{R}, \mathbb{Q}, \mathbb{Z}$.

Définition 1.1 (Graphe orienté multi-dimensionnel à énergie paramétrée). *Un Graphe orienté multi-dimensionnel à énergie paramétrée avec d dimensions et n paramètres est un n -uplet $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ où :*

- V est un ensemble fini de sommets,
- $E \subseteq V \times V$ est l'ensemble des arcs,
- d est la dimension du graphe,
- n est le nombre de paramètres du graphe,
- Pour tout $e \in E$, $\mu_e \in \mathbb{N}^d$ est le coefficient de l'arc,
- Pour tout $e \in E$, $\alpha_e : \mathbb{A}^n \rightarrow \mathbb{A}^d$ est une fonction affine à coefficients dans \mathbb{A} , que nous appelons la fonction d'énergie de l'arc,
- Pour tout $v \in V$, $\pi_v \in \mathbb{Z}$ est la priorité du sommet v .

Dans la figure 1.1 nous avons représenté un exemple G_{ex} de graphe orienté multi-dimensionnel à énergie paramétrée. Les sommets sont nommés $u_1 u_2 \dots u_5$ et sont représentés par des disques oranges et les arcs nommés $e_1 e_2 \dots e_8$ par des flèches noires. Les aspects « multi-dimensionnel » et « à énergie paramétrée » du graphe ne sont pas directement représentés sur la figure par souci de lisibilité. En effet les représenter impliquerait de développer les définitions des arcs e_i pour faire apparaître les valeurs des μ_{e_i} ainsi que les expressions mathématiques correspondants aux α_{e_i} .

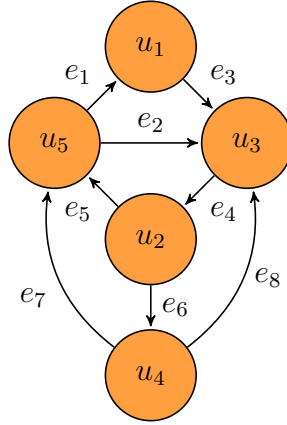


FIGURE 1.1 – Un exemple g_{ex} de graphe orienté multi-dimensionnel à énergie paramétrée.

Par la suite, pour tout a appartenant à S^k , avec $S \in \{\mathbb{N}, \mathbb{R}, \mathbb{A}, \mathbb{Z}, \mathbb{Q}\}$ et $k \in \mathbb{N}$ nous noterons a_i la i -ème coordonnée de a .

Définition 1.2 (Chemins). Soit $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée, soit $\rho = e_1 e_2 \dots$ une suite potentiellement infinie d'arcs avec pour tout $i \geq 1$, $e_i = (v_i, v'_i)$. Nous avons ρ un chemin de G si et seulement si pour tout $i \geq 1$, $v_{i+1} = v'_i$. Le sommet $v_1 \in V$ est le sommet initial de ρ . Un chemin est dit infini si la suite d'arc ρ est infinie et dit fini sinon. Si ρ est un chemin fini, alors l'entier k tel que $\rho = e_1 \dots e_k$ est la longueur du chemin, que nous notons $|\rho|$. Le sommet $v_{k+1} \in V$ est le sommet final de ρ .

Nous disons que l'arc $e = (u, v)$ est dirigé vers v .

Dans la suite, en l'absence de spécification un chemin pourra être fini ou infini.

Soient ρ et ρ' des chemins tels que ρ soit un chemin fini et que le sommet final de ρ soit également le sommet initial de ρ' . Nous noterons $\rho\rho'$ le chemin issu de la concaténation des chemins ρ et ρ' .

Nous avons représenté sur la figure 1.2 le chemin $\rho_1 = e_1 e_3 e_4 e_6$ et sur la figure 1.3 le chemin $\rho_2 = e_1 e_3 e_4 e_6 e_8$, des chemins du graphe G_{ex} de la figure 1.1.

Définition 1.3 (Cycle). Un chemin fini est un cycle si et seulement si son sommet initial est égal à son sommet final.

Nous avons représenté sur la figure 1.4 le chemin $e_2 e_4 e_6 e_7$ de G_{ex} . Ce chemin est un cycle car son sommet initial et son sommet final sont u_5 .

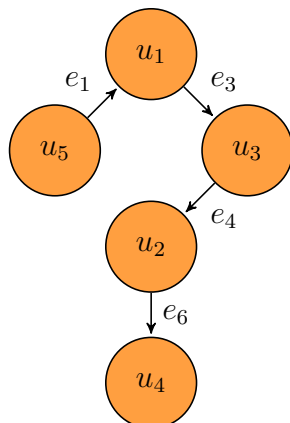


FIGURE 1.2 – Le chemin ρ_1 de G_{ex} .

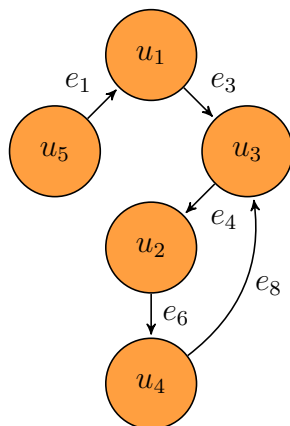


FIGURE 1.3 – Le chemin ρ_2 de G_{ex} .

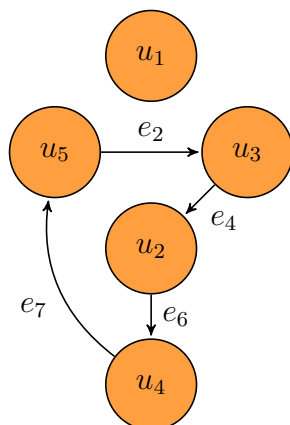


FIGURE 1.4 – Un cycle de G_{ex} .

Définition 1.4 (Chemin simple). Soit $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée, soit $\rho = e_1 \dots e_k$ un chemin fini tel que pour tout $i \in \llbracket 1, k \rrbracket$, $e_i = (v_i, v_{i+1})$. Le chemin ρ est un chemin simple si et seulement si ρ est le chemin vide ou si pour tout $(j, j') \in \llbracket 1, k+1 \rrbracket^2$ tel que $j \neq j'$, nous avons $v_j \neq v_{j'}$.

Intuitivement un chemin simple est un chemin sans aucun cycle, c'est à dire un chemin où chaque sommet apparaît au plus une seule fois.

Nous pouvons remarquer que le chemin ρ_1 de la figure 1.2 est un chemin simple car aucun sommet n'est visité deux fois tandis que le chemin ρ_2 de la figure 1.3 n'est pas un chemin simple car le sommet u_3 est visité deux fois.

Définition 1.5 (Cycle simple). Soit $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée, soit $\rho = e_1 \dots e_k$ un chemin fini avec pour tout $i \in \llbracket 1, k \rrbracket$, $e_i = (v_i, v_{i+1})$. Le chemin ρ est un cycle simple si et seulement si ρ est vide ou :

- pour tout $(j, j') \in \llbracket 1, k \rrbracket^2$ tels que $j \neq j'$, $v_j \neq v_{j'}$,
- et ρ est un cycle.

Intuitivement un cycle simple est un cycle dans lequel le seul sommet qui apparaît deux fois est le sommet initial (et final).

Nous pouvons remarquer que le cycle représenté sur la figure 1.4 est un cycle simple car le seul sommet visité deux fois est le sommet initial (et final) du cycle.

Définition 1.6 (Préfixe). Soit $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée. Soit $\rho = e_1 e_2 \dots$ un chemin de G , un préfixe de ρ est le chemin vide ou tout chemin $\rho_i = e_1 \dots e_i$ avec $1 \leq i$. Nous définissons également $\text{pref}(\rho)$ comme l'ensemble des préfixes de ρ .

Nous avons représenté sur la figure 1.5 les cinq préfixes du chemin ρ_1 de la figure 1.2. Notons que ρ_1 est un préfixe de lui-même et que le chemin vide est un préfixe de tous les chemins.

Définition 1.7 (Énergie d'un arc). Pour tout $e \in E$ et $x \in \mathbb{A}^n$, soit $\phi_e^x : \mathbb{A}^d \rightarrow \mathbb{A}^d, y \mapsto \mu_e * y + \alpha_e(x)$ l'énergie de e . Avec pour tout $i \in \llbracket 0, d-1 \rrbracket$, $(\mu_e * y)_i = (\mu_e)_i y_i$.

La figure 1.6 consiste en un arc e_1 tel que $\mu_{e_1} = (2, 2)$ et $\alpha_{e_1} : x \mapsto (3 - x_2 + 5x_4, -x_1 + x_2)$. Nous pouvons observer l'aspect multi-dimensionnel du graphe avec la dimension de l'espace d'arrivé de α_{e_1} : le graphe auquel appartient e_1 est de dimension 2 ($d = 2$). De plus

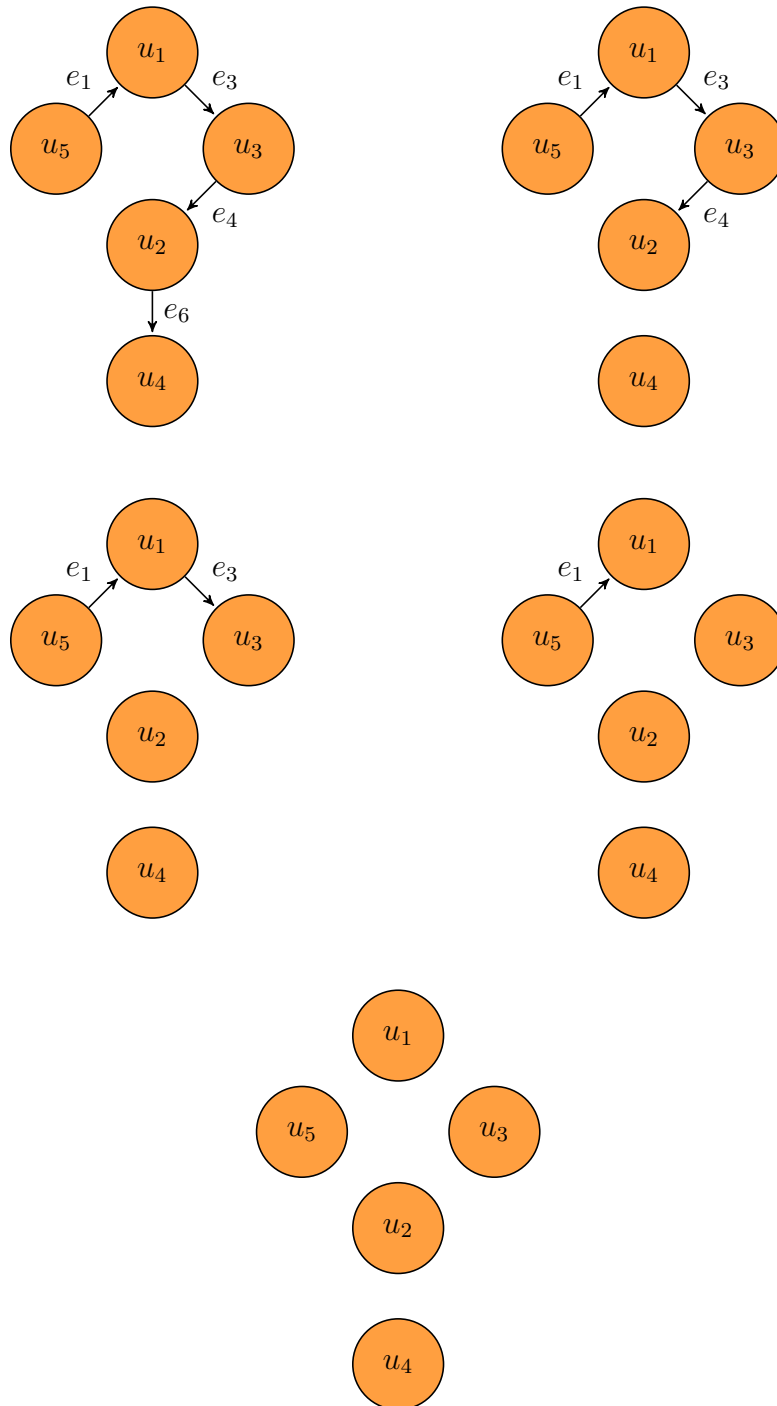
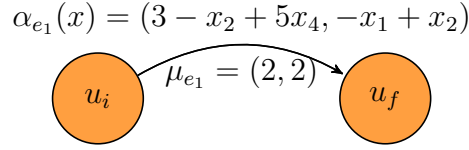


FIGURE 1.5 – Les préfixes du chemin ρ_1 .

FIGURE 1.6 – L'arc e_1 .

comme x_4 apparait dans α_{e_1} nous avons au moins 4 paramètres ($n \geq 4$) dans le graphe auquel appartient e_1 . Enfin, en prenant $y \in \mathbb{A}^2$ la valeur de l'énergie avant d'emprunter e_1 , nous avons $y' = (2y_1 + 3 - x_2 + 5x_4, 2y_2 - x_1 + x_2)$ la valeur de l'énergie en u_f en fonction des valeurs des paramètres.

Définition 1.8 (Énergie d'un chemin). *Pour tout chemin fini $\rho = e_1 \dots e_k$ et pour tout $x \in \mathbb{A}^n$, soit $\phi^x(\rho) := \phi_{e_k}^x \circ \dots \circ \phi_{e_1}^x$ l'énergie du chemin ρ .*

Pour un chemin ρ , une valeur $y \in \mathbb{A}^d$ et des valeurs de paramètres $x \in \mathbb{A}^n$, afin d'alléger les notations nous noterons $\phi^x(\rho)(y) : \phi^x(y, \rho)$

Un arc $e \in E$ est un *arc croissant* si et seulement si $(\mu_e)_i \geq 1$ pour tout i . Un arc e est un *arc constant* si et seulement si $(\mu_e)_i = 0$ pour tout i .

Les arcs sont dits constants car l'énergie d'un chemin sera toujours la même indépendamment de la valeur de l'énergie avant d'emprunter cet arc : les arcs constants modifient l'énergie mais la valeur prise par l'énergie suite à un arc constant donné est toujours la même.

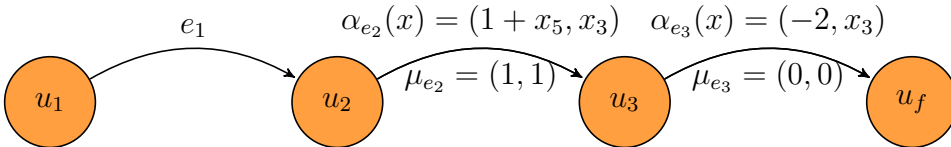


FIGURE 1.7 – Un exemple de chemin.

Nous pouvons observer sur la figure 1.7 un exemple de chemin composé de trois arcs : $e_1 e_2 e_3$ avec e_1 l'arc détaillé sur la figure 1.6. Supposons que la valeur de l'énergie soit de $y \in \mathbb{A}^2$ avant d'emprunter e_1 , nous avons alors une énergie de $(2y_1 + 3 - x_2 + 5x_4, 2y_2 - x_1 + x_2)$ avant d'emprunter e_2 . Puisque $\mu_{e_2} = (1, 1)$ et $\alpha_{e_2} : x \mapsto (1 + x_5, x_3)$, la valeur de l'énergie en u_3 est de $(1(2y_1 + 3 - x_2 + 5x_4) + 1 + x_5, 1(2y_2 - x_1 + x_2) + x_3)$ soit $(2y_1 + 4 - x_2 + 5x_4 + x_5, 2y_2 - x_1 + x_2 + x_3)$. Enfin comme $\mu_{e_3} = (0, 0)$ l'énergie prend la valeur $(-2, x_3)$ après avoir fini de parcourir la totalité du chemin.

Nous pouvons remarquer que la valeur finale de l'énergie ne dépend pas de y et par

conséquent pas de la valeur de l'énergie avant le parcours du chemin, cela est dû au fait que e_3 est un arc constant, la valeur de l'énergie sera donc toujours de $(-2, x_3)$ en u_f c'est-à-dire la valeur $\alpha_{e_3}(x)$.

Offset d'énergie Certaines modélisations des graphes définissent un offset d'énergie E_{off} , ce qui implique que l'énergie initiale n'est pas zéro mais E_{off} . Nous n'utiliserons pas ce concept car il est inclu dans notre modèle, en effet nous pouvons représenter n'importe quel offset sur l'énergie par un arc en début de chemin qui augmente l'énergie de la valeur d'un paramètre, ce paramètre pouvant prendre toutes les valeurs possibles de E_{off} et ainsi modéliser un offset d'énergie.

1.2.2 Machine à compteurs

Dans cette section nous définissons les machines à deux compteurs qui sont utiles aux preuves d'indécidabilité ainsi que différentes notions sur les machines à compteurs dont nous aurons besoin.

Définition 1.9 (Machine à deux compteurs). *Une machine à deux compteurs est un n -uplet $M = (Q, q_0, h, T, \delta, C_1, C_2)$ où :*

- Q est un ensemble fini d'états,
- $q_0 \in Q$ est un état initial,
- $h \in Q$ est un état d'arrêt,
- $T \subseteq Q \times Q$ est l'ensemble des transitions,
- C_1 et C_2 sont des compteurs,
- $\delta : T \rightarrow \{C_1, C_2\} \times \{-, +, =, >\}$ est la fonction qui associe une instruction à chaque transition.

Définition 1.10 (Configuration). *Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs, une configuration (q, x_1, x_2) de M est telle que $q \in Q$ est un état, $x_1 \in \mathbb{N}$ est une valeur du compteur C_1 , et $x_2 \in \mathbb{N}$ est une valeur du compteur C_2 .*

Définition 1.11 (Accessible en une étape). *Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs. Une configuration $c' = (s', x'_1, x'_2)$ est dite accessible en une étape dans M depuis une configuration $c = (s, x_1, x_2)$ si et seulement si il existe une transition $t = (s, s') \in T$ telle que :*

- $(x'_1, x'_2) = (x_1 + 1, x_2)$ (resp. $(x'_1, x'_2) = (x_1, x_2 + 1)$) si $\delta(t) = (C_1, +)$ (resp. $\delta(t) = (C_2, +)$),

- $(x'_1, x'_2) = (x_1 - 1, x_2)$ (resp. $(x'_1, x'_2) = (x_1, x_2 - 1)$) si $\delta(t) = (C_1, -)$ et $x_1 > 0$ (resp. $\delta(t) = (C_2, -)$ et $x_2 > 0$),
- $(x'_1, x'_2) = (x_1, x_2)$ et $x_1 = 0$ (resp. $x_2 = 0$) si $\delta(t) = (C_1, =)$ (resp. $\delta(t) = (C_2, =)$),
- $(x'_1, x'_2) = (x_1, x_2)$ et $x_1 > 0$ (resp. $x_2 > 0$) si $\delta(t) = (C_1, >)$ (resp. $\delta(t) = (C_2, >)$).

Nous le noterons $(s, x_1, x_2) \xrightarrow{t} (s', x'_1, x'_2)$ et nous disons que t est tirable depuis la configuration c .

Définition 1.12 (Exécution et chemin). Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs, une exécution dans M est une séquence $r = \gamma_1 \gamma_2 \dots$ de configurations telles que $\gamma_i \xrightarrow{t_i} \gamma_{i+1}$ avec $t_i \in T$ pour tout $i \in \llbracket 1, k-1 \rrbracket$.

Nous appelons γ_1 la configuration initiale de r .

Le chemin correspondant à r est la séquence de transitions $\rho = t_1 t_2 \dots$.

Si r est finie alors nous avons $r = \gamma_1 \gamma_2 \dots \gamma_k$. Nous appelons k la longueur de r , et γ_k la configuration finale de r .

Définition 1.13 (Accessibilité). Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs, une configuration γ est dite accessible dans M s'il existe une exécution r telle que γ est la configuration finale de r et que la configuration initiale de r soit $(q_0, 0, 0)$.

Nous définissons les préfixes d'un chemin dans une machine à deux compteurs de manière analogue aux graphes.

Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs, soient $r = \gamma_0 \gamma_1 \dots \gamma_k$ une exécution et $\rho = t_1 t_2 \dots t_l$ son chemin associé. Soit $\gamma_k = (s_k, x_{1,k}, x_{2,k})$, nous notons $c_1(\rho) = c_1(r) = x_{1,k}$ (resp. $c_2(\rho) = c_2(r) = x_{2,k}$) la valeur de C_1 (resp. C_2) après le parcours du chemin ρ ou de manière équivalente après l'exécution r .

1.2.3 Problèmes

Dans cette section nous définissons les différents problèmes que nous considérerons. Sauf indication contraire les problèmes sont définis dans le cadre de $G = (V, E, d, n, (\mu_e \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée.

Par souci de lisibilité nous noterons $(L_i, U_i)_1^k$ l'ensemble de couples $\{(L_i, U_i), \text{ pour } i \in \llbracket 1, k \rrbracket\}$.

Définition 1.14 (Chemin borné). Soient ρ un chemin de G et les bornes $(L_i, U_i)_1^d \in (\mathbb{A} \cup \{-\infty, +\infty\})^d$. Nous disons que ρ est borné par les contraintes $(L_i, U_i)_1^d$ pour $\lambda \in \mathbb{A}^n$ si et seulement si $\forall u \in \text{pref}(\rho)$ et $i \in \llbracket 1, d \rrbracket, L_i \leq \phi^\lambda(u)_i \leq U_i$.

Définition 1.15 (Problème d'accessibilité bornée). Soient $(v, w) \in V^2$ deux sommets, soient $(L, U) \in (\mathbb{A}^d)^2$ des contraintes. Nous appellerons problème LU total sur G la question de savoir si $\exists \lambda \in \mathbb{A}^n$ et ρ un chemin de G de v à w tel que ρ est borné par $(L_i, U_i)_1^d$.

Nous notons ce problème $EnTot_{LU}(G, (L_i, U_i)_1^d, \mathbb{A}, v, w)$.

Définition 1.16 (Problème d'accessibilité avec borne inférieure). Soient $(v, w) \in V^2$ deux sommets. Nous appellerons problème L total sur G la question de savoir si $\exists \lambda \in \mathbb{A}^n$ et ρ un chemin de G de v à w tel que ρ est borné par $(0, +\infty)_1^d$.

Nous notons ce problème $EnTot_L(G, \mathbb{A}, v, w)$.

Pour tout $u \in \mathbb{Z}^\omega$, $\text{inf}(u)$ est l'ensemble des entiers qui apparaissent infiniment souvent dans u .

Définition 1.17 (Parité d'un chemin). Soient $\rho = e_1 e_2 \dots$ un chemin et pour tout i , $e_i = (v_i, v_{i+1})$. Soit $\pi(\rho) = \pi_{v_1} \pi_{v_2} \dots \in \mathbb{Z}^\omega$, nous dirons que ρ est pair si et seulement si $\min(\text{inf}(\pi(\rho)))$ est pair et que ρ est impair sinon.

Notons que pour tout chemin infini ρ , $\text{inf}(\pi(\rho))$ est non vide car V est fini, il y a donc au moins un sommet qui apparaît infiniment souvent dans ρ et $\text{inf}(\pi(\rho))$ contient au moins la parité de ce sommet. De plus \min est bien défini car les sommets étant en nombre fini, il ne peut y avoir un nombre infini de parités présentes dans $\text{inf}(\pi(\rho))$. Remarquons également que la parité d'un chemin ne dépend pas de la valeur des paramètres.

Définition 1.18 (Problème à parité avec borne inférieure). Pour tout $\lambda \in \mathbb{A}^n$, nous définissons $W^\lambda \subseteq E^\omega$ tel que $\rho \in W^\lambda$ si et seulement si ρ est pair et pour tout $k \in \mathbb{N}$ nous avons $0 \leq \phi^\lambda(\rho_k, 0)$ avec ρ_k le préfixe de ρ de longueur k . Nous appelons problème à parité avec borne inférieure sur G la question de savoir si $\exists \lambda \in \mathbb{A}^n$ tel que $W^\lambda \neq \emptyset$.

Nous définissons enfin le problème de l'arrêt sur les machines à compteurs dont nous servirons dans certaines preuves.

Définition 1.19 (Problème de l'arrêt). Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs. Nous appelons problème de l'arrêt sur M la question de savoir s'il existe ou non une exécution $r = \gamma_1, \gamma_2 \dots \gamma_k$ telle que $\gamma_1 = (q_0, 0, 0)$ et $\gamma_k = (h, c_1(\rho), c_2(\rho))$ avec ρ le chemin associé à r .

Nous disons que M s'arrête s'il existe une telle exécution.

Le problème de l'arrêt est connu pour être indécidable et Π_1^0 -complet [Min67].

1.3 État de l'art

1.3.1 Plus courts chemins

De nombreuses réponses ont été apportées au problème consistant à calculer le plus court chemin entre deux sommets dans un graphe pondéré non paramétré. Les algorithmes les plus utilisés pour répondre à cette question sont les algorithmes de Dijkstra et de Bellman-Ford ainsi que l'algorithme A^* .

L'algorithme de Dijkstra [Dij59] renvoie le chemin le plus court entre deux sommets dans un graphe orienté dans lequel l'énergie des arcs se limite à des réels positifs. Sa complexité en pire cas est en $O(|E| + |V| \log |V|)$.

L'algorithme de Bellman-Ford [For56] permet de considérer une pondération négative des arcs et sa complexité est en $O(|E||V|)$. Il permet également de détecter l'existence d'un cycle de coût négatif, configuration pour laquelle la notion de plus court chemin n'existera pas.

L'algorithme A^* [HNR68] [HNR72] est un algorithme de recherche de chemin entre deux sommets donnés qui étend l'algorithme de Dijkstra. Il a la particularité, contrairement aux algorithmes précédemment cités, d'utiliser une heuristique. Celle-ci consiste en une méthode d'approximation visant à prioriser l'exploration des solutions potentielles afin de commencer par celles étant les plus prometteuses, l'algorithme de Dijkstra pouvant être vu comme un cas particulier de A^* lorsqu'une certaine heuristique est utilisée.

Il existe également l'algorithme de Floyd-Warshall [Flo62] [War62] qui vise à déterminer des plus courts chemins entre toutes les paires de sommets dans un graphe orienté pondéré avec une complexité en $O(|V|^3)$. À l'instar de l'algorithme de Bellman-Ford on peut considérer des arcs possédant une énergie négative, dans la mesure où, il n'existe pas de cycle à énergie strictement négative.

1.3.2 Problèmes d'accessibilité paramétrée en dimension 1

Quand des systèmes qui ne sont pas encore bien connus sont abordés, la phase amont du processus de conception peut générer des incertitudes. Dans de telles circonstances nous rappelons que ces incertitudes peuvent être modélisées par l'utilisation de paramètres dans les énergies au sein de ces systèmes. Les problèmes qui suscitent l'intérêt sont les problèmes de synthèse des paramètres dans lesquels nous tentons par exemple de trouver les valeurs de paramètres telles que certains chemins s'avèrent optimaux, ou encore qu'un

sommet cible puisse être atteint avec une borne imposée sur l'énergie accumulée.

Automates temporisés paramétrés

De manière surprenante, ces problèmes n'ont pas été beaucoup étudiés pour la configuration dans laquelle les énergies elles-mêmes sont des paramètres. Un Automate Temporisé Paramétré (PTA) [AHV93] permet que les horloges soient testées contre les paramètres ce qui pourrait, dans une certaine mesure, permettre la réalisation de simulations d'énergies paramétrées. En effet les contraintes sur les transitions peuvent être utilisées pour modéliser l'énergie d'un arc. L'accessibilité lorsque l'on peut comparer des horloges aux valeurs des paramètres a été étudiée et différents cas ont été identifiés selon le nombre d'horloges différentes pouvant être testées contre les paramètres. Lorsqu'une seule horloge est paramétrée, quel que soit le nombre de paramètres, le problème est décidable et NEXP-complet [AHV93 ; BO14 ; BO17]. Lorsque deux horloges peuvent être comparées aux paramètres le problème de l'accessibilité a été démontré décidable s'il n'y a qu'un paramètre une complexité EXPSPACE-complète a été prouvée [BO14 ; BO17 ; GH21]. De plus dès lors qu'il y a au moins trois horloges paramétrées l'accessibilité devient indécidable [BO14 ; AHV93]. Récemment, le problème de l'accessibilité dans un temps optimal a également été étudié et montré indécidable pour les automates temporisés lorsque les délais sont paramétrés [And+19a]. De façon similaire, le problème d'accessibilité en énergie bornée a été démontré indécidable pour un formalisme en lien avec les automates temporisés paramétrés appelé réseau de Petri à temps paramétré dans [LRS19]. L'accessibilité à temps borné a également été étudiée pour les réseaux de Pétri à temps paramétré, elle est prouvée indécidable dans [MLA20]. Dans [And+19b], les auteurs étendent le modèle des automates temporisés paramétrés en ajoutant des énergies dans les arcs pour modéliser les problèmes de sécurité. Cependant la topologie du système considéré est toujours celle d'un arbre.

Machines paramétrées à un compteur

De façon très différente, les machines paramétrées à un compteur devraient pouvoir modéliser une énergie paramétrée à une dimension, la valeur du compteur représentant la valeur de l'énergie. Les problèmes de synthèse de paramètre n'ont pas été étudiés sur ce type de modèle mais les problèmes d'existence sont considérés, ce qui pourrait permettre tout de même de modéliser des bornes sur l'énergie. Des questions d'accessibilités sont étudiées dans [Haa+09] et les auteurs démontrent que le problème de l'accessibilité dans

les machines paramétrées à un compteur est décidable lorsque les paramètres ne prennent que des valeurs positives. Des propriétés de logique temporelle ont aussi été abordées sur les machines paramétrées à un compteur. La vérification d'une propriété LTL (Linear Temporal Logic) est décidable avec une complexité coNP -complète [Göl+10] lorsque la propriété est fixée et coNEXP -complète lorsqu'elle est quelconque [Göl+10]. Les problèmes de vérification de propriétés CTL (Computation Tree Logic) [CES86] et de μ -calcul sont quant à eux indécidables avec une complexité Π_1^0 -complète à la fois pour la version du problème où la propriété est fixée et celle où la propriété est quelconque [Göl+10]. Certaines classes de machines paramétrées à un compteur ont également été étudiées de part leurs liens avec les automates temporisés et dans [BO17] nous avons que l'accessibilité dans les machines paramétrées à un compteur est décidable pour deux classes de machines à compteurs. La première classe est composée des machines avec un nombre quelconque de paramètres x_i , telles que les seules modifications de compteurs soient -1 , 0 et $+1$ et que les seules comparaisons possibles soient $\geq x_i$ et ≤ 1 . La seconde classe est celle des machines n'ayant qu'un seul paramètre x mais ayant plus d'opérations permises que pour les machines de la première classe tout en conservant les mêmes comparaisons, les modifications permises étant $-k$, $+k$, $-x$ ou $+x$ avec $k \in \mathbb{N}$.

Les machines paramétrées à un compteur sont très proches du modèle des graphes à énergie paramétrée. Les problèmes que nous considérons sont immédiatement transposables à ces machines, le compteur représentant la valeur de l'énergie. Cependant il réside dans les opérations possibles une différence subtile bien qu'importante entre les graphes et les machines à compteur. En effet il ne semble pas immédiat de modéliser le test à zéro de la valeur de l'énergie dans notre modèle. Ainsi nous ne pouvons donc pas réutiliser facilement les résultats sur les machines à compteur pour les transposer à nos problèmes sur les graphes paramétrés.

Graphes orientés à énergie paramétrée

Dans [And10], l'auteur propose un algorithme qui calcul un chemin d'énergie minimale entre deux sommets grâce à l'algorithme de Floyd-Warshall puis renvoie des inégalités dépendant des énergies des arcs, vues comme des paramètres. Ces inégalités doivent être vérifiées pour que le chemin calculé reste d'énergie minimale.

Dans [KO81] les auteurs considèrent des graphes dans lesquels les énergies sont exprimées comme une fonction d'un seul paramètre. Ils partitionnent les nombres réels de façon finie, telle que pour deux valeurs de paramètre dans une partition donnée, les che-

mins optimaux depuis un certain sommet source vers tous les autres sommets soient les mêmes. Les arbres ainsi constitués par ces chemins sont également explicités dans l'article. De tels graphes paramétrés, avec un unique paramètre, ont été étudiés par la suite dans [YTO91] et dans [Cha+10a] pour leur intérêt dans les applications, notamment aux télécommunications, aux questions de chemins les plus courts.

1.3.3 Accessibilité avec contraintes sur l'énergie dans les graphes non paramétrés

De nombreuses questions d'accessibilité avec contraintes ont été posées dans les graphes non paramétrés, en plus des problèmes d'accessibilité bornée ou d'accessibilité avec borne inférieure que nous avons déjà définis. Des contraintes peuvent être imposées sur l'énergie moyenne des chemins ou encore sur leur variation moyenne d'énergie et nous pouvons définir de nouveaux problèmes comme suit :

Définition 1.20 (Problème d'accessibilité avec contrainte sur l'énergie moyenne). *Soit $G = (V, E, 1, 0, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe non paramétré à une dimension. Soient $(u, v) \in V^2$ deux sommets, et $s \in \mathbb{R}$ un seuil, nous appelons problème d'accessibilité avec contrainte sur l'énergie moyenne sur G la question de savoir s'il existe un chemin ρ de v à w tel que pour tout préfixe non vide $\rho' \in \text{pref}(\rho)$ nous ayons $s \leq \frac{\phi(\rho')}{|\rho'|}$.*

Définition 1.21 (Problème d'accessibilité avec contrainte sur la variation d'énergie moyenne). *Soit $G = (V, E, 1, 0, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe non paramétré à une dimension. Soient $(u, v) \in V^2$ deux sommets, et $s \in \mathbb{R}$ un seuil, nous appelons problème d'accessibilité avec contrainte sur la variation d'énergie moyenne sur G la question de savoir s'il existe un chemin $\rho = e_1 \dots e_k$ de G de v à w tel que pour tout $i \in \llbracket 1, k \rrbracket$ nous ayons $s \leq \frac{1}{i} \sum_{j=1}^i \sum_{l=1}^j \alpha_{e_l}$, ce qui est équivalent à avoir ρ tel que pour tout $i \in \llbracket 1, k \rrbracket$ nous ayons $s \leq \frac{1}{i} \sum_{j=1}^i \phi(e_1 \dots e_j)$.*

Notons que dans les définitions précédentes les α_e sont constants car les graphes considérés sont non paramétrés.

Les résultats proposés par la littérature considèrent généralement des chemins infinis car ceux-ci sont vus comme des cas particuliers lors de l'étude des jeux à énergie sur des graphes [Bou+18; Bou+17; Bou+08], les contraintes sur l'énergie devant toujours être vérifiées pour tous les préfixes du chemin. En effet la recherche d'un chemin infini correspond au cas particulier de la recherche d'une stratégie gagnante d'un jeu dans lequel un

joueur contrôle tous les sommets. Les jeux à énergie sur des graphes consistent à attribuer chaque sommet du graphe à un joueur, cela signifie que c'est le joueur qui contrôle le sommet qui décide quel arc est emprunté lorsque la question se pose. Les problèmes sur les jeux consistent à se demander s'il est possible pour un joueur d'imposer certaines propriétés aux chemins formés lors de l'exécution du jeu. Ces jeux sont très utiles pour modéliser des systèmes devant s'adapter ou résister à des actions extérieures, que ce soit d'agents ou de l'environnement car il suffit alors de voir si le joueur symbolisant le système modélisé peut atteindre son objectif malgré les actions de l'environnement. En effet la recherche de chemins infinis vérifiant certaines propriétés dans un graphe correspond au cas particulier d'un jeu à énergie imposant aux chemins gagnants de vérifier certaines propriétés lorsqu'un joueur ne contrôle aucun sommet. Nous pouvons constater que les chemins finis ne sont que très peu étudiés pour les types de problèmes ne s'intéressant pas à l'accessibilité avec contrainte sur l'énergie moyenne.

Nous pouvons également remarquer que les questions relatives à l'accessibilité bornée et à l'accessibilité avec contrainte sur la variation d'énergie moyenne ont aussi été étudiées en conjonction avec des problèmes de borne. Les différents résultats concernant les graphes mono-dimensionnels sont compilés dans le tableau suivant. Les colonnes indiquent si le problème d'accessibilité avec contrainte sur la variation d'énergie moyenne est considéré et les lignes indiquent le type de bornes imposées aux chemins. Par exemple la deuxième case de la deuxième ligne correspond à la décidabilité de savoir s'il existe un chemin étant solution à la fois du problème d'accessibilité avec borne inférieure et du problème d'accessibilité avec contrainte sur la variation d'énergie moyenne.

	Variation d'énergie moyenne \surd	Variation d'énergie moyenne \times
$EnTot_L$	PSPACE-facile / NP-difficile [Bou+18]	P [Bou+08]
$EnTot_{LU}$	PSPACE-complet [Bou+18]	PSPACE-complet [FJ15]
$EnTot_{LU}^*$?	P [Bou+08]
\emptyset	P [Bou+18]	NLOGSPACE [Gou19]

La dernière case de la dernière ligne du tableau correspond au cas du problème de l'accessibilité puisqu'aucune contrainte n'est imposée à l'énergie du chemin ni à la variation d'énergie moyenne.

L'avant dernière ligne du tableau correspond à $EnTot_{LU}^*$, le problème de l'accessibilité bornée avec borne supérieure faible, la définition du problème est la même que dans le cas de $EnTot_{LU}$ mais le calcul de l'énergie est modifié. En effet, dans les graphes définis

précédemment, pour un chemin $\rho = e_1 \dots e_k$, nous avons $\phi(\rho) := \phi_{e_k} \circ \dots \circ \phi_{e_1}$ alors que dans le cas du problème d'accessibilité bornée avec borne supérieure faible nous avons une énergie telle que $\phi^*(\rho)_i = \max(\phi(\rho)_i, U_i)$, avec $\phi(\rho)$ l'énergie de ρ comme définie précédemment et U la borne supérieure faible. Intuitivement cela revient à dire que la borne supérieure est indépassable et que si un arc devait faire augmenter l'énergie d'un chemin à plus de U alors il fait augmenter l'énergie du chemin à U à la place, comme une batterie qui ne pourrait pas excéder sa charge maximale par exemple.

1.3.4 Problèmes multi-dimensionnels

Dans les graphes non paramétrés

Les problèmes auxquels nous nous intéressons ont été étudiés sur les graphes multi-dimensionnels de dimension au moins égale à 2 dans leurs versions non paramétrées. Nous savons grâce à [Fah+11] que la version de dimension quelconque du problème $EnTot_L$ est EXPSpace-complète et que celles de $EnTot_{LU}$ et $EnTot_{LU}^*$ sont PSPACE-complètes. Les versions multi-dimensionnelles dont la dimension est fixée ont également été considérées et les problèmes $EnTot_L$ dont l'énergie est de dimension k sont NP-difficiles [Fah+11] et sont dans k -EXPTIME [BJK10]. Les variantes des problèmes $EnTot_{LU}$ de dimension k et celles de $EnTot_{LU}^*$ lorsque la dimension est au moins 4 ($k \geq 4$) sont PSPACE-complètes [Fah+11] et les versions de $EnTot_{LU}^*$ pour $k \in \{2, 3\}$ sont NP-difficiles et dans PSPACE. Nous résumons ces différents résultats dans le tableau suivant :

	Dimension quelconque	Dimension k fixée	
$EnTot_L$	EXPSpace-complet [Fah+11]	NP-difficile [Fah+11] et k -EXPTIME [BJK10]	
$EnTot_{LU}$	PSPACE-complet [Fah+11]	PSPACE-complet [Fah+11]	
$EnTot_{LU}^*$	PSPACE-complet [Fah+11]	$2 \leq k \leq 3$ NP-difficile et PSPACE [Fah+11]	$k \geq 4$ PSPACE-complet [Fah+11]

Dans les jeux sur des graphes non paramétrés

De plus, les problèmes que nous considérons ont également été étudiés dans leurs variantes adaptées aux cas des jeux d'énergie à plusieurs dimensions, l'ajout de dimensions supplémentaires à l'énergie permettant de modéliser des objectifs ne se focalisant pas sur un seul critère mais plusieurs. Une énergie à deux dimensions peut notamment représenter

deux ressources énergétiques différentes (électrique et thermique par exemple). C'est le cas dans [Bou+17] où le problème de la variation moyenne d'énergie dans lequel on ajoute des bornes au chemin est prouvé comme appartenant à $\text{NEXPTIME} \cap \text{coNEXPTIME}$. Notons que ce problème est l'application aux jeux d'énergie à plusieurs dimensions du problème d'accessibilité bornée avec borne supérieure faible en conjonction avec des contraintes sur la variation d'énergie moyenne. C'est le problème analogue à celui de la case avec un point d'interrogation du tableau présenté dans la section précédente. Nous pouvons remarquer que ce genre de résultat peut donner des pistes de réflexion concernant l'approche des problèmes dans leurs versions paramétrées. Les variantes adaptées aux jeux d'énergie multi-dimensionnelle des problèmes $EnTot_L$, $EnTot_{LU}$ et $EnTot_{LU}^*$ ont également été considérées. Que la dimension soit quelconque ou fixée et supérieure ou égale à 2, nous avons que la version de $EnTot_L$ est EXPSPACE-difficile [Fah+11] et celles de $EnTot_{LU}$ et $EnTot_{LU}^*$ sont EXPTIME-complètes [Fah+11]. Les variantes des problèmes $EnTot_L$ dont l'énergie est de dimension k sont également dans k -EXPTIME [BJK10]. Ces derniers résultats sont récapitulés dans le tableau suivant :

	Dimension quelconque	Dimension k fixée (≥ 2)
$EnTot_L$	EXPSPACE-difficile [Fah+11]	EXPSPACE-difficile [Fah+11] et k -EXPTIME [BJK10]
$EnTot_{LU}$	EXPSPACE-complet [Fah+11]	EXPSPACE-complet [Fah+11]
$EnTot_{LU}^*$	EXPSPACE-complet [Fah+11]	EXPSPACE-complet [Fah+11]

Dans les jeux sur des graphes paramétrés

Des versions paramétrées des problèmes $EnTot_L$, $EnTot_{LU}$ et $EnTot_{LU}^*$ dans les cas où l'énergie est multi-dimensionnelle ont été étudiées dans leur application aux jeux d'énergie dans [JGR13]. Une contrainte est cependant imposée aux arcs, lorsqu'un ou plusieurs paramètres interviennent dans l'énergie d'un arc, cette énergie ne peut pas être négative. Les auteurs montrent qu'avec cette condition les problèmes $EnTot_L$, $EnTot_{LU}$ et $EnTot_{LU}^*$ sont décidables quel que soit le nombre de paramètres.

Plusieurs résultats d'indécidabilité sont également prouvés dans [Bou+17]. Le problème de la variation moyenne d'énergie est indécidable pour les jeux d'énergie lorsque la dimension de l'énergie est supérieure ou égale à 3. Le problème de la variation moyenne d'énergie avec borne inférieure est quant à lui indécidable pour ces jeux lorsque la dimension de l'énergie est supérieure ou égale à 2. Ces preuves se basent sur la réduction de problèmes connus pour être indécidables aux problèmes étudiés. La première preuve est

obtenue en réduisant le problème à une version particulière du problème de l'arrêt appelée jeu robotique bidimensionnel [NPR16] et la seconde est une réduction au problème étudié du problème de l'arrêt dans les machines à deux compteurs. Notons que ces résultats d'indécidabilité ne peuvent pas s'étendre naturellement aux graphes dont l'énergie est multi-dimensionnelle mais peuvent être de bons points de départ dans la recherche de la décidabilité de telles questions.

Une autre approche est adoptée pour la généralisation aux graphes et aux jeux à plusieurs dimensions d'énergie des problèmes $EnTot_L$, $EnTot_{LU}$ et $EnTot_{LU}^*$ dans [Fah+11] ainsi que dans [JGR13] et [Cha+10b]. En effet le problème abordé est celui de la recherche des bornes supérieures telles qu'il existe un chemin vérifiant les contraintes dans le graphe ou que le joueur 1 soit gagnant dans le jeu. Dans [JGR13], les problèmes $EnTot_{LU}$ et $EnTot_{LU}^*$ sont démontrés décidables et il est prouvé que la version du problème $EnTot_{LU}$ dans les jeux à énergie multi-dimensionnelle lorsqu'on cherche quelles sont les bornes supérieures telles que le joueur 1 ait une stratégie gagnante appartient à $2k$ -EXPTIME où k est le nombre de dimensions de l'énergie. Les versions de $EnTot_L$, $EnTot_{LU}$ et $EnTot_{LU}^*$ étudiées sur les graphes sont quant à elles paramétrées, les paramètres étant le crédit initial ainsi que la borne supérieure dans le cas des problèmes $EnTot_{LU}$ et $EnTot_{LU}^*$, l'objectif étant de savoir s'il existe des valeurs de paramètres telles qu'il y ait un chemin vérifiant les contraintes sur l'énergie. Cette variante du problème $EnTot_L$ est décidable en temps polynomial quelle que soit la dimension de l'énergie [Cha+10b] et les variantes des problèmes $EnTot_{LU}$ et $EnTot_{LU}^*$ sont décidables en temps polynomial si les paramètres correspondant à la borne supérieure n'interviennent pas dans les énergies des arcs quelle que soit la dimension de l'énergie [Fah+11]. Si les paramètres correspondant à la borne supérieure peuvent apparaître dans les énergies des arcs, les variantes du problème $EnTot_{LU}$ sont indécidables quelle que soit la dimension (supérieure à 2) de l'énergie [Fah+11] et les versions des problèmes $EnTot_{LU}^*$ sont indécidables lorsque l'énergie est de dimension supérieure ou égale à 4 [Fah+11], la décidabilité des problèmes de dimensions 2 et 3 étant toujours ouverte. Ces derniers résultats sont présentés dans le tableau suivant :

	Paramètres de U absents des énergies	Paramètres de U dans les énergies	
$EnTot_{LU}$	P [Fah+11]	Indécidable pour $k \geq 2$ [Fah+11]	
$EnTot_{LU}^*$	P [Fah+11]	$2 \leq k \leq 3$?	$k \geq 4$ Indécidable [Fah+11]

Dans les automates temporisés

Des problèmes multi-dimensionnels ont également été étudiés pour les jeux sur des automates temporisés. Les variantes des problèmes $EnTot_L$, $EnTot_{LU}$ et $EnTot_{LU}^*$ ont été prouvées comme indécidables dans les jeux sur les automates temporisés multidimensionnels à une horloge quelle que soit la dimension (supérieure ou égale à 2) [Fah+11]. Concernant les variations de ces problèmes appliquées aux automates temporisés multidimensionnels à une horloge, les versions du problème $EnTot_L$ sont indécidables quelle que soit la dimension [Fah+11] et les problèmes $EnTot_{LU}$ et $EnTot_{LU}^*$ sont indécidables pour les dimensions supérieures ou égales à 4 [Fah+11]. Nous résumons ces résultats dans le tableau suivant :

	Dimension $k \in \{2; 3\}$	Dimension $k \geq 4$
$EnTot_L$	Indécidable [Fah+11]	Indécidable [Fah+11]
$EnTot_{LU}$?	Indécidable [Fah+11]
$EnTot_{LU}^*$?	Indécidable [Fah+11]

Une autre approche est adoptée dans [LR05], en effet les auteurs montrent la décidabilité de l'accessibilité dans les automates temporisés multi-dimensionnels lorsqu'on cherche à minimiser l'énergie d'une dimension donnée et que l'on souhaite que chaque autre dimension de l'énergie ne dépasse pas une certaine borne. Ces travaux sont poursuivis dans [LR08] qui apporte la preuve de la décidabilité du problème de l'accessibilité dans les automates temporisés telle que chaque dimension de l'énergie soit minimale. Nous pouvons remarquer que la résolution de ce problème implique de trouver toutes les solutions qui ne sont pas comparables entre elles, en effet nous pouvons par exemple dire que $(0, 2) < (1, 3)$ mais nous ne pouvons pas comparer $(0, 2)$ et $(1, 1)$.

Un autre aspect des problèmes multi-dimensionnels est abordé dans [BBL08]. En effet les auteurs s'intéressent au cas des chemins infinis dans les automates temporisés à deux dimensions en considérant une notion de chemin d'énergie minimale étendue avec la recherche de chemins minimisant le ratio de la première dimension de l'énergie par sa seconde dimension. Les auteurs prouvent la décidabilité de ce problème et montrent qu'il est PSPACE-complet.

1.4 Contributions et plan

La prolongation des résultats de [KO81] aux graphes paramétrés avec de multiples paramètres rend le problème beaucoup plus complexe puisqu'il ne faut plus partitionner un espace monodimensionnel mais un espace réel de dimension n (où n est le nombre de paramètres). Le chapitre 2 se concentre sur l'extension du calcul des plus courts chemins depuis un sommet donné à tous les autres sommets d'un graphe paramétré à une seule dimension comprenant un nombre quelconque de paramètres. Nous présentons un algorithme et nous prouvons qu'il a une complexité temporelle exponentielle dans le nombre de sommets et polynomiale dans le nombre d'arcs ainsi que dans le nombre de paramètres du graphe considéré et permet de considérer des graphes avec un nombre quelconque de paramètres. Ce travail a abouti à la publication d'un article à la conférence LATA 2021 [JLS21b] ayant obtenu le prix du meilleur article de la conférence.

L'accessibilité bornée avec contraintes ayant été largement étudiée dans un cadre non paramétré, nous avons cherché à étendre les résultats concernant le problème d'accessibilité bornée au cas des graphes paramétrés. Cependant il n'existe pas une unique manière de paramétrer le problème d'accessibilité bornée. En effet, l'espace des valeurs des paramètres peut différer et nous nous concentrons, dans le chapitre 3, sur la preuve de la décidabilité et de la complexité PSPACE-complète du problème d'accessibilité bornée dans les graphes paramétrés multi-dimensionnels de dimension quelconque lorsque les paramètres prennent des valeurs entières (section 3.2). Nous prouvons également l'indécidabilité du problème d'accessibilité bornée dans les graphes paramétrés multi-dimensionnels de dimension supérieure ou égale à 2 lorsque l'espace des valeurs des paramètres contient \mathbb{Q} dès lors qu'un paramètre intervient dans l'énergie des arcs (section 3.3).

Nous nous intéressons ensuite à l'accessibilité avec borne inférieure ainsi qu'au problème à parité avec borne inférieure dans les graphes paramétrés à une seule dimension. Dans un travail effectué avec Stéphane Le Roux, nous prouvons dans le chapitre 4 la décidabilité de l'accessibilité avec borne inférieure dans les graphes paramétrés à une seule dimension ainsi qu'au problème à parité avec borne inférieure dans ces mêmes graphes. Nous généralisons ensuite ces résultats aux cas des jeux de parité à énergie paramétrée et nous prouvons des bornes de la complexité des problèmes à un et à deux joueurs. Le problème à un joueur est NP-complet et le problème à deux joueurs est PSPACE-difficile ainsi que NEXPTIME [Jez+on].

En parallèle des travaux rapportés dans ce manuscrit nous avons également travaillé

sur la question de l'accessibilité dans les réseaux de Petri ce qui a abouti à la publication d'un article à la conférence Petri Nets 2021 [JLS21a] dans lequel nous proposons un algorithme paresseux de calcul de l'accessibilité dans les réseaux de Petri avec et sans arcs inhibiteurs. Nous prouvons cet algorithme ainsi que sa complexité et nous discutons également les résultats expérimentaux d'une implémentation de cet algorithme dans le *model-checker* ROMEO [Lim+09].

ARBRES DES PLUS COURTS CHEMINS

2.1 Introduction

Comme nous l'avons déjà expliqué précédemment, des graphes dans lesquels les énergies sont exprimées comme une fonction d'un seul paramètre ont déjà été étudiés [KO81]. L'objectif étant de partitionner les nombres réels de façon finie, de telle sorte que pour deux valeurs de paramètre dans une partition donnée, les chemins optimaux depuis un certain sommet source vers tous les autres sommets soient les mêmes. Ces chemins optimaux sont donnés sous la forme d'un arbre couvrant du graphe. La prolongation de ces résultats aux graphes paramétrés avec plusieurs paramètres rend le problème beaucoup plus complexe puisqu'il faut partitionner un espace réel de dimension n (où n est le nombre de paramètres). La difficulté vient notamment du fait que lorsque la dimension devient supérieure ou égale à deux, nous ne pouvons plus chercher à trouver l'unique borne supérieure d'un intervalle, élément central de l'approche de [KO81], puisqu'il n'y a plus d'intervalle.

Nous proposons dans ce chapitre un algorithme prenant en entrée un graphe mono-dimensionnel orienté à énergie paramétrée G ainsi qu'un sommet s de G et renvoyant en sortie une liste d'arbres associés à des zones de l'espace des paramètres. Chaque arbre de la liste contient tous les sommets du graphe et est associé à une zone contenant des valeurs de paramètres pour lesquelles l'arbre représente les chemins d'énergie minimale dans le graphe. Les zones renvoyées par l'algorithme ne se recouvrent que sur leur éventuelle frontière en commun et l'algorithme prend en compte le cas où il n'existe pas de chemin d'énergie minimale c'est-à-dire les valeurs de paramètres pour lesquelles il existe au moins un cycle faisant strictement diminuer l'énergie. Cet algorithme a une complexité temporelle exponentielle dans le nombre de sommets et est polynomial dans le nombre d'arcs ainsi que dans le nombre de paramètres du graphe considéré et permet de considérer des graphes avec un nombre quelconque de paramètres. Ce travail a été l'objet d'un article publié à LATA 2021 et ayant obtenu le prix du meilleur article de la conférence [JLS21b].

Ce chapitre est organisé de la manière suivante : dans la Section 2.2 nous introduisons les notations utilisées ainsi que les définitions induites par les spécificités de notre problème ; dans la Section 2.3 nous présentons notre algorithme de façon informelle à l'aide d'un exemple complet ; dans la Section 2.4, nous développons l'algorithme avec les preuves associées de correction, de complétude, de terminaison et de complexité ; dans la Section 2.6 nous concluons.

2.2 Définitions

Dans cette section nous introduisons les spécificités liées aux problèmes que nous considérons ainsi qu'une définition simplifiée des graphes. Nous définissons ensuite de nouveaux objets et notations qui seront utiles pour la résolution du problème.

2.2.1 Graphes paramétrés

Dans ce chapitre nous prendrons $\mathbb{A} = \mathbb{R}$. De plus, avec un léger abus de notations, nous noterons λ_i non seulement les paramètres mais également leurs valeurs. Nous utiliserons également la notation $\vec{\lambda} = (\lambda_1 \dots \lambda_n)$.

Nous ne considérerons dans ce chapitre que des graphes $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ à une seule dimension, c'est-à-dire tels que $d = 1$. Les coefficients des arcs seront tous égaux à 1 : $\forall e \in E, \mu_e = 1$. De plus pour tout $e \in E, \lambda \in \mathbb{A}^n$ nous aurons $\alpha_e(\vec{\lambda}) = f(e) + \sum_{i=1}^n \lambda_i \Lambda_i(e)$. Avec f une fonction à valeurs réelles et Λ_i une fonction à valeurs dans $\{0, -1, 1\}$. De plus nous ne considérerons que des chemins finis, les priorités π_e ne seront donc pas considérées.

Nous avons représenté un exemple de graphe que nous allons considérer dans ce chapitre sur la figure 2.1. Nous pouvons constater que dans les différentes fonctions α_e les λ_i ne sont pas multipliés entre eux. Les paramètres ne sont pas non plus multipliés par d'autres entiers que 0, 1 et -1 . Nous faisons cela sans perte de généralité. En effet un arc qui ajoute $2\lambda_1$ à la valeur de l'énergie peut être représenté par deux arcs ajoutant chacun λ_1 à l'énergie ainsi qu'un sommet intermédiaire n'intervenant dans aucun autre arc du graphe.

Nous obtenons la définition suivante pour les graphes mono-dimensionnels orientés à énergie paramétrée, que nous appellerons *graphes paramétrés* par soucis de concision : un *graphe paramétré* avec n paramètres sera le 4-uplet $G = (V, E, n, (\alpha_e)_{e \in E})$.

Étant données les simplifications des coefficients et des fonctions d'énergie des arcs

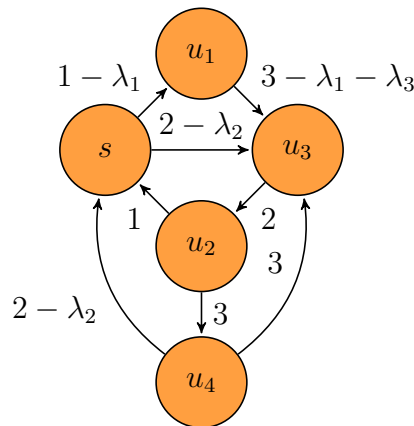


FIGURE 2.1 – Un exemple de graphe paramétré mono-dimensionnel à trois paramètres.

nous pouvons redéfinir les énergies des arcs et des chemins comme suit :

Définition 2.1 (Énergie d'un arc). Soit $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré, l'énergie d'un arc $e \in E$ est de la forme :

$$\alpha_e(\vec{\lambda}) = f_e + \sum_{i=1}^n \lambda_i \Lambda_{i,e}.$$

Avec $f_e \in \mathbb{R}$ la part de l'énergie ne dépendant pas des paramètres et avec $\Lambda_{i,e} \in \{-1, 0, 1\}$.

Définition 2.2 (Énergie d'un chemin). Soit $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré, soit $\rho = e_1 \dots e_k$ un chemin de G , l'énergie de ρ est :

$$\phi(\rho, \vec{\lambda}) = \sum_{j=1}^k \alpha_{e_j}(\vec{\lambda}) = \sum_{j=1}^k \left(f(e_j) + \sum_{i=1}^n \lambda_i \Lambda_i(e_j) \right) = \sum_{j=1}^k f(e_j) + \sum_{i=1}^n \lambda_i \sum_{j=1}^k \Lambda_i(e_j).$$

Définition 2.3 (Chemin d'énergie minimale). Soit $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré. Soient ρ un chemin de G et $\vec{\lambda} \in \mathbb{R}^n$. Le chemin ρ est un chemin d'énergie minimale pour $\vec{\lambda}$ si et seulement si pour tout chemin ρ' ayant les mêmes sommets initiaux et finaux que ρ nous avons : $\phi(\rho, \vec{\lambda}) \leq \phi(\rho', \vec{\lambda})$.

Il ne peut y avoir des chemins d'énergie minimale que s'il n'existe pas de cycle diminuant strictement l'énergie. Nous définissons les cycles négatifs qui sont les cycles diminuant strictement l'énergie.

Définition 2.4 (Cycle négatif). Soit $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré. Soient c un cycle de G et $\vec{\lambda} \in \mathbb{R}^n$. Nous appelons énergie de c la valeur dont est modifiée l'énergie après avoir parcouru c , c'est-à-dire la valeur $\phi(c, \vec{\lambda})$.

Le cycle c sera dit négatif pour $\vec{\lambda}$ si et seulement si son énergie est strictement négative.

2.2.2 Arbres sur des graphes paramétrés

Par la suite, lorsque nous considérerons un graphe $G = (V, E, n, (\alpha_e)_{e \in E})$, nous nous intéresserons aux chemins d'énergie minimale d'un sommet s donné vers tous les autres sommets de V . Nous supposons que de tels chemins existent dans G .

Définition 2.5 (Arbre). Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et un sommet $s \in V$. Nous définissons un arbre de G enraciné en s (ou de source s) comme le graphe particulier représenté par le 3-uplet $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ tel que :

- $V_T = V$,
- $E_T \subseteq E$ est tel que pour $v \in V \setminus \{s\}$, $|\{(u, v) : (u, v) \in E_T\}| = 1$, et pour tout $\{(u, s) : (u, s) \in E_T\} = \emptyset$ et il n'y a pas de cycle dans T .

Nous noterons que de tels arbres comportent systématiquement $|V| - 1$ arcs. De plus, il y a toujours exactement un chemin depuis s vers chaque sommet.

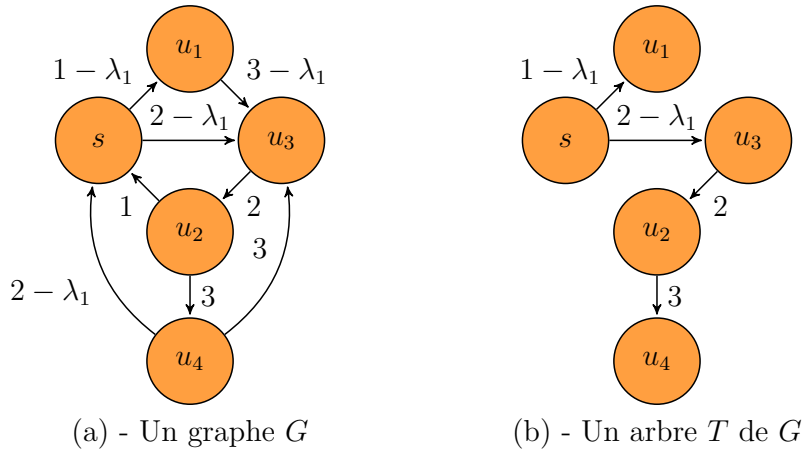


FIGURE 2.2 – Un graphe G et un exemple d'un arbre T de G .

Dans la Figure 2.2 (a), nous montrons un exemple G d'un graphe avec un paramètre, λ_1 . Nous représentons T dans la Figure 2.2 (b), un exemple d'arbre de G enraciné en s .

Définition 2.6 (Énergie). Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et un sommet $s \in V$. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit v un sommet de G , l'énergie $d(T, v, \vec{\lambda})$ est l'énergie de l'unique chemin de s à v dans T .

Définition 2.7 (Énergies partielles). Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et un sommet $s \in V$. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit v

un sommet de G et soit $e_1 \dots e_k$ l'unique chemin de s à v dans T . L'énergie partielle non paramétrée entre s et v dans T est :

$$d_f(T, v) = \sum_{j=1}^k f(e_j).$$

Pour tout $i \in \llbracket 1, n \rrbracket$, les énergies partielles paramétrées entre s et v sont :

$$d_{\Lambda_i}(T, v) = \sum_{j=1}^k \Lambda_i(e_j).$$

Nous pouvons remarquer que $d(T, v, \vec{\lambda}) = d_f(T, v) + \sum_{i=1}^n \lambda_i d_{\Lambda_i}(T, v)$.

Définition 2.8 (Arbre d'énergies minimales). Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Nous disons que T est un arbre d'énergies minimales pour $\vec{\lambda} \in \mathbb{R}^n$ si et seulement si pour tout $v \in V$, l'unique chemin de s à v dans T est un chemin d'énergie minimale pour $\vec{\lambda}$ de s à v dans G .

De plus, soit $S \subseteq \mathbb{R}^n$ tel que T est un arbre d'énergies minimales pour tout $\vec{\lambda} \in S$, nous dirons que T est un arbre d'énergies minimales sur S .

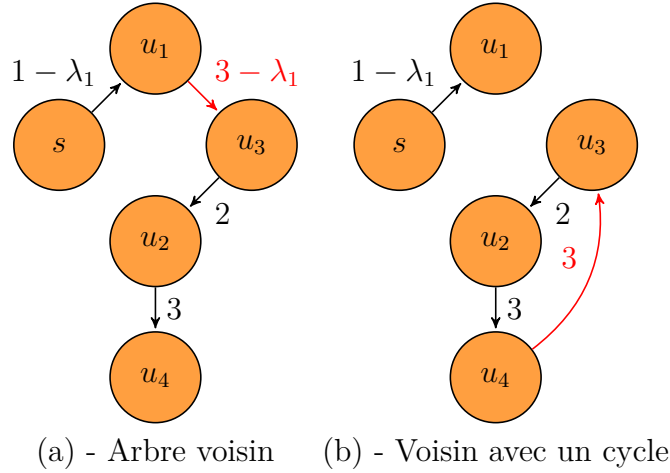
Définition 2.9 (Voisins). Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit $e = (u, v)$ un arc de E , le voisin de T généré par e est le n -uplet $N(T, e) = (V_N, E_N, \alpha_e)_{e \in E_N}$ tel que :

- $V_N = V$,
- $E_N = (E_T \setminus \{(u', v) : (u', v) \in E_T\}) \cup \{e\}$.

En d'autres termes, $N(T, e)$ est obtenu à partir de T en retirant le seul arc $e' = (u', v')$ tel que $v' = v$ et en ajoutant e .

Nous pouvons tout d'abord remarquer que pour tout arc $e \in E_T$, $N(T, e) = T$. De plus notons que pour un arc $e' = (u, v)$ qui n'est pas dans un arbre ($e' \in E \setminus E_T$), un arc dirigé vers le sommet v n'existe pas nécessairement dans T . En effet un arc e tel que $e = (u, s)$ ne sera jamais dans T (T étant un arbre enraciné en s) et peut générer un voisin de T comme n'importe quel autre arc. En particulier, cela signifie que le voisin d'un arbre n'est pas nécessairement un arbre.

Nous pouvons par exemple considérer le graphe G de la Figure 2.2 et prendre $e = (s, u_3)$, $e_1 = (u_1, u_3)$ et $e_2 = (u_4, u_3)$. La Figure 2.2 (b) représente un arbre T enraciné en


 FIGURE 2.3 – Deux exemples de voisins de T dans G de la Figure 2.2.

s , la Figure 2.3 (a) représente $N(T, e_1)$, qui est également un arbre, et la Figure 2.3 (b) représente $N(T, e_2)$, qui n'est pas un arbre.

Les propositions suivantes précisent les cas dans lesquels un voisin d'un arbre sera effectivement un arbre.

Proposition 2.1. Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soient $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s et $e = (u, v) \in E$ un arc. $N(T, e)$ est un arbre si et seulement si v n'est pas sur l'unique chemin de s à u dans T .

Démonstration. Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soient $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s et $e = (u, v)$ un arc.

Dans le cas où $e = (u, s)$ la proposition est vraie parce que : (1) s est sur le chemin de s à n'importe quel sommet ; (2) aucun arc ne sera retiré de T pour construire $N(T, e)$ puisqu'il n'y a aucun sommet v de T tel que (v, s) soit un arc de T , par définition de T ; et (3) il y a un chemin de s à u dans T par définition, par conséquent ajouter e à T créera un cycle. Dans le reste de cette preuve nous considérerons que $v \neq s$.

Dans le cas où $e = (u, u)$ le lemme est vrai parce que : (1) u est sur le chemin de s à u , et (2) ajouter e à T créera nécessairement un cycle. Dans le reste de cette preuve nous considérerons que $v \neq u$.

(\Rightarrow) Supposons que v est sur le chemin ρ de s à u dans T . Nous avons $\rho = e_1 \dots e_k$ et $\forall i \in \llbracket 1, k \rrbracket$, $e_i = (v_i, v_{i+1})$ tel que $v_1 = s$, $v_{k+1} = u$, et $\exists j \in \llbracket 2, k \rrbracket$ tel que $v_j = v$. Sachant que $N(T, e)$ est obtenu en retirant e_{j-1} et en ajoutant e à T . Puisque $e = (u, v)$ n'apparaît pas dans T et que $v_j = v$, le chemin $\rho' = ee_j e_{j+1} \dots e_k$ est un chemin de $N(T, e)$. De plus,

comme $v_{k+1} = u$, le sommet initial et le sommet final de ρ' sont identiques, et nous avons donc que ρ' est un cycle de $N(T, e)$. Par conséquent, si v est sur l'unique chemin de s à u dans T alors $N(T, e)$ n'est pas un arbre.

(\Leftarrow) Supposons que v n'est pas sur le chemin de s à u dans T . Nous avons $\rho = e_1 \dots e_k$ et pour tout $i \in \llbracket 1, k \rrbracket$, $e_i = (v_i, v_{i+1})$ tel que $v_1 = s$, $v_{k+1} = u$, et $\forall i \in \llbracket 1, k+1 \rrbracket$, $v_i \neq v$. Pour tout $w \in V$, soit ρ' le chemin de s à w dans T . $\rho' = e'_1 \dots e'_{k'}$ et pour tout $i \in \llbracket 1, k' \rrbracket$, $e'_i = (v'_i, v'_{i+1})$ tel que $v'_1 = s$ et $v'_{k'+1} = w$. Pour obtenir $N(T, e)$ nous retirons de T le seul arc $e_r = (u_r, v_r)$ tel que $v_r = v$, un tel arc existe car $v \neq s$. Dans le cas où v n'apparaît pas dans ρ' , cela signifie que $\forall i \in \llbracket 1, k'+1 \rrbracket$, $v'_i \neq v$, et donc que e_r n'apparaît pas dans ρ' . Par conséquent ρ' existe dans $N(T, e)$: il y a ainsi un chemin de s à w dans $N(T, e)$. Dans le cas où v apparaît dans ρ' , cela signifie que $\exists j \in \llbracket 1, k' \rrbracket$ tel que $v'_j = v$. En d'autres termes, nous avons $e_r = e'_{j-1}$. Par conséquent, le chemin $\rho_w = e_1 \dots e_k e e'_j e'_{j+1} \dots e'_{k'}$ est un chemin de $N(T, e)$. De plus $v_1 = s$ et $v'_{k'+1} = w$, nous avons donc un chemin de s à w dans $N(T, e)$.

En conclusion, dans tous les cas nous avons :

- (a) Un chemin de s à w dans $N(T, e)$, pour tout sommet $w \in V$.
- (b) T et $N(T, e)$ ont le même nombre d'arcs. En effet, exactement un arc (e_r) a été retiré de T et exactement un arc (e) a été ajouté à T dans le processus de construction de $N(T, e)$.

En considérant (a) d'une part et (b) d'autre part, nous pouvons conclure que $N(T, e)$ est un arbre. \square

Avant de pouvoir donner d'autres propriétés des arbres et de leurs voisins, nous avons besoin de définir quelques notations. Pour un arc $e = (u, v) \in E$ nous noterons :

- $\Delta_f(T, e) = d_f(T, u) + f(e) - d_f(T, v)$,
- $\forall i \in \llbracket 1, n \rrbracket$, $\Delta_{\Lambda_i}(T, e) = d_{\Lambda_i}(T, u) + \Lambda_i(e) - d_{\Lambda_i}(T, v)$.

Ces deltas représentent les différences d'énergie de s à v entre T et son voisin généré par e . Δ_f est la différence de la part non paramétrée de l'énergie. Chaque Δ_{Λ_i} représente les différences du nombre d'occurrences du paramètre λ_i correspondant. Nous pouvons remarquer que pour tout arc $e \in E_T$, nous avons toujours $\Delta_{\Lambda_i}(T, e) = 0$.

Par exemple dans la Figure 2.2, avec les mêmes notations que précédemment nous avons : $\Delta_f(T, e_1) = d_f(T, u_1) + f(e_1) - d_f(T, u_3) = 1 + 3 - 2 = 2$, et $\Delta_{\Lambda_1}(T, e_1) = d_{\Lambda_1}(T, u_1) + \Lambda_1(e_1) - d_{\Lambda_1}(T, u_3) = 1 + 1 - 1$.

Nous nous intéressons maintenant aux propriétés des voisins d'un arbre et lorsque nous avons $\Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e) = 0$. Nous montrons que, quand cela est vérifié, si

le voisin est un arbre alors les énergies des sommets sont les mêmes dans l'arbre et dans son voisin, et si le voisin n'est pas un arbre alors il a un cycle d'énergie nulle.

Proposition 2.2. *Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit $e = (u, v)$ un arc tel que $e \notin E_T$. Si $N(T, e)$ est un arbre alors $\forall w \in V, d(T, w, \vec{\lambda}) = d(N(T, e), w, \vec{\lambda})$ si et seulement si $\Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e) = 0$.*

Démonstration. Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit $e = (u, v) \in E \setminus E_T$ un arc tel que $N(T, e)$ est un arbre.

Pour tous les sommets w tels que v n'est pas sur l'unique chemin de s à w dans T nous avons $\forall \vec{\lambda} \in \mathbb{R}^n, d(T, w, \vec{\lambda}) = d(N(T, e), w, \vec{\lambda})$. En effet, dans ce cas, l'arc $e_r = (u_r, v)$ qui est retiré de T afin d'obtenir $N(T, e)$ n'apparaît pas dans le chemin de s à w .

Considérons maintenant les sommets w tels que v est sur le chemin de s à w dans T . Soient ρ_u le chemin de s à u dans T , ρ_v le chemin de s à v dans T , et ρ_w le chemin de v à w dans T . Le chemin de s à w est $\rho_v \rho_w$ et le chemin de s à w dans T' est $\rho_u e \rho_w$. Par conséquent, les valeurs de $\vec{\lambda}$ telles que $d(T, w, \vec{\lambda}) = d(N(T, e), w, \vec{\lambda})$ sont les mêmes que les valeurs telles que $d(T, v, \vec{\lambda}) = d(N(T, e), v, \vec{\lambda})$. En d'autres termes, ce sont les valeurs de $\vec{\lambda}$ telles que l'énergie de ρ_v soit la même que l'énergie de ρ_u plus l'énergie de e . De plus, pour tout $i \in \llbracket 1, n \rrbracket$ nous avons $d_{\Lambda_i}(N(T, e), v) = d_{\Lambda_i}(T, v) + \Delta_{\Lambda_i}(T, e)$ et $d_f(N(T, e), v) = d_f(T, v) + \Delta_f(T, e)$. Ainsi, $d(N(T, e), v, \vec{\lambda}) = d_f(T, v) + \Delta_f(T, e) + \sum_{i=1}^n \lambda_i (d_{\Lambda_i}(T, v) + \Delta_{\Lambda_i}(T, e))$.

Il en découle que $d(N(T, e), v, \vec{\lambda}) = d(T, v, \vec{\lambda}) + \Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e)$.

Par conséquent, les valeurs de $\vec{\lambda}$ telles que $d(N(T, e), v, \vec{\lambda}) = d(T, v, \vec{\lambda})$ sont celles vérifiant $\Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e) = 0$. \square

Proposition 2.3. *Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit $e = (u, v) \in E$ un arc. Si $N(T, e)$ n'est pas un arbre alors $N(T, e)$ a un cycle d'énergie 0 si et seulement si $\vec{\lambda}$ est tel que $\Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e) = 0$.*

Démonstration. Soient $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré et $s \in V$ un sommet. Soit $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s . Soit $e = (u, v)$ un arc tel que $N(T, e)$ n'est pas un arbre. Cela signifie que v est sur le chemin de s à u dans T par la

Proposition 2.1, d'où le fait qu'un cycle de G soit formé par le chemin de v à u auquel nous ajoutons l'arc e . L'énergie de ce cycle est : $d_f(T, u) - d_f(T, v) + f(e) + \sum_{i=1}^n \lambda_i (d_{\Lambda_i}(T, u) - d_{\Lambda_i}(T, v) + f_{\Lambda_i}(e)) = \Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e)$.

Par conséquent les valeurs de $\vec{\lambda}$ telles que G a un cycle d'énergie nulle vérifient : $\Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e) = 0$. \square

2.2.3 Contraintes et zones associées aux arbres

Pour les définitions suivantes nous considérerons $G = (V, E, n, (\alpha_e)_{e \in E})$ un graphe paramétré, $s \in V$ un sommet et $T = (V_T, E_T, (\alpha_e)_{e \in E_T})$ un arbre de G enraciné en s .

Définition 2.10 (Contrainte). *Soit $e \in E \setminus E_T$ un arc n'appartenant pas à T , la contrainte associée à e est :*

$$C_{T,e} : \Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e) \geq 0.$$

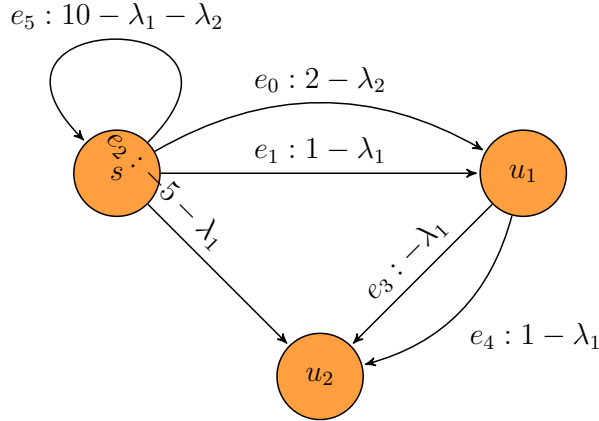
Définition 2.11 (Zone). *Soit $E_c \subseteq E \setminus E_T$ un ensemble d'arcs tel que $\forall e_c \in E_c, \exists i \in \llbracket 1, n \rrbracket, \Delta_{\Lambda_i}(T, e_c) \neq 0$. La zone définie par les contraintes associées aux arcs de E_c est l'ensemble S tel que : $\vec{\lambda} \in S$ si et seulement si $\forall e_c \in E_c, \Delta_f(T, e_c) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e_c) \geq 0$.*

Nous pouvons remarquer que les zones sont convexes par construction.

Définition 2.12 (Contrainte active). *Soit $E_c \subseteq E \setminus E_T$ un ensemble d'arcs tel que $\forall e_c \in E_c, \exists i \in \llbracket 1, n \rrbracket, \Delta_{\Lambda_i}(T, e_c) \neq 0$. Soit $e_c \in E_c$ un arc. Soit S la zone définie par les contraintes associées aux arcs de E_c . Soit $S_{/e_c}$ la zone définie par les contraintes associées aux arcs de $E_c \setminus \{e_c\}$. La contrainte C_{T,e_c} est dite active si et seulement si $S_{/e_c} \neq S$.*

2.3 Présentation de notre algorithme de calcul des énergies minimales

Dans cette section, nous illustrons sur un exemple l'algorithme avant de le présenter formellement dans la section suivante. Notre algorithme prend en entrée un graphe G et renvoie une liste d'arbres et une liste de zones disjointes à leur frontière près. Chaque arbre T est associé à une zone S , telle que T est un arbre d'énergies minimales sur S . De plus, l'union des zones renvoyées est une zone de \mathbb{R}^n pour laquelle il n'existe pas de cycle


 FIGURE 2.4 – Un graphe G_{ex} .

négatif dans G . Dans cette section, nous montrons comment cet algorithme fonctionne sur l'exemple G_{ex} , un graphe paramétré avec deux paramètres, présenté en Figure 2.4.

La première étape de l'algorithme consiste à trouver la zone $Z_{cycles \geq 0}$ pour laquelle le concept d'énergie minimale est pertinent. Il s'agit des valeurs de λ_1 and λ_2 pour lesquelles il n'existe pas de cycle négatif. Ici, l'unique cycle est $e_5 e_5 e_5 \dots$. Ainsi, le seul cycle négatif possible intervient quand l'énergie de e_5 est négative, donc quand $10 - \lambda_1 - \lambda_2 < 0$. Nous avons donc $Z_{cycles \geq 0} = \{(\lambda_1, \lambda_2) \in \mathbb{R}^2 : \lambda_1 + \lambda_2 \leq 10\}$.

L'objectif sera de couvrir $Z_{cycles \geq 0}$ avec des zones associées à des arbres d'énergies minimales. Pour cela l'algorithme énumère des zones, associées avec des arbres, allant d'une zone à une autre en considérant les arbres voisins des arbres associés. L'algorithme commence par calculer un premier arbre T_0 . Cet arbre est un arbre d'énergies minimales pour une paire $(\lambda_{1,init}, \lambda_{2,init})$ telle qu'il n'existe pas de cycle négatif dans G . Ici nous prenons $(\lambda_{1,init}, \lambda_{2,init}) = (-20, -20)$, en effet cette valeur nous assure qu'aucun arc ne diminuera strictement l'énergie ce qui garantit qu'il n'y a pas de cycle négatif dans le graphe pour ces valeurs des paramètres. Ce choix de T_0 est représenté en Figure 2.5.

Nous regardons maintenant les contraintes associées avec les voisins de T_0 pour caractériser la zone associée S_0 . Les contraintes actives nous permettront également de savoir quels arbres seront considérés par la suite. Ici, les arcs qui génèrent les voisins sont e_0 , e_3 , e_4 et e_5 . Ainsi, il faut considérer les contraintes C_0 , C_3 , C_4 et C_5 associées respectivement à $N(T_0, e_0)$, $N(T_0, e_3)$, $N(T_0, e_4)$ et $N(T_0, e_5)$.

Nous avons $C_0 : 1 + \lambda_1 - \lambda_2 \geq 0$, $C_3 : 6 - \lambda_1 \geq 0$, $C_4 : 7 - \lambda_1 \geq 0$ et $C_5 : 10 - \lambda_1 - \lambda_2 \geq 0$. Parmi ces contraintes, C_0 , C_3 et C_5 sont actives : la zone définie par les contraintes C_0 , C_3 et C_5 est égale à la zone définie par les contraintes C_0 , C_3 , C_4 et C_5 . Par conséquent,

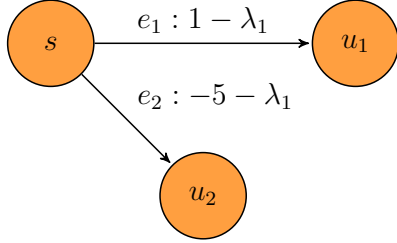


FIGURE 2.5 – T_0 .

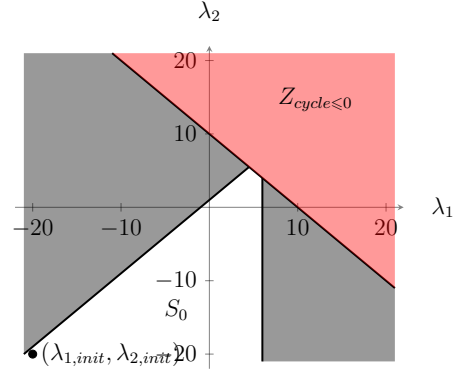


FIGURE 2.6 – $(\lambda_{1,init}, \lambda_{2,init})$ et S_0 .

nous prenons $S_0 = \{(\lambda_1, \lambda_2) \in \mathbb{R}^2 : 1 + \lambda_1 - \lambda_2 \geq 0, 6 - \lambda_1 \geq 0, 10 - \lambda_1 - \lambda_2 \geq 0\}$, comme représenté en figure 2.6. Les parties grises correspondent aux zones que nous n’avons pas encore considérées et les parties blanches sont les zones déjà considérées et sont identifiées par leur nom. Pour l’instant nous n’avons identifié que S_0 donc il n’y a qu’une zone blanche sur la figure 2.6. Les droites noires représentent les contraintes que nous avons identifiées comme actives et la zone en rouge correspond aux valeurs de paramètres pour lesquelles il existe au moins un cycle négatif dans le graphe.

Étant donné que $N(T_0, e_0)$ et $N(T_0, e_3)$ sont associés à des contraintes actives et n’ont pas encore été considérés, ils sont ajoutés dans une liste d’arbres pour être considérés ultérieurement. Cette liste est nommée *listeAFaire*. $N(T_0, e_5)$ n’est pas ajouté à *listeAFaire* car il ne s’agit pas d’un arbre. La paire (T_0, S_0) est ajoutée à une liste nommée *listeExplo*. Cette liste sera renvoyée par l’algorithme à la fin de ses calculs.

À ce moment nous avons donc :

- $listeAFaire = \{N(T_0, e_0), N(T_0, e_3)\}$,
- $listeExplo = \{(T_0, S_0)\}$.

À partir de cette étape, l’algorithme considère de façon itérative les arbres de la liste *listeAFaire*. Considérons, par exemple, que cela débute avec $N(T_0, e_0) = T_1$, représenté sur la Figure 2.7. Les arcs qui génèrent des voisins sont e_1, e_3, e_4 et e_5 . Les contraintes associées sont $C_1 : -1 - \lambda_1 + \lambda_2 \geq 0, C_3 : 7 - \lambda_2 \geq 0, C_4 : 8 - \lambda_2 \geq 0$ et $C_5 : 10 - \lambda_1 - \lambda_2 \geq 0$. Parmi ces dernières, $C_1, C_3,$ et C_5 sont actives. À partir de cela il nous est possible de définir S_1 , représentée sur la Figure 2.8. Étant donné que $N(T_1, e_1) = T_0$ a préalablement été considéré, seul $N(T_1, e_3)$ est ajouté à *listeAFaire* (nous rappelons que $N(T_1, e_5)$ n’est pas un arbre). (T_1, S_1) est ajouté à *listeExplo*.

Nous avons alors :

- $listeAFaire = \{N(T_0, e_3), N(T_1, e_3)\}$,
- $listeExplo = \{(T_0, S_0), (T_1, S_1)\}$.

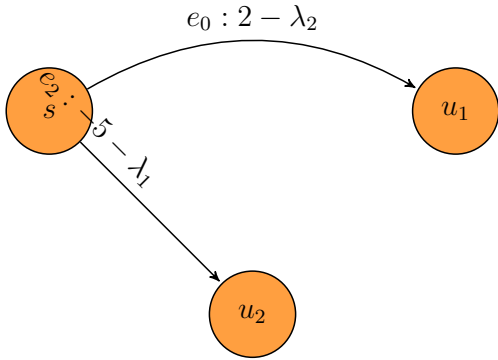


FIGURE 2.7 – T_1 .

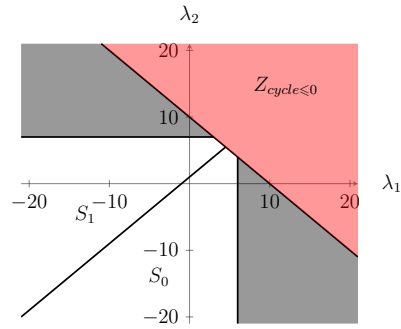


FIGURE 2.8 – S_0 et S_1 .

Imaginons que l'algorithme considère $N(T_0, e_3) = T_2$ par la suite. Cet arbre est représenté sur la Figure 2.9. Les contraintes utilisées afin de définir S_2 sont celles associées à e_0 , e_2 et e_5 . e_4 n'est pas considéré car $\Delta_{\Lambda_1} T_2, e_4 = \Delta_{\Lambda_2} T_2, e_4 = 0$. Toutes les contraintes sont actives. La zone S_2 est représentée sur la Figure 2.10. Aucun arbre n'est ajouté à $listeAFaire$ puisque nous avons $N(T_2, e_0) = N(T_1, e_3)$, qui est déjà inclus dans $listeAFaire$ et $N(T_2, e_2) = T_0$, qui a déjà été considéré. (T_2, S_2) est ajouté à $listeExplo$.

Après cette étape nous avons :

- $listeAFaire = \{N(T_1, e_3)\}$,
- $listeExplo = \{(T_0, S_0), (T_1, S_1), (T_2, S_2)\}$.

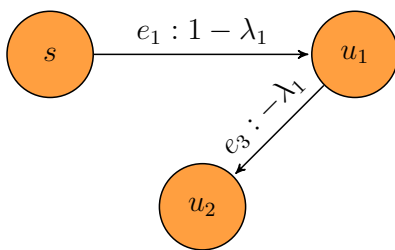


FIGURE 2.9 – T_2 .

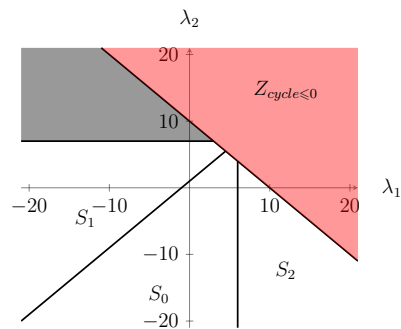


FIGURE 2.10 – S_0 , S_1 et S_2 .

Il nous reste ensuite à considérer $N(T_1, e_3) = T_3$, représenté sur la Figure 2.11. Trois contraintes sont considérées, elles sont cette fois associées à e_1 , e_2 et e_5 . Toutes les contraintes sont actives. La surface S_3 ainsi obtenue est représentée sur la Figure 2.12.

Les voisins de T_3 sont $N(T_3, e_1)$ qui correspond à T_2 , $N(T_3, e_2)$ qui correspond à T_1 et $N(T_3, e_5)$. Aucun arbre n'est ajouté à *listeAFaire*. En effet les deux premiers voisins, T_1 et T_2 , ont déjà été considérés et $N(T_3, e_5)$ n'est pas un arbre.

Nous ajoutons ensuite (T_3, S_3) à *listeExplo*.

Après cette étape nous avons :

- *listeAFaire* est vide,
- *listeExplo* = $\{(T_0, S_0), (T_1, S_1), (T_2, S_2), (T_3, S_3)\}$.

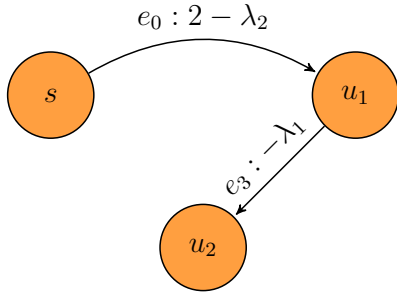


FIGURE 2.11 – T_3 .

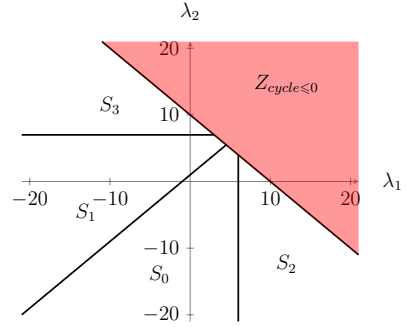


FIGURE 2.12 – S_0, S_1, S_2 , et S_3 .

Quand nous arrivons à ce stade, l'algorithme termine car *listeAFaire* est vide. Il renvoie *listeExplo* qui contient les couples (T_0, S_0) , (T_1, S_1) , (T_2, S_2) et (T_3, S_3) . Nous pouvons noter que nous avons bien $S_0 \cup S_1 \cup S_2 \cup S_3 = Z_{cycles \leq 0}$.

Nous avons bien que les zones S_0, S_1, S_2 et S_3 sont disjointes à leur frontière près et leur union est exactement $Z_{cycles \geq 0}$, ce qui signifie que toute valeur de paramètre est soit associée à un arbre, soit correspond à un cas où il existe au moins un cycle négatif dans le graphe. Les arbres sont bien des arbres d'énergies minimales sur les zones auxquelles ils sont associés. Cela reste vrai dans le cas général car nous construisons les zones avec les contraintes actives associées aux voisins ce qui nous permet de garantir ces propriétés.

2.4 Présentation formelle de notre algorithme

Dans cette section nous formalisons l'algorithme qui a été présenté dans l'exemple de la section précédente. Nous nous attachons ensuite à prouver la validité de l'algorithme.

2.4.1 L'algorithme

L'algorithme 1 est un algorithme qui, étant donné un graphe paramétré G et un sommet s , renvoie une liste de paires (T, S) telles que chaque T est un arbre d'énergies minimales sur S et que l'union de toutes les zones S au sein de la liste est égale à la zone telle qu'il n'existe pas de cycle négatif dans G .

Algorithme 1 Arbres d'énergies minimales de $G = (V, E, f, (\lambda_i)_1^n, (\Lambda_i)_1^n)$.

```

1: Si il existe une valeur de  $\vec{\lambda}$  telle qu'il n'y a pas de cycle négatif dans  $G$  Alors
2:   Soit  $\vec{\lambda}_{init}$  un tel  $\vec{\lambda}$ 
3:   Sinon renvoyer  $\emptyset$ 
4: Fin Si
5: Soit  $listeExplo = \emptyset$ .
6: Soit  $T_0$  un arbre d'énergies minimales pour  $\vec{\lambda}_{init}$ 
7: Soit  $listeAFaire = \{T_0\}$ 
8: Tant que  $listeAFaire \neq \emptyset$  Faire
9:   Choisir un arbre  $T$  dans  $listeAFaire$  (et le supprimer de  $listeAFaire$ )
10:  Soit  $Neighbours$  l'ensemble de tous les arcs  $e$  possibles tels que  $\exists i \in \llbracket 1, n \rrbracket, \Delta_{\Lambda_i}(T, e) \neq 0$ .

11:  Soit  $S$  la zone définie par les contraintes associées aux arcs de  $Neighbours$ 
12:  Soit  $Active$  le sous-ensemble de  $Neighbours$  ne contenant que les arcs associés à des
    contraintes actives
13:  Pour chaque arc  $e$  dans  $Active$  tel que  $N(T, e)$  est un arbre Faire
14:    Soit  $T_N = N(T, e)$ 
15:    Si  $T_N \notin listeAFaire$  et  $T_N \notin listeExplo$  Alors
16:      Ajouter  $T_N$  à  $listeAFaire$ 
17:    Fin Si
18:  Fin Pour
19:  Ajouter  $(T, S)$  à  $listeExplo$ 
20: Fin Tant que
21: Renvoyer  $listeExplo$ 

```

Remarquons que le calcul de T_0 à la ligne 6 de l'algorithme se fait avec un algorithme classique de calcul des plus courts chemins.

Pour la suite et tout comme dans l'exemple de la section précédente nous ferons référence à la zone où il n'y a pas de cycle négatif comme étant $Z_{cycles \geq 0}$. Nous appellerons Z_{explo} l'union des zones dans $listeExplo$.

L'algorithme 1 commence par calculer une valeur des paramètres pour laquelle il n'existe pas de cycle négatif à la première ligne. S'il n'existe pas de telles valeurs alors l'algorithme termine et renvoie la liste vide avec l'instruction de la ligne 3, sinon nous prenons l'une de ces valeurs pour être $\vec{\lambda}_{init}$ à la ligne 2. Ensuite T_0 est calculé à la ligne 6

comme un arbre d'énergies minimales pour $\vec{\lambda}_{init}$, cet arbre est l'arbre à partir duquel l'espace des valeurs des paramètres commencera à être exploré. Nous choisissons ainsi cet arbre initial afin de nous assurer que l'utilisation du concept d'énergie minimale est pertinente, c'est-à-dire qu'il n'existe pas de cycle négatif pour la valeur des paramètres retenue pour calculer T_0 . Nous ajoutons ensuite T_0 à *listeAFaire* à la ligne 7, cette liste représente les arbres que l'algorithme doit encore considérer.

Pour chaque arbre T considéré, l'algorithme commence par énumérer tous les arcs e qui peuvent générer un voisin avec l'instruction de la ligne 10. A partir des contraintes associées nous caractérisons S à la ligne 11, une zone où T est un arbre d'énergies minimales. Ensuite l'algorithme effectue deux actions : (1) à la ligne 16 il ajoute à *listeAFaire* tous les arbres voisins associés à des contraintes actives et qui n'ont pas encore été considérés par l'algorithme (c'est-à-dire ceux qui ne sont pas déjà dans *listeAFaire* ou dans *listeExplo*) et (2) il ajoute les nouveaux résultats dans la liste renvoyée en ajoutant T et S à *listeExplo* à la ligne 19. Nous pouvons noter que, par construction, les zones dans *listeExplo* sont disjointes à leur éventuelle frontière en commun près.

Quand *listeAFaire* est vide l'algorithme renvoie *listeExplo* avec l'instruction de la ligne 21.

Nous noterons que dans le cas où il n'y aurait qu'un seul paramètre notre algorithme est équivalent à l'algorithme présenté dans [KO81] puisque les contraintes ont une meilleure forme, ce qui rend le calcul des contraintes actives équivalent à une recherche de maximum. Ainsi il est possible d'effectuer une exploration plus efficace en raison de la présence d'une unique dimension.

2.4.2 Preuve de l'algorithme

Dans cette partie nous allons prouver que notre algorithme termine et qu'il renvoie des résultats exacts. Pour cela nous commençons par prouver grâce à un invariant de boucle que les couples (T, S) renvoyés par l'algorithme sont tels que T soit un arbre d'énergies minimales sur S , que S ne contient aucune valeur de paramètre telle qu'il y ait un cycle négatif dans le graphe et que les différentes zones renvoyées sont disjointes entre elles à leur frontière près.

Le premier lemme de cette partie prouve l'invariant de boucle qui sera un outil pour la démonstration de la validité des résultats de l'algorithme.

Lemme 2.1 (Invariant de boucle). *Soit $Z_{nonExplo} = Z_{cycles \geq 0} \setminus Z_{explo}$, à chaque itération*

de la boucle tant que de l'algorithme 1 nous avons :

(1) : $Z_{explo} \cup Z_{nonExplo} = Z_{cycles \geq 0}$ (en particulier il n'y a pas de $\vec{\lambda} \in Z_{explo}$ tel qu'il y ait un cycle négatif dans G pour $\vec{\lambda}$).

(2) : Pour tout $(T, S) \in listeExplo$, T est un arbre d'énergies minimales pour tout $\vec{\lambda} \in S$.

Démonstration. Initialement nous avons $Z_{explo} = \emptyset$ donc $Z_{nonExplo} = Z_{cycles \geq 0}$ et (1) est vraie. Par ailleurs $listeExplo$ est vide donc (2) est vraie.

Supposons que l'invariant de boucle est vrai pour les k premières itérations :

Nous avons $listeExplo = (T_i, S_i)_0^k$ et $Z_{explo} = \bigcup_{i=0}^k S_i$, comme l'invariant de boucle est vrai pour les k premières itérations nous avons $Z_{explo} \subseteq Z_{cycles \geq 0}$ et pour tout $(T, S) \in listeExplo$, T est un arbre d'énergies minimales pour tout $\vec{\lambda} \in S$.

Si l'algorithme termine avant la $(k + 1)$ -ème itération alors l'invariant de boucle est vrai, si l'algorithme ne termine pas, à la fin de la $(k + 1)$ -ème itération nous avons un arbre T_{k+1} et sa zone associée S_{k+1} ajoutés à $listeExplo$.

(1) est vraie après la $(k + 1)$ -ème itération si et seulement si $S_{k+1} \subseteq Z_{cycles \geq 0}$ (c'est à dire qu'il n'y a pas $\vec{\lambda} \in S_{k+1}$ tel qu'il existe un cycle négatif dans G pour $\vec{\lambda}$) et (2) est vraie si et seulement si T_{k+1} est un arbre d'énergies minimales sur S_{k+1} car (1) et (2) sont vraies pour les premières k itérations.

Si (2) est fausse cela signifie qu'il existe $\vec{\lambda}^* \in S_{k+1}$ tel qu'il y ait un arbre T' et un sommet v' tel que $d(T_{k+1}, v', \vec{\lambda}^*) > d(T', v', \vec{\lambda}^*)$ et pour tout $v \in V$, $d(T_{k+1}, v, \vec{\lambda}^*) \geq d(T', v, \vec{\lambda}^*)$.

Soit v^* le premier sommet sur le chemin allant de s à v' dans T' tel que $d(T_{k+1}, v^*, \vec{\lambda}^*) > d(T', v^*, \vec{\lambda}^*)$. Soient $sv_1v_2 \dots v_{i_0}$ les sommets du chemin allant de s à v^* dans T' et soit $e = (v_{i_0}, v^*)$. Nous avons $e \notin T_{k+1}$ car les énergies de s à v_{i_0} sont les mêmes dans T' et dans T_{k+1} . Étant donné que l'invariant de boucle est vrai pour les k premières itérations il est également vrai pour toutes les frontières de S_{k+1} avec les $S_{i \leq k}$ car par la proposition 2.2 les énergies sont les mêmes sur les frontières avec les voisins. L'arbre T_{k+1} est nécessairement le voisin d'au moins un $T_{i \leq k}$ et comme les énergies sont les mêmes sur les frontières, T_{k+1} est un arbre d'énergies minimales sur cette frontière, cela signifie que pour tout $v \in V$, $d(T_{k+1}, v, \vec{\lambda}_f) \leq d(T', v, \vec{\lambda}_f)$ avec λ_f les valeurs des paramètres appartenant à la frontière.

Soit $T^* = N(T_{k+1}, e)$, nous remarquerons que T^* est toujours strictement meilleur que T_{k+1} car $d(T_{k+1}, v^*, \vec{\lambda}^*) > d(T^*, v^*, \vec{\lambda}^*)$. La contrainte associée à T^* est $\Delta_f(T_{k+1}, e) - \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T_{k+1}, e) \geq 0$ et d'après la proposition 2.2 les énergies sont les mêmes pour $\vec{\lambda}$

sur la frontière définie par $\Delta_f(T_{k+1}, e) - \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T_{k+1}, e) = 0$. Soit cette contrainte est active et cela signifie que $\vec{\lambda}^* \notin S_{k+1}$, soit cette contrainte n'est pas active, ce qui signifie qu'il existe une contrainte plus restrictive et nous avons également $\vec{\lambda}^* \notin S_{k+1}$. Comme (2) est vraie dans tous les cas nous avons donc que (2) est vraie après la $(k+1)$ -ème itération.

Si (1) est fausse cela signifie qu'il existe une valeur $\vec{\lambda}^* \in S_{k+1}$ telle qu'il y ait un cycle négatif dans G . Comme l'invariant de boucle est vrai avant la dernière itération, l'énergie de ce cycle sur la frontière de S_{k+1} avec les $S_{i \leq k}$ est positive comme illustré dans un cas avec deux paramètres sur la figure 2.13.

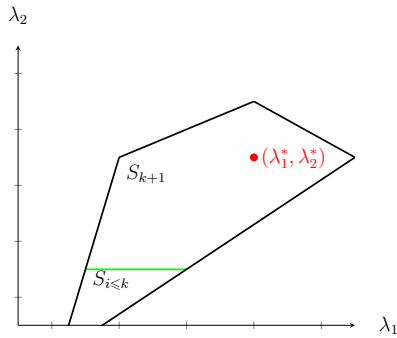


FIGURE 2.13 – L'invariant de boucle est vrai sur la frontière.

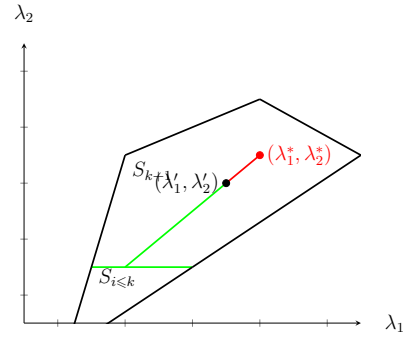
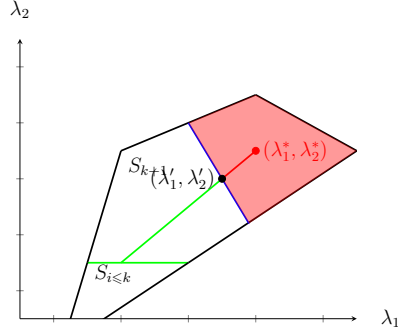


FIGURE 2.14 – (λ'_1, λ'_2) tel que le cycle soit d'énergie nulle.

Puisque S_{k+1} est convexe, cela signifie qu'il existe $\vec{\lambda}' \in S_{k+1}$ tel qu'il n'y ait pas de cycle négatif et un cycle d'énergie nulle comprenant au moins un paramètre dans l'énergie d'un de ses arcs comme illustré dans un cas avec deux paramètres sur la figure 2.14.

Soit $C' = e_1 e_2 \dots e_j$ l'un de ces cycles et soit j' tel que $e_{j'} \notin T_{k+1}$. Soit $e_{j'} = (v_{j'-1}, v_{j'})$. Comme nous avons que T_{k+1} est un arbre d'énergies minimales, nous avons que $d_f(T_{k+1}, v_{j'}) + \sum_{i=1}^n \lambda'_i d_{\Lambda_i}(T_{k+1}, v_{j'}) = d_f(T_{k+1}, v_{j'-1}) + f(e_{j'}) + \sum_{i=1}^n \lambda'_i (d_{\Lambda_i}(T_{k+1}, v_{j'-1}) + f_{\Lambda_i}(e_{j'}))$. Par conséquent $\Delta_f(T_{k+1}, e_{j'}) + \sum_{i=1}^n \lambda'_i \Delta_{\Lambda_i}(T_{k+1}, e_{j'}) = 0$, ce qui signifie que $\vec{\lambda}'$ est sur la contrainte associée à $e_{j'}$. Dans le cas où cette contrainte est active alors nous avons $\vec{\lambda}'$ qui est sur la frontière de S_{k+1} , ce qui contredit que $\vec{\lambda}^* \in S_{k+1}$ comme illustré dans un cas avec deux paramètres sur la figure 2.15. Dans le cas où cette contrainte n'est pas active, cela signifie qu'il existe des contraintes plus restrictives et donc $\vec{\lambda}' \notin S_{k+1}$ ce qui contredit également que $\vec{\lambda}^* \in S_{k+1}$. Par conséquent (1) est vraie dans tous les cas et nous avons donc que (1) est vraie après la $(k+1)$ -ème itération.

Puisque (1) et (2) sont vraies après la $(k+1)$ -ème itération nous avons que l'invariant est vrai après la $(k+1)$ -ème itération. Par conséquent nous avons démontré l'invariant de


 FIGURE 2.15 – $(\lambda_1^*, \lambda_2^*)$ n'est pas dans S_{k+1} .

boucle. □

En nous appuyant sur les lemmes précédents, nous prouvons le théorème principal de ce chapitre, qui dit que l'algorithme que nous avons présenté termine et renvoie des résultats exacts.

Théorème 2.1. *L'algorithme termine et renvoie listeExplo telle que pour tout $(T, S) \in \text{listeExplo}$, T soit un arbre d'énergies minimales pour tout $\vec{\lambda} \in S$, et $\bigcup_{(T,S) \in \text{listeExplo}} S = Z_{\text{cycles} \geq 0}$ et pour tout (T, S) dans listeExplo et $\vec{\lambda} \in S$ nous avons qu'il existe (T', S') avec $S \neq S'$ tel que $\vec{\lambda} \in S'$ si et seulement si il existe $C : \Delta_f(T, e_c) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e_c) \geq 0$ une contrainte active définissant S avec $N(T, e_c) = T'$, telle que $\vec{\lambda}$ vérifie $\Delta_f(T, e_c) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e_c) = 0$.*

Le dernière condition du théorème signifie que si λ appartient à deux zones différentes de *listeExplo* alors λ appartient à la frontière de ces zones, c'est-à-dire que les zones de *listeExplo* sont disjointes entre elles à l'exception de leur éventuelle frontière commune.

Démonstration. Si l'algorithme termine et renvoie \emptyset via la ligne 3 alors le théorème est vrai.

Dans le cas où l'algorithme ne termine pas cela signifie qu'il y a une boucle infinie.

Puisque nous considérons un nouvel arbre à chaque tour de boucle et qu'il y a un nombre fini d'arbres (possibles), il est impossible d'avoir une boucle infinie et donc l'algorithme termine bien. Nous avons également besoin que $Z_{\text{explo}} = Z_{\text{cycles} \geq 0}$ soit vrai à la fin de l'algorithme, c'est-à-dire lorsque *listeAFaire* est vide. Dans les cas où $Z_{\text{explo}} = Z_{\text{cycles} \geq 0}$ serait faux cela signifie que l'algorithme a manqué une ou plusieurs zones et cela n'est possible uniquement s'il existe au moins une zone S_p telle que pour tout $(T, S) \in \text{ListeExplo}$,

S_p n'ait pas de frontière commune avec S . Ce qui est impossible car $Z_{cycles \geq 0}$ est convexe par définition. En effet en prenant C_y l'ensemble des cycles simples de G nous pouvons définir $Z_{cycles \geq 0}$ comme suit : $Z_{cycles \geq 0} = \{ \vec{\lambda} \text{ tel que pour tout } c \in C_y, \phi(\rho, \vec{\lambda}) \geq 0 \}$.

Par conséquent l'algorithme termine et nous avons $\bigcup_{(T,S) \in \text{listeExplo}} S = Z_{cycles \geq 0}$.

Pour tout $(T, S) \in \text{listeExplo}$ nous avons également que T est arbre d'énergies minimales pour tout $\vec{\lambda} \in S$ et $\bigcup_{(T,S) \in \text{listeExplo}} S = Z_{cycles \geq 0}$ d'après l'invariant de boucle.

Enfin, considérons (T, S) et (T', S') appartenant *listeExplo* tels que $S \neq S'$ et $\vec{\lambda}$ tel que $\vec{\lambda} \in S$ et $\vec{\lambda} \in S'$. Nous avons donc que les zones S et S' sont voisines, soient e et e' les arcs tels que $T' = N(T, e)$ et que $T = N(T', e')$. Nous pouvons noter que les arcs e et e' ont par définition le même sommet d'arrivée, soient les sommets u, u' et v tels que $e = (u, v)$ et $e' = (u', v)$.

Soient C la contrainte associée à e et C' celle associée à e' . Nous avons alors $C(\vec{\lambda}) = \Delta_f(T, e) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e)$ et $C'(\vec{\lambda}) = \Delta_f(T', e') + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T', e')$. Avec :

- $\Delta_f(T, e) = d_f(T, u) + f(e) - d_f(T, v)$,
- pour tout $i \in \llbracket 1, n \rrbracket$, $\Delta_{\Lambda_i}(T, e) = d_{\Lambda_i}(T, u) + \Lambda_i(e) - d_{\Lambda_i}(T, v)$,
- $\Delta_f(T', e') = d_f(T', u') + f(e') - d_f(T', v)$,
- pour tout $i \in \llbracket 1, n \rrbracket$, $\Delta_{\Lambda_i}(T', e') = d_{\Lambda_i}(T', u') + \Lambda_i(e') - d_{\Lambda_i}(T', v)$.

Comme $T = N(T', e')$ alors nous avons que $d_f(T', u') + f(e') = d_f(T, v)$ et pour tout $i \in \llbracket 1, n \rrbracket$, nous avons que $d_{\Lambda_i}(T', u') + \Lambda_i(e') = d_{\Lambda_i}(T, v)$.

De manière symétrique, comme $T' = N(T, e)$ alors nous avons que $d_f(T, u) + f(e) = d_f(T', v)$ et pour tout $i \in \llbracket 1, n \rrbracket$, nous avons que $d_{\Lambda_i}(T, u) + \Lambda_i(e) = d_{\Lambda_i}(T', v)$.

Par conséquent nous avons que $C(\vec{\lambda}) = -C'(\vec{\lambda})$.

Par ailleurs puisque nous avons $\vec{\lambda} \in S$, nous avons également $C(\vec{\lambda}) \geq 0$ et comme nous avons $\vec{\lambda} \in S'$ nous avons aussi $C'(\vec{\lambda}) \geq 0$. Or comme $C(\vec{\lambda}) = -C'(\vec{\lambda})$ nous avons que $C(\vec{\lambda}) = 0$, c'est-à-dire que $\vec{\lambda}$ vérifie bien $\Delta_f(T, e_c) + \sum_{i=1}^n \lambda_i \Delta_{\Lambda_i}(T, e_c) = 0$.

Nous avons donc bien que si $\vec{\lambda}$ appartient à deux zones différentes de *listeExplo* alors nous avons une contrainte active C telle que $C(\vec{\lambda}) = 0$.

□

2.4.3 Complexité

Nous concluons la présentation de l'algorithme 1 en développant sa complexité temporelle dans le pire des cas. Nous prouvons qu'elle est exponentielle dans le nombre de

sommets et polynomiale dans le nombre d'arcs ainsi que dans le nombre de paramètres du graphe donné en entrée de l'algorithme. Cette complexité est due à l'opération de calcul des contraintes actives effectuée à chaque itération, en effet pour savoir si une contrainte est active nous avons besoin de tester l'égalité entre la zone définie par toutes les contraintes et celle définie par toutes les contraintes sauf celle considérée. Si les zones sont égales alors la contrainte n'est pas active si non elle l'est, ce test d'égalité doit être effectué pour chaque contrainte générée c'est-à-dire pour chaque voisin possible de l'arbre considéré par l'algorithme. De plus nous devons répéter cela à chaque itération ce qui explique la complexité.

Théorème 2.2. *La complexité en pire cas de l'algorithme 1 est bornée par $O(|V|^{|V|-1}(|E|-|V|)^3 n^3 \log(M))$.*

Démonstration. La complexité C_p de l'Algorithme 1 peut être notée : $C_p = \lambda_0 + T_{init} + Ajout(0) + \sum_{i=0}^{n_{it}-1} (C_i + S_i + \sum_{e \in E_i} Ct_i(e) + A_i + k_i Test + Arbre_i(Rech_i(Ltodo_i) + Rech_i(Lexp_i)) + |ArbreAF_i|Ajout(Ltodo_i) + |ArbreExp_i|Ajout(Lexp_i))$, où :

- λ_0 est le nombre d'opérations nécessaires au calcul de λ_{init} ,
- T_{init} est le nombre d'opérations nécessaires pour calculer T_0 ,
- $Ajout(l)$ est le nombre d'opérations nécessaires pour ajouter un élément dans une liste de longueur l ,
- n_{it} est le nombre d'itérations de la boucle *tant que* dans l'algorithme 1,
- C_i est le nombre d'opérations nécessaires pour choisir un arbre dans *listeAFaire* et le supprimer après cela dans la i -ème itération,
- S_i est le nombre d'opérations nécessaires pour chercher tous les arcs e qui génèrent des voisins qui ont au moins un l tel que $\Delta_{\Lambda_l}(T_i, e) > 0$ dans la i -ème itération de boucle,
- E_i est l'ensemble de tels arcs,
- $Ct_i(e)$ est le nombre d'opérations nécessaires pour calculer les contraintes associées à e dans la i -ème itération,
- A_i est le nombre d'opérations nécessaires pour déterminer les contraintes actives dans la i -ème itération,
- k_i est le nombre de contraintes actives qui définissent la zone associée à T_i ,
- $Test$ est le nombre d'opérations nécessaires pour tester si un voisin est un arbre,
- $Arbre_i$ est le nombre de contraintes actives associées à des arbres, considérées dans la i -ème itération,

- $Rech(l)$ est le nombre d'opérations nécessaires pour chercher si un arbre apparaît dans une liste de longueur l ,
- $Ltodo_i$ est la longueur de *listeAFaire* dans la i -ème itération, avant d'y ajouter un arbre et après avoir supprimé T_i de celle-ci,
- $Lexp_i$ est la longueur de *listeExplo* dans la i -ème itération, avant d'y ajouter un arbre,
- $ArbreAF_i$ est l'ensemble d'arbres associés à des contraintes actives qui ne sont pas dans *listeAFaire* dans la i -ème itération,
- $ArbreExp_i$ est l'ensemble d'arbres associés à des contraintes actives qui ne sont pas dans *listeExplo* dans la i -ème itération.

Dans la perspective où nous nous intéressons à la complexité dans le pire des cas nous pouvons donner une borne supérieure pour beaucoup des opérations développées auparavant, concernant le nombre de sommets $|V|$ dans G , le nombre d'arcs $|E|$ dans G , et le nombre de paramètres n .

Tout d'abord nous rassemblons les différentes opérations intervenant dans la complexité en sept termes qui sont :

- $Init = \lambda_0 + T_{init} + Ajout(0)$ les opérations effectuées avant l'entrée dans la boucle "tant que".
- $Voisins = \sum_{i=0}^{n_{it}-1} (C_i + S_i)$ toutes les opérations de sélection d'un arbre et de recherche de tous ses voisins effectuées dans la boucle "tant que".
- $Ct = \sum_{i=0}^{n_{it}-1} \sum_{e \in E_i} Ct_i(e)$ toutes les opérations de calcul des contraintes associées aux voisins effectuées dans la boucle "tant que".
- $Ct_{act} = \sum_{i=0}^{n_{it}-1} A_i$ toutes les opérations de calcul des contraintes actives effectuées dans la boucle "tant que".
- $TestArbre = \sum_{i=0}^{n_{it}-1} k_i Test$ toutes les opérations pour tester si un voisin associé à une contrainte active est un arbre ou non effectuées dans la boucle "tant que".
- $Recherche = \sum_{i=0}^{n_{it}-1} Arbre_i(Rech_i(Ltodo_i) + Rech_i(Lexp_i))$ toutes les opérations de recherche d'arbres dans les listes *listeAFaire* et *listeExplo* effectuées dans la boucle "tant que".
- $AjoutListe = \sum_{i=0}^{n_{it}-1} (|ArbreAF_i|Ajout(Ltodo_i) + |ArbreExp_i|Ajout(Lexp_i))$ toutes les opérations d'ajout d'arbres dans les listes *listeAFaire* et *listeExplo* effectuées dans la boucle "tant que".

Ce qui aboutit à l'expression de la complexité suivante : $Cp = Init + Voisins + Ct +$

$Ct_{act} + TestArbre + Recherche + AjoutListe$.

Commençons par remarquer la somme $\sum_{i=0}^{n_{it}-1}$ qui intervient dans tous les termes à l'exception de *Init* et qui correspond au nombre d'itérations de la boucle "tant que" de l'algorithme. Dans le pire des cas notre algorithme doit considérer chaque arbre enraciné en s possible et donc $n_{it} = O(|V|^{|V|-1})$ d'après la formule de Cayley [Cay89].

Terme principal : Ct_{act} . Considérons à présent $Ct_{act} = \sum_{i=0}^{n_{it}-1} A_i$, le terme responsable de la complexité de l'algorithme.

Soit Z_1 une zone définie par k_1 contraintes et Z_2 une zone définie par k_2 contraintes. Nous prenons la contrainte complémentaire de l'une des contraintes de Z_2 et nous testons si la zone définie par les k_1 contraintes de Z_1 et cette contrainte complémentaire est vide en utilisant la méthode de l'ellipsoïde [GL81]. Dans le cas où pour chaque contrainte de Z_2 , la zone définie par sa contrainte complémentaire et les k_1 contraintes de Z_1 est vide alors $Z_1 \subseteq Z_2$. Nous procédons de manière analogue pour tester si $Z_2 \subseteq Z_1$. Comme la méthode de l'ellipsoïde termine en moins de $6n^2L$ opérations, avec L la taille de l'espace utilisé pour définir le problème, nous avons $Eq(k_1, k_2) \leq (k_1 + k_2)6n^2L$, avec $Eq(k_1, k_2)$ le nombre d'opérations nécessaires pour déterminer si une zone définie par k_1 contraintes est égale à une zone définie par k_2 contraintes.

Comme nous avons $nb_{ct} = \max(k_1, k_2) + 1$ contraintes pour les zones manipulées par notre algorithme nous avons $L \leq (nb_{ct}n + n) \log(M)$ avec M la plus grande constante apparaissant dans les contraintes. Puisque les constantes des contraintes sont les deltas, nous avons : $M = \max((\sum_{e \in E} \Lambda_i(e))_{i=1..n}, \sum_{e \in E} |f(e)|)$. Donc pour toutes les zones considérées par l'algorithme nous avons : $Eq(k_1, k_2) \leq 6(|E| - (|V| - 1))n^3(|E| - (|V| - 1) + 2) \log(M)$ et $Eq(k_1, k_2) = O((|E| - |V|)^2 n^3 \log(M))$.

Comme nous devons tester chaque contrainte pour savoir si elle est active ou non, nous devons faire $|E_i|$ tests d'égalité. Dans le pire des cas toutes les contraintes sont actives. En effet si une contrainte n'est pas active nous pouvons la supprimer et ne plus la considérer dans les prochains tests d'égalité. Pour tout i , nous avons $A_i \leq |E_i| Eq(|E_i| + 1, |E_i|)$ et donc nous avons $A_i \leq 6(|E| - (|V| - 1))^2 n^3 (|E| - (|V| - 1) + 2) \log(M)$ et $A_i = O((|E| - |V|)^3 n^3 \log(M))$.

Par conséquent nous avons $Ct_{act} = O(|V|^{|V|-1} (|E| - |V|)^3 n^3 \log(M))$.

Init. Intéressons nous maintenant au terme *Init* représentant les opérations effectuées avant d'entrer dans la boucle "tant que".

Il y a au plus $2^{|V|}$ cycles simples dans G qui génèrent le même nombre de contraintes sur λ . Nous pouvons savoir si une zone définie par $2^{|V|}$ contraintes est vide en utilisant la méthode de l'ellipsoïde [GL81]. Nous n'avons à considérer que les cycles simples car si un cycle négatif c est une combinaison de deux ou plus cycles simples, alors la contrainte générée par c sera une combinaison des contraintes générées par les cycles simples qui composent c , par exemple si un cycle C est la combinaison de deux cycles simples C_1 et C_2 , la contrainte $C_C \geq 0$ associée à C sera la somme des contraintes $C_{C_1} \geq 0$ associée à C_1 et $C_{C_2} \geq 0$ associée à C_2 , c'est à dire $C_C = C_{C_1} + C_{C_2}$. Comme la méthode de l'ellipsoïde termine en moins de $6n^2L_c$ opérations, avec L_c la taille de l'espace utilisé pour définir le problème (voir le point suivant). Nous avons donc $\lambda_0 = 6n^2L_c$.

Comme dans le pire des cas nous avons $2^{|V|}$ contraintes pour définir la zone, nous avons $L_c \leq (2^{|V|}n+n) \log(M_c)$ avec M_c la plus grande constante apparaissant dans les contraintes. Puisque les contraintes sont les énergies totales des cycles, nous avons :

$$M_c = \max\left(\sum_{e \in E} \Lambda_i(e)_{i=1..n}, \sum_{e \in E} |f(e)|\right).$$

Par conséquent nous avons $\lambda_0 = O(6n^2(2^{|V|}n + n) \log(M_c))$.

En calculant T_0 avec l'algorithme de Bellman-Ford nous avons $T_{init} = O(|V||E|)$ [For56].

Enfin $Ajout(0)$ consiste à ajouter un élément à une liste donc nous avons $Ajout(0) = O(1)$.

Nous avons donc $Init = O(6n^2(2^{|V|}n + n) \log(M_c)) + O(|V||E|) + O(1)$ c'est-à-dire $Init = O(6n^2(2^{|V|}n + n) \log(M_c))$ et nous avons bien $Init \ll Ct_{act}$.

Voisins. Examinons maintenant le terme *Voisins* correspondant aux opérations de sélection d'un arbre et de recherche de tous ses voisins dans la boucle "tant que".

Comme nous pouvons toujours prendre le premier élément de *listeAFaire* nous pouvons considérer que pour tout i , $C_i = 1$.

Considérons un arbre T , chaque arc qui n'est pas dans T peut générer un voisin et nous avons donc $S_i = |E| - (|V| - 1)$ puisque T a $|V| - 1$ arcs soit $S_i = O(|E| - |V|)$.

Nous avons $Voisins = O(|V|^{|V|-1}(|E| - |V|))$ et par conséquent $Voisins \ll Ct_{act}$.

Ct. Nous considérons à présent le terme *Ct* représentant les calculs des contraintes associées aux voisins dans la boucle "tant que".

Comme chaque arbre n'appartenant pas à un arbre peut générer un voisin nous avons $E_i = |E| - (|V| - 1)$ dans le pire des cas tout comme $S_i = |E| - (|V| - 1)$ dans le pire des

cas.

Soit T un arbre et $e = (u, v)$ un arc qui n'est pas dans T . Pour calculer la contrainte associée à $N(T, e)$ nous avons besoin de calculer le chemin de s à u ainsi que le chemin de s à v dans T . Comme chaque arbre a $|V| - 1$ arcs, quel que soit i , quel que soit e qui n'est pas dans T_i nous avons $Ct_i(e) \leq 2(|V| - 1)$ et donc nous avons $Ct_i = O(|V|)$.

Par conséquent nous avons $Ct = O(|V|^{|V|-1}(|E| - |V|)|V|)$, nous avons donc bien $Ct \ll Ct_{act}$.

TestArbre. Nous nous intéressons à présent au terme *TestArbre* correspondant aux tests pour savoir si les voisins associés à une contrainte active sont ou non des arbres.

Dans le pire des cas, durant la i -ème itération tous les arcs e qui ne sont pas dans T_i sont associés à des contraintes actives et nous avons $k_i \leq |E| - (|V| - 1)$. Ce qui revient à $k_i = O(|E| - |V|)$.

Soit T un arbre et e un arc qui n'est pas dans T , nous avons seulement besoin de vérifier si v est sur le chemin de s à u dans T pour savoir si nous sommes dans le cas où $N(T, e)$ est un arbre ou non. Nous avons donc $Test \leq |V| - 1$ et $Test = O(|V|)$.

Par conséquent $TestArbre = O(|V|^{|V|-1}(|E| - |V|)|V|)$, et nous avons bien $TestArbre \ll Ct_{act}$.

Recherche. Considérons maintenant le terme *Recherche* des opérations de recherche d'arbres dans les listes *listeAFaire* et *listeExplo*.

Dans le pire des cas, durant la i -ème itération tous les voisins qui sont associés à des contraintes actives sont des arbres et nous avons $Arbre_i \leq k_i$ soit $Arbre_i \leq |E| - (|V| - 1)$. Ce qui revient à $Arbre_i = O(|E| - |V|)$.

Puisque nous vérifions si un arbre n'est pas déjà dans une liste avant de l'ajouter à cette liste, dans le pire des cas tous les arbres possibles sont dans cette liste et pour tout i nous avons $Ltodo_i \leq |V|^2$ et $Lexp_i \leq |V|^2$ donc nous avons $Ltodo_i = O(|V|^2)$ et $Lexp_i = O(|V|^2)$.

Si nous représentons les arbres manipulés par l'algorithme par des tableaux de taille $|V| - 1$, avec une case pour chaque sommet v de $V \setminus \{s\}$, donnant le prédécesseur de v , et si *listeAFaire* et *listeExplo* sont des arbres binaires de recherche auto-équilibrés nous avons $Rech(l) = O(\log((|V| - 1)|l|))$.

Par conséquent $Recherche = O(|V|^{|V|-1}(|E| - |V|)|V|^3 \log((|V| - 1)))$, et nous avons bien $Recherche \ll Ct_{act}$.

AjoutListe. Pour finir examinons le terme *AjoutListe* des opérations d'ajout d'arbres dans les listes *listeAFaire* et *listeExplo*.

En utilisant le même raisonnement que pour $Arbre_i$ nous obtenons que $|ArbreAF_i| \leq |E| - (|V| - 1)$ et $|ArbreExp_i| \leq |E| - (|V| - 1)$. C'est-à-dire $|ArbreAF_i| = O(|E| - |V|)$ et $|ArbreExp_i| = O(|E| - |V|)$.

Pour des raisons identiques au calcul de $Rech(l)$ nous avons $Ajout(l) = O(\log((|V| - 1)|l|))$. Par conséquent $AjoutListe = O(|V|^{|V|-1}(|E| - |V|)|V|^3 \log(|V| - 1))$, et nous avons bien $AjoutListe \ll Ct_{act}$.

Tous les termes de l'expression de la complexité de l'algorithme sont négligeables devant Ct_{act} , par conséquent nous avons bien $Cp = O(|V|^{|V|-1}(|E| - |V|)^3 n^3 \log(M))$ ce qui conclut cette preuve. \square

2.5 Cas avec des paramètres à valeurs entières

Nous pouvons nous demander s'il serait possible d'adapter les preuves de l'algorithme présentées ici au cas où les paramètres peuvent uniquement prendre des valeurs dans \mathbb{Z} afin de ne pas avoir à considérer des valeurs réelles de paramètres pour ensuite restreindre les résultats renvoyés par notre algorithme. En effet l'algorithme ne peut pas simplement être appliqué à des graphes paramétrés dont les paramètres prennent uniquement des valeurs entières comme nous le montrons dans l'exemple présenté sur la figure 2.16.

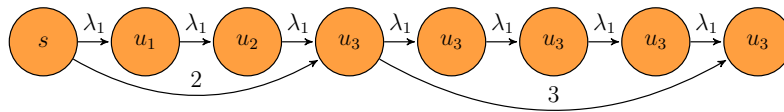
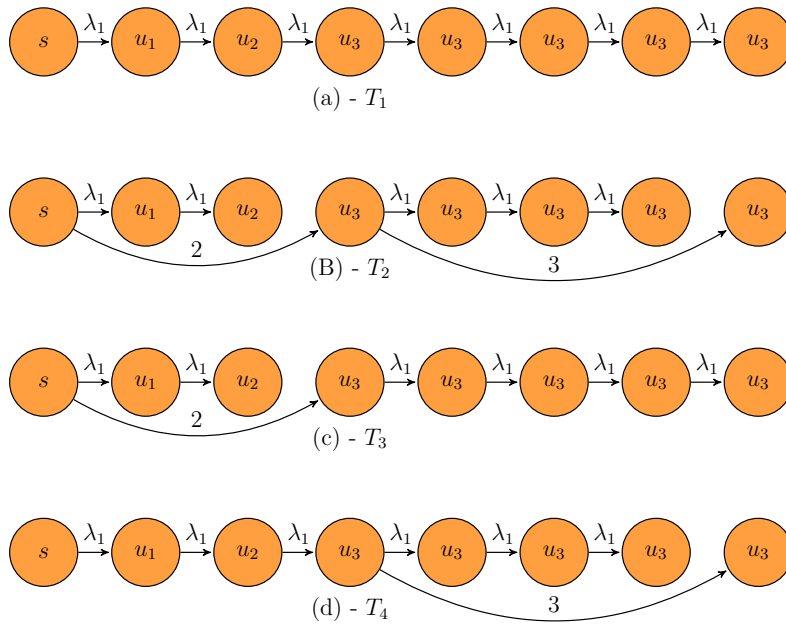


FIGURE 2.16 – Un graphe G avec un paramètre.

Ce graphe n'a qu'un seul paramètre et quatre sous graphes qui sont des arbres, nous les avons représentés dans la Figure 2.17.

Notons que T_1 est un arbre d'énergies minimales pour tout $\lambda \leq 0$, que T_2 est un arbre d'énergies minimales pour tout $\lambda \geq 1$ et que pour tout $\lambda \in \mathbb{Z}$, T_3 et T_4 ne sont pas des arbres d'énergies minimales. Comme T_1 et T_2 ne sont pas voisins notre algorithme n'est plus applicable quand les paramètres prennent des valeurs de nombres entiers.


 FIGURE 2.17 – Les quatre arbres sous graphes du graphe G .

2.6 Conclusion

Nous avons proposé un algorithme visant à trouver les chemins optimaux pour aller d'une unique source à tous les autres sommets dans un graphe paramétré dans lequel les énergies des arcs font appel à un nombre arbitraire de paramètres à valeurs réelles. Puisque ces chemins optimaux changent avec les valeurs des paramètres, le résultat de notre algorithme est un ensemble fini d'arbres, chacun de ces arbres étant associé par l'algorithme à une zone de l'espace des paramètres sur laquelle il est optimal. Ces zones couvrent l'espace des paramètres pour lequel il n'y a pas de cycle d'énergie négative au sein du graphe. Cet algorithme permet de généraliser des travaux préalablement réalisés par Karp et Orlin dans lesquels seul un unique paramètre était considéré [KO81].

De futurs travaux visant à prolonger le travail sur les plus courts chemins d'une source à tous les autres sommets d'un graphe paramétré pourraient consister en une implémentation de l'algorithme ainsi qu'une évaluation de son efficacité sur des études de cas réels. Il serait également possible de s'intéresser au cas où les paramètres prennent des valeurs entières. En effet dans ce cas la notion d'arbres voisins ne permettrait plus d'énumérer les arbres de manière exhaustive car certains arbres voisins qui sont optimaux pour certaines valeurs rationnelles ne sont optimaux pour aucune valeur entière ce qui implique des modifications importantes de l'algorithme si nous voulons qu'il puisse prendre en compte ce

cas. Nous pourrions également essayer de généraliser la notion d'arbres voisins aux cas où l'énergie a plusieurs dimensions afin de chercher à étendre notre algorithme aux graphes multi-dimensionnels dans lesquels la définition d'énergie minimale ne s'applique plus.

ÉTUDE DE LA DÉCIDABILITÉ DE DEUX PROBLÈMES LU TOTAUX EN FONCTION DES VALEURS PRISES PAR LES PARAMÈTRES

3.1 Introduction

Nous avons étudié dans le chapitre précédent le sujet des chemins avec une énergie la plus faible possible qui consiste à imposer une contrainte sur l'énergie relative aux énergies des autres chemins. Cependant ce genre de problématique est incapable de répondre à des questions concernant des contraintes plus absolues sur l'énergie. La notion d'énergie minimale disparaît également lorsque l'énergie est à plusieurs dimensions. En effet à moins d'imposer arbitrairement une convention, un chemin ayant une énergie de $(1, 2)$ et un autre ayant une énergie de $(2, 1)$ ne sont pas comparables au sens des chemins à énergie minimale.

Nous allons à présent nous intéresser à un type de contraintes plus absolu à imposer à l'énergie des chemins. Un tel modèle pourrait servir à représenter par exemple un moteur hybride. En effet, la présence de deux sources d'énergie différentes pour faire fonctionner un moteur thermique et un moteur électrique implique des contraintes différentes sur la gestion des deux sources d'énergie. Nous allons donc chercher à savoir s'il existe un chemin allant d'un sommet donné à un autre tel que son énergie reste comprise entre des bornes. De plus nous allons considérer une énergie multi-dimensionnelle, chaque dimension pouvant avoir ses propres bornes, et nous nous intéresserons à la décidabilité de différents problèmes où nous imposons à l'énergie d'être bornée.

Dans ce chapitre nous nous intéresserons à plusieurs variations du problème $EnTot_{LU}(G, (L_i, U_i)_1^d, \mathbb{A}, v, w)$. Les différents problèmes sont obtenus en faisant varier dif-

férents aspects. Nous allons en effet nous intéresser à la décidabilité du problème LU total en faisant varier d le nombre de dimension de l'énergie mais également en modifiant l'ensemble des valeurs possibles des paramètres, c'est-à-dire en faisant varier \mathbb{A} . Les problèmes LU totaux étant des problèmes d'accessibilité nous ne nous préoccupons pas des parités des arcs dans ce chapitre.

Nous proposons dans ce chapitre des preuves de décidabilité sur deux variations du problème LU total. Nous prouvons tout d'abord la décidabilité du problème LU total lorsque les paramètres sont à valeurs dans \mathbb{Z} . Nous prouvons que ce résultat reste valide et que le problème est PSPACE-complet quels que soient la dimension de l'énergie et le nombre de paramètres. Nous nous intéressons ensuite au problème LU total lorsque les paramètres peuvent prendre des valeurs rationnelles et nous prouvons que dès que la dimension du graphe est d'au moins 2 ($d \geq 2$), alors le problème LU total est indécidable dès qu'il y a au moins un paramètre qui intervient dans les énergies des arcs ($n \geq 1$).

Ce chapitre est organisé de la manière suivante : dans la Section 3.2 nous prouvons que la version du problème LU total est décidable et PSPACE-complet indépendamment de la dimension de l'énergie lorsque les paramètres sont à valeurs entières ; dans la Section 3.3, nous prouvons que le problème LU total est indécidable dans sa version où les paramètres peuvent prendre des valeurs rationnelles et que l'énergie est de dimension au moins deux ; enfin dans la Section 3.4 nous concluons.

3.2 Problème LU total avec paramètres à valeurs entières

Soit $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée. Soient $L \in \mathbb{Z}^d$ et $U \in \mathbb{Z}^d$. Nous considérons dans cette section le problème $EnTot_{LU}(G, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ où les paramètres prennent des valeurs entières. Nous prouvons que ce problème est décidable et PSPACE-complet quelle que soit la dimension d de G .

Pour cela nous commençons par identifier les arcs du graphe dont l'énergie est influencée par les paramètres.

Nous cherchons ensuite les valeurs de paramètres telles qu'au moins un arc soit empruntable parmi ceux dont l'énergie est influencée par les paramètres.

Enfin nous montrons que de telles valeurs de paramètres sont en nombre fini et nous pouvons donc les énumérer afin de résoudre $EnTot_{LU}(G, L, U, \mathbb{Z}, v, w)$.

Théorème 3.1. Soient $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée, $m \in \mathbb{N}$, $L \in \mathbb{Z}^d$ et $U \in \mathbb{Z}^d$. Le problème $EnTot_{LU}(G, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ est décidable et est PSPACE-complet.

Démonstration. Soient $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée, $d \in \mathbb{N}$, $L \in \mathbb{Z}^d$ et $U \in \mathbb{Z}^d$. Pour tout $i \in \llbracket 1, d \rrbracket$ nous définissons $E_{\lambda, i}$ l'ensemble des arcs $e \in E$ tels que $\exists y \in \mathbb{Z}, \exists \lambda \in \mathbb{Z}^n$ et $\exists \lambda' \in \mathbb{Z}^n$ tels que $\lambda \neq \lambda'$ et $\phi_e^\lambda(y)_i \neq \phi_e^{\lambda'}(y)_i$. Intuitivement $E_{\lambda, i}$ est l'ensemble des arcs dont la i -ème dimension de l'énergie est influencée par la valeur des paramètres.

Soit S_i l'ensemble des $\lambda \in \mathbb{Z}^n$ valeurs des paramètres pour lesquelles il existe un $e \in E_{\lambda, i}$ et $y \in [L_i, U_i]$ tels que $\phi_e^\lambda(y)_i \in [L_i, U_i]$. Intuitivement S_i est l'ensemble des valeurs de paramètres telles qu'au moins un arc de $E_{\lambda, i}$ puisse être emprunté dans G en respectant la contrainte sur la i -ème dimension de l'énergie.

Pour tout $i \in \llbracket 1, d \rrbracket$ et $e \in E_{\lambda, i}$ soit S_i^e l'ensemble des $\lambda \in \mathbb{Z}^n$ valeurs des paramètres pour lesquelles il existe $y \in [L_i, U_i]$ tel que $\phi_e^\lambda(y)_i \in [L_i, U_i]$. C'est-à-dire que S_i^e est l'ensemble des valeurs des paramètres telles qu'il soit possible d'emprunter l'arc e sans enfreindre les contraintes sur la i -ème dimension de l'énergie. Notons que pour tout $i \in \llbracket 1, d \rrbracket$ nous avons $S_i = \bigcup_{e \in E_{\lambda, i}} S_i^e$.

Pour tout $i \in \llbracket 1, d \rrbracket$, pour tout $e \in E_{\lambda, i}$, nous avons α_e une fonction affine de coefficient directeur non nul et donc il n'existe qu'un nombre fini de valeurs possibles λ telles que pour tout $y \in [L_i, U_i]$ nous ayons $\phi_e^\lambda(y)_i \in [L_i, U_i]$, puisque $\phi_e^\lambda(y)_i = y + \alpha_e(\lambda)$. Par conséquent S_i^e est un ensemble fini, quels que soient $i \in \llbracket 1, d \rrbracket$ et $e \in E_{\lambda, i}$. Comme E est fini, $E_{\lambda, i} \subset E$ est également fini et pour tout i , $S_i = \bigcup_{e \in E_{\lambda, i}} S_i^e$ est un ensemble fini.

Nous définissons $S = \bigcap_{i \in \llbracket 1, d \rrbracket} S_i^e$, l'ensemble des valeurs de paramètres telles qu'il y ait au moins un arc $e \in E_{\lambda, i}$ qui puisse être emprunté dans G en respectant les contraintes sur toutes les dimensions de l'énergie. Comme les S_i sont des ensembles finis quel que soit i nous avons que S est un ensemble fini.

Pour tout $\lambda \in S$, soit G_λ le graphe où les valeurs des paramètres ont été fixées à λ . Comme S est fini, nous avons un nombre fini de G_λ et nous pouvons résoudre $EnTot_{LU}(G_\lambda, L, U, \mathbb{Z}, v, w)$ pour chaque $\lambda \in S$. En effet le problème $EnTot_{LU}(G_\lambda, L, U, \mathbb{Z}, v, w)$ ne fait intervenir aucun paramètre, est décidable et est PSPACE-complet [Fah+11]. S'il existe λ_s tel que la réponse à $EnTot_{LU}(G_{\lambda_s}, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ soit oui alors la réponse à $EnTot_{LU}(G, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ est oui et un chemin de G_{λ_s} vérifiant les contraintes sur l'énergie est également un chemin de G vérifiant les contraintes sur l'énergie lorsque les paramètres prennent la valeur λ_s . Pour les mêmes raisons, si pour tout $\lambda \in S$, la réponse

à $EnTot_{LU}(G_\lambda, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ est non alors la réponse à $EnTot_{LU}(G, L, U, \mathbb{Z}, v, w)$ est non.

Dans tous les cas cela nous permet de conclure que $EnTot_{LU}(G, L, U, \mathbb{Z}, v, w)$ est bien décidable.

Comme la construction de chaque S_i nécessite l'énumération des $|E|$ arcs de G , nous pouvons construire S en temps polynomial en d la dimension de G et $|E|$ son nombre d'arcs. Pour tout $\lambda \in S$, puisque la création de G_λ consiste à fixer la valeur de l'énergie de chaque arc dans lesquels interviennent des paramètres, nous pouvons construire G_λ en temps polynomial en $|E|$ le nombre d'arcs de G . Enfin puisque pour tout $\lambda \in S$, le problème $EnTot_{LU}(G_\lambda, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ est PSPACE-complet [Fah+11] nous avons que le problème $EnTot_{LU}(G, (L_i, U_i)_1^d, \mathbb{Z}, v, w)$ est PSPACE-complet. \square

Nous pouvons remarquer que les résultats sont toujours valides si les paramètres sont à valeurs dans \mathbb{N} au lieu d'être à valeurs dans \mathbb{Z} . Par conséquent la variante du problème LU total lorsque les paramètres prennent des valeurs dans \mathbb{N} est également décidable indépendamment de la dimension de l'énergie et est également PSPACE-complet.

3.3 Problème LU total avec paramètres à valeurs rationnelles

Soit $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée tel que $d \geq 2$. Soient $L \in \mathbb{R}^d$ et $U \in \mathbb{R}^d$. Nous considérons dans cette section le problème $EnTot_{LU}(G, L, U, \mathbb{A}, v, w)$ avec $\mathbb{Q} \subseteq \mathbb{A}$ où les paramètres prennent des valeurs pouvant être rationnelles.

Nous prouvons avec le théorème suivant que $EnTot_{LU}(G, L, U, \mathbb{A}, v, w)$ avec $\mathbb{Q} \subseteq \mathbb{A}$ et $d \geq 2$ est indécidable dès lors qu'un paramètre intervient dans les énergies des arcs, c'est-à-dire si $n \geq 1$. Nous montrons cela en réduisant notre problème au problème de l'arrêt des machines à deux compteurs qui est connu pour être indécidable [Min67].

Pour cela nous commençons par définir une transformation qui permet de créer un graphe G à partir d'une machine à deux compteurs M . Cela nous permettra d'associer n'importe quel chemin de la machine à compteurs à un chemin du graphe mais également d'appliquer le problème $EnTot_{LU}(G, L, U, \mathbb{A}, v, w)$ à ce graphe particulier.

Nous montrons ensuite par récurrence que si M s'arrête alors il existe un chemin dans G tel que la réponse à $EnTot_{LU}(G, L, U, \mathbb{A}, v, w)$ soit oui. Puis nous prouvons que, récipro-

quement, s'il existe un chemin dans G tel que la réponse à $EnTot_{LU}(G, L, U, \mathbb{A}, v, w)$ soit oui, alors M s'arrête.

Nous pouvons alors conclure étant donné que le problème de l'arrêt est indécidable et que M s'arrête si et seulement si la réponse à $EnTot_{LU}(G, L, U, \mathbb{A}, v, w)$ est oui.

Théorème 3.2. *Soient \mathbb{A} tel que $\mathbb{Q} \subseteq \mathbb{A}$ et $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un graphe multi-dimensionnel orienté à énergie paramétrée tel que $d \geq 2$ et $n \geq 1$. Soient $(v, w) \in V^2$ et $(L, U) \in (\mathbb{R}^d)^2$. Le problème $EnTot_{LU}(G, (L, U), \mathbb{A}, v, w)$ est indécidable.*

Démonstration. Nous prouvons l'indécidabilité en réduisant le problème de l'arrêt pour les machines à deux compteurs à une sous-classe de notre problème : les problèmes de la forme $EnTot_{LU}(G, (\vec{0}, \vec{1}), \mathbb{A}, v, w)$.

Soit $M = (Q, q_0, h, T, \delta, C_1, C_2)$ une machine à deux compteurs. Nous construisons $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E})$ et $U \in \mathbb{R}^m$ tels que $\exists \vec{\lambda} \in \mathbb{A}^n$ et un chemin ρ de G ayant pour sommet initial v et sommet final w tel que ρ est borné par $(\vec{0}, \vec{1})$ si et seulement si M s'arrête. Avec $\vec{0}$ tel que pour tout i , $\vec{0}_i = 0$ et $\vec{1}$ tel que pour tout i , $\vec{1}_i = 1$.

Nous définissons E , l'ensemble des arcs de G ainsi que les fonctions d'énergie des arcs $\alpha_e : \mathbb{A}^n \rightarrow \mathbb{A}^d$ à partir de T l'ensemble des transitions de M et de δ .

Comme nous ne ferons intervenir qu'un seul paramètre dans G , nous prendrons $n = 1$ pour la suite de cette preuve et $\vec{\lambda} = \lambda$.

Soit $t = (q_M, q'_M) \in T$ une transition de M , soit q_G (resp. q'_G) le sommet de G correspondant à q_M (resp. q'_M). Nous avons $\delta(t) = (C_i, \sim)$ avec $i \in \{1, 2\}$ et $\sim \in \{-, +, =, >\}$:

- si \sim est $-$ alors nous avons $e_1 = (q_G, q'_G) \in E$ tel que $\alpha_{e_1}(\lambda)_i = -\lambda$ et $\alpha_{e_1}(\lambda)_j = 0$ pour $j \neq i$ comme représenté dans la figure 3.1 pour le cas $i = 1$. Nous disons alors que le chemin $\rho = e_1$ dans G est le chemin associé à t .

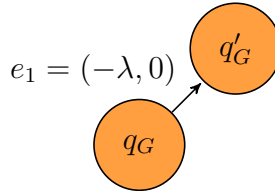
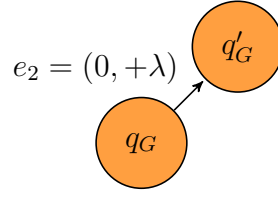
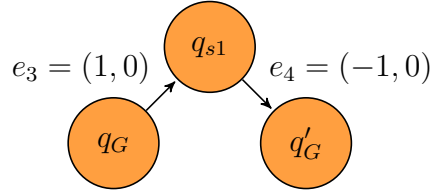


FIGURE 3.1 – Un gadget qui soustrait 1 du compteur C_1 .

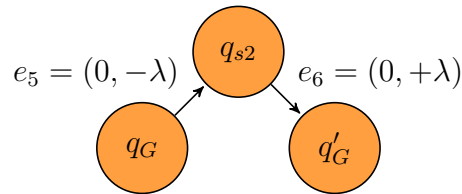
- si \sim est $+$ alors nous avons $e_2 = (q_G, q'_G) \in E$ tel que $\alpha_{e_2}(\lambda)_i = +\lambda$ et $\alpha_{e_2}(\lambda)_j = 0$ pour $j \neq i$ comme représenté dans la figure 3.2 pour le cas $i = 2$. Nous disons alors que le chemin $\rho = e_2$ dans G est le chemin associé à t .


 FIGURE 3.2 – Un gadget qui ajoute 1 au compteur C_2 .

- si \sim est = alors nous avons un sommet spécial $q_{s1} \in Q_s$ et deux arcs $e_3 = (q_G, q_{s1})$ et $e_4 = (q_{s1}, q'_G)$, $(e_3, e_4) \in E^2$ tels que $\alpha_{e_3}(\lambda)_i = i$, $\alpha_{e_3}(\lambda)_j = 0$ pour $j \neq i$, et $\alpha_{e_4}(\lambda)_i = -i$, et $\alpha_{e_4}(\lambda)_j = 0$ pour $j \neq i$. Nous avons représenté cela dans la figure 3.3 pour le cas $i = 1$. Nous disons alors que le chemin $\rho = e_3e_4$ dans G est le chemin associé à t .


 FIGURE 3.3 – Un gadget qui teste si le compteur C_1 est à zéro.

- si \sim est $>$ alors nous avons un sommet spécial $q_{s2} \in Q_s$ et deux arcs $e_5 = (q_G, q_{s2})$ et $e_6 = (q_{s2}, q'_G)$, $(e_5, e_6) \in E^2$ avec e_5 tel que $\alpha_{e_5}(\lambda)_i = -\lambda$, et $\alpha_{e_5}(\lambda)_j = 0$ pour $j \neq i$ exactement comme e_1 défini précédemment. Avec e_6 tel que $\alpha_{e_6}(\lambda)_i = \lambda$ et $\alpha_{e_6}(\lambda)_j = 0$ pour $j \neq i$ exactement comme e_2 défini précédemment. Nous avons représenté ce gadget dans la figure 3.4 pour le cas $i = 2$. Nous disons alors que le chemin $\rho = e_5e_6$ dans G est le chemin associé à t .


 FIGURE 3.4 – Un gadget qui teste si le compteur C_2 est supérieur à zéro.

Nous prenons Q_s l'ensemble contenant tous les sommets spéciaux (nommés q_{s1} et q_{s2} dans la définition de E) induits par la transformation des transitions telles que \sim soit = ou que \sim soit $>$.

Les autres éléments de G sont définis comme suit :

- $V = Q \cup Q_s$ avec Q_s l'ensemble des sommets spéciaux défini précédemment.
- $n \geq 1$ puisque seul un paramètre interviendra dans l'énergie mais nous pouvons en avoir un nombre quelconque.
- $d \geq 2$ en effet les arcs ne modifient que les deux premières dimensions de l'énergie.
- $\mu_e = 1$ pour tout $e \in E$.

Nous appelons $q_{0,G}$ le sommet de G correspondant à l'état initial q_0 de M et h_G le sommet de G correspondant à l'état d'arrêt h de M .

Soit $\rho_M = t_1 t_2 \dots t_k$ un chemin de M , nous définissons son *chemin associé* ρ_G un chemin de G . Pour tout $j \in \llbracket 1, k \rrbracket$ soit $\delta(t_j) = (C_j, \sim_j)$ avec $C_j \in \{C_1, C_2\}$ et $\sim_j \in \{-, +, =, >\}$. Nous avons $\rho_G = \rho_1 \rho_2 \dots \rho_k$ tel que pour tout $i \in \llbracket 1, k \rrbracket$, ρ_i est le chemin associé à t_i .

Nous pouvons remarquer que, par construction, nous avons $\rho_G = e_1 e_2 \dots e_l$ avec $l \geq k$.

Soit ρ_G un chemin de G dont le sommet final n'appartient pas à Q_s , nous définissons de manière analogue ρ_M le chemin de M associé à ρ_G . Nous pouvons faire cela car les seules ambiguïtés potentielles sont entre des arcs tels que e_1 et e_5 définis précédemment ou entre des arcs tels que e_2 et e_6 définis précédemment. Cependant les arcs e_1 et e_2 ne font pas intervenir de sommet de Q_s contrairement aux arcs e_5 et e_6 ce qui signifie qu'il n'y a bien aucune ambiguïté et que ρ_M est bien défini. En effet comme M est déterministe nous pouvons définir la projection de ρ_G sur Q puis ρ_M à l'aide de celle-ci.

Remarquons que, pour un chemin ρ_G de G dont le sommet final appartient à Q_s , nous ne pouvons pas définir de chemin associé dans M car les sommets de Q_s correspondent à des étapes intermédiaires dans la transcription des transitions de M lors de la construction de G .

Il nous reste à prouver que pour tout \mathbb{A} tel que $\mathbb{Q} \subseteq \mathbb{A}$ et $d \geq 2$, $\exists \lambda \in \mathbb{A}$ et ρ un chemin de G de $q_{0,G}$ à h_G tel que ρ soit borné par $(\vec{0}, \vec{1})$ si et seulement si M s'arrête.

Soit $r_M = \sigma_0 \sigma_1 \dots \sigma_\ell$ une exécution dans M telle que M s'arrête. Soit $\rho_M = t_1 t_2 \dots t_\ell$ le chemin correspondant à r_M . Nous appelons $\rho_{M,i}$ le préfixe de longueur i de ρ_M . Pour tout $i \in \llbracket 0, \ell \rrbracket$, soit $\sigma_i = (q_i, c_1(\rho_{M,i}), c_2(\rho_{M,i}))$. Soient $max_1 = \max_{i \in \llbracket 0, \ell \rrbracket} c_1(\rho_{M,i})$ et $max_2 = \max_{i \in \llbracket 0, \ell \rrbracket} c_2(\rho_{M,i})$.

Soit ρ_G le chemin de G associé à ρ_M , nous nommons $\rho_{G,i}$ le préfixe de ρ_G associé à $\rho_{M,i}$. Notons que $\rho_{G,i}$ n'est pas nécessairement le préfixe de longueur i de ρ_G .

Nous commençons par prouver par récurrence que si M s'arrête alors $\exists \lambda \in \mathbb{A}$ et ρ un chemin de G de v à w tel que ρ soit borné par $(\vec{0}, \vec{1})$ avec $v = q_{0,G}$ et $w = h_G$.

Soit $max_C = \max_{i \in \{1,2\}} (max_i)$.

Si $max_C = 0$, cela signifie que les valeurs des deux compteurs restent constantes égales à zéro pendant toute l'exécution c'est-à-dire que pour tout i , $\gamma_i = (q_i, 0, 0)$. Toutes les transitions sont donc des tests à zéro c'est-à-dire que pour tout i , nous avons $\delta(t_i) = (C_{t_i}, =)$ avec $C_{t_i} \in \{C_1, C_2\}$. Comme les cas $C_{t_i} = C_1$ et $C_{t_i} = C_2$ sont symétriques nous ne considérerons que le cas $C_{t_i} = C_1$.

Pour tout i , soit ρ_i le chemin associé à t_i , nous avons $\rho_i = e_{i,1}e_{i,2}$ avec pour tout λ , $\alpha_{e_{i,1}}(\lambda)_2 = \alpha_{e_{i,2}}(\lambda)_2 = 0$, $\alpha_{e_{i,1}}(\lambda)_1 = 1$ et $\alpha_{e_{i,2}}(\lambda)_1 = -1$.

Nous avons donc que pour tout préfixe ρ_{pref} de ρ_G , pour tout $\lambda \in \mathbb{A}^2$, $\phi^\lambda(0, \rho_{pref} \in \{0, 1\}$. Le chemin ρ est donc borné par $(\vec{0}, \vec{1})$.

De plus comme l'état initial de ρ_M est q_0 et que son état final est h , nous avons que ρ est un chemin de $q_{0,G}$ à h_G .

Par conséquent si M s'arrête et $max_C = 0$, pour tout λ nous avons ρ_G un chemin borné par $(\vec{0}, \vec{1})$ allant de $q_{0,G}$ à h_G .

Si $max_C \neq 0$, soit $\lambda^* = \min_{i \in \{1,2\}} (\frac{1}{max_i})$.

Soit $r_M = \sigma_0 \sigma_1 \dots \sigma_\ell$ une exécution dans M telle que M s'arrête. Soit $\rho_M = t_1 t_2 \dots t_\ell$ le chemin correspondant à r_M . Soit ρ_G le chemin de G associé à ρ_M .

Hypothèse de récurrence : Pour tout $k \in \llbracket 0, \ell \rrbracket$ et pour $i \in \llbracket 1, 2 \rrbracket$, $\phi(\rho_{G,k}, \lambda^*)_i = c_i(\rho_{M,k}) * \lambda^*$ et pour tout $\rho \in pref(\rho_{G,k})$, $0 \leq \phi(\rho, \lambda^*)_i \leq 1$.

Initialisation : Pour $k = 0$, pour $i \in \llbracket 1, 2 \rrbracket$ nous avons $\phi(\rho_{G,0}, \lambda^*)_i = 0$ et donc $0 \leq \phi(\rho_{G,0}, \lambda^*)_i \leq 1$. De plus comme la valeur initiale des compteurs est nulle nous avons $c_i(\rho_{M,0}) = 0$ et $c(\rho_{G,0}, \lambda^*)_i = c_i(\rho_{M,0}) * \lambda^*$. Comme $pref(\rho_{G,0}) = \{\rho_{G,0}\}$ l'hypothèse de récurrence est vraie pour $k = 0$.

Hérédité : Supposons l'hypothèse de récurrence vraie pour un $k \geq 0$. Donc pour $i \in \llbracket 1, 2 \rrbracket$ nous avons que, $\phi(\rho_{G,k}, \lambda^*)_i = c_i(\rho_{M,k}) * \lambda^*$ et pour tout $\rho \in pref(\rho_{G,k})$, $0 \leq \phi(\rho, \lambda^*)_i \leq 1$. Soit t_{k+1} la $k + 1$ -ème transition de ρ_M avec $\delta(t_{k+1}) = (C_{k+1}, \sim_{k+1})$. Il n'y a que huit possibilités :

1. $\sim_{k+1} = -$ et $C_{k+1} = C_1$,
2. $\sim_{k+1} = +$ et $C_{k+1} = C_1$,
3. $\sim_{k+1} = =$ et $C_{k+1} = C_1$,
4. $\sim_{k+1} = >$ et $C_{k+1} = C_1$,

5. $\sim_{k+1} = -$ et $C_{k+1} = C_2$,
6. $\sim_{k+1} = +$ et $C_{k+1} = C_2$,
7. $\sim_{k+1} = =$ et $C_{k+1} = C_2$,
8. $\sim_{k+1} = >$ et $C_{k+1} = C_2$.

Nous pouvons commencer par remarquer la symétrie des différents cas entre C_1 et C_2 et donc que les cas (5) à (8) sont symétriques aux cas (1) à (4). Dans la suite nous ne traiterons donc que les cas (1) à (4).

Cas (1). Nous avons $\delta(t_{k+1}) = (C_1, -)$ et $\rho_{G,k+1}$ n'a qu'un arc e de plus que $\rho_{G,k}$. Comme nous avons $\alpha_e(\lambda^*)_2 = 0$, $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho_{G,k}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et comme $0 \leq \phi(\rho_{G,k}, \lambda^*)_2 \leq 1$ nous avons également $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_2 \leq 1$.

Comme nous avons $\alpha_e(\lambda^*)_1 = -\lambda^*$, $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k}) - 1$, $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^*$ et $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^* + \alpha_e(\lambda^*)_1$, nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$ puisque $\alpha_e(\lambda^*)_1 = -\lambda^*$.

De plus, comme $c_1(\rho_{M,k+1}) \geq 0$ nous avons que $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_1$ et nous avons également $\phi(\rho_{G,k+1}, \lambda^*)_1 < \phi(\rho_{G,k}, \lambda^*)_1 \leq 1$. L'hypothèse de récurrence est donc vraie pour $k + 1$.

Cas (2). Nous avons $\delta(t_{k+1}) = (C_1, +)$ et $\rho_{G,k+1}$ n'a qu'un arc e de plus que $\rho_{G,k}$. Comme nous avons $\alpha_e(\lambda^*)_2 = 0$, $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho_{G,k}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et comme $0 \leq \phi(\rho_{G,k}, \lambda^*)_2 \leq 1$ nous avons également $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_2 \leq 1$.

Comme nous avons $\alpha_e(\lambda^*)_1 = \lambda^*$, $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k}) + 1$, $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^*$ et $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^* + \alpha_e(\lambda^*)_1$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$ puisque $\alpha_e(\lambda^*)_1 = \lambda^*$.

De plus, comme $\lambda^* = \min_{i \in \{1,2\}} (\frac{1}{\max_i})$ nous avons que $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_1 \leq 1$ et l'hypothèse de récurrence est vraie pour $k + 1$.

Cas (3). Nous avons $\delta(t_{k+1}) = (C_1, =)$, $\rho_{G,k+1}$ a deux arcs e_1 et e_2 de plus que $\rho_{G,k}$. Soient e_1 l'arc tel que $\alpha_{e_1}(\lambda^*)_1 = 1$ et e_2 l'arc tel que $\alpha_{e_2}(\lambda^*)_1 = -1$. Soit ρ' le chemin $\rho_{G,k}$ suivi de l'arc e_1 (c'est-à-dire $\rho_{G,k+1}$ sans son dernier arc : e_2). Comme nous avons $\alpha_{e_1}(\lambda^*)_2 = \alpha_{e_2}(\lambda^*)_2 = 0$, $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho', \lambda^*)_2 = \phi(\rho_{G,k+1}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et comme $0 \leq \phi(\rho_{G,k}, \lambda^*)_2 \leq 1$ nous avons également $0 \leq \phi(\rho', \lambda^*)_2 \leq 1$ et $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_2 \leq 1$.

Comme nous avons $\alpha_{e_1}(\lambda^*)_1 + \alpha_{e_2}(\lambda^*)_1 = 0$, $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k}) = 0$, $\phi(\rho_{G,k}, \lambda^*)_1 = 0$ et $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^* + \alpha_{e_1}(\lambda^*)_1 + \alpha_{e_2}(\lambda^*)_1$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_1 =$

$c_1(\rho_{M,k+1}) * \lambda^*$ puisque $\alpha_{e_1}(\lambda^*)_1 = 1$ et $\alpha_{e_2}(\lambda)_1 = -1$.

De plus nous avons que $\phi(\rho', \lambda^*)_1 = \phi(\rho_{G,k}, \lambda^*)_1 = 1$ et $0 = \phi(\rho_{G,k+1}, \lambda^*)_1$ et l'hypothèse de récurrence est vraie pour $k + 1$.

Cas (4). Nous avons $\delta(t_{k+1}) = (C_1, >)$, $\rho_{G,k+1}$ a deux arcs e_1 et e_2 de plus que $\rho_{G,k}$. Soient e_1 l'arc tel que $\alpha_{e_1}(\lambda^*)_1 = -\lambda^*$ et e_2 l'arc tel que $\alpha_{e_2}(\lambda^*)_1 = \lambda^*$. Soit ρ' le chemin $\rho_{G,k}$ suivi de l'arc e_1 (c'est-à-dire $\rho_{G,k+1}$ sans son dernier arc : e_2). Comme nous avons $\alpha_{e_1}(\lambda^*)_2 = \alpha_{e_2}(\lambda^*)_2 = 0$, $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho', \lambda^*)_2 = \phi(\rho_{G,k+1}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et comme $0 \leq \phi(\rho_{G,k}, \lambda^*)_2 \leq 1$ nous avons également $0 \leq \phi(\rho', \lambda^*)_2 \leq 1$ et $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_2 \leq 1$.

Comme nous avons $\alpha_{e_1}(\lambda^*)_1 + \alpha_{e_2}(\lambda^*)_1 = 0$, nous avons que $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k})$, $c_1(\rho_{M,k}) \geq 1$, $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^*$ et $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^* + \alpha_{e_1}(\lambda^*)_1 + \alpha_{e_2}(\lambda^*)_1$. Nous en déduisons que $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$ puisque $\alpha_{e_1}(\lambda^*)_1 = -\lambda^*$ et $\alpha_{e_2}(\lambda^*)_1 = \lambda^*$.

De plus, comme $c_1(\rho_{M,k}) \geq 1$ nous avons que $0 \leq \phi(\rho_{G,k}, \lambda^*)_1 - \lambda^* = \phi(\rho', \lambda^*)_1$ et nous avons également $\phi(\rho', \lambda^*)_1 < \phi(\rho_{G,k}, \lambda^*)_1 \leq 1$. Comme $\lambda^* = \min_{i \in \{1,2\}} (\frac{1}{\max_i})$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_1 \leq 1$ et nous avons aussi $0 \leq \phi(\rho', \lambda^*)_1 \leq \phi(\rho_{G,k+1}, \lambda^*)_1$. L'hypothèse de récurrence est donc vraie pour $k + 1$.

Comme l'hypothèse de récurrence est vraie dans tous les cas nous avons que l'hypothèse de récurrence est vraie pour $k + 1$.

Par conséquent si M s'arrête nous avons un chemin ρ_G de $q_{0,G}$ à h_G et une valuation des paramètres $\lambda^* \in \mathbb{Q}$ telle que pour tout chemin $\rho \in \text{pref}(\rho_G)$, pour $i \in \{1, 2\}$, $0 \leq \phi(\rho, \lambda^*)_i \leq 1$ et donc si M s'arrête alors la réponse à $\text{EnTot}_{LU}(G, (\vec{0}, \vec{1})_1^2, \mathbb{A}, q_{0,G}, h_G)$ est oui.

Réciproquement si $\exists \lambda > 0$ tel que $\lambda \in \mathbb{A}$ et ρ un chemin de G de v à w tel que ρ soit borné par $(\vec{0}, \vec{1})$ nous prouvons par induction que M s'arrête.

Considérons $\lambda^* \in \mathbb{A}$ tel que $\lambda^* > 0$ et un chemin ρ_G de $q_{0,G}$ à h_G dans G tel que $\forall \rho \in \text{pref}(\rho_G)$ et $i \in \{1, 2\}$, $0 \leq \phi(\rho, \lambda^*)_i \leq 1$. Puisque $h_G \notin Q_s$ nous avons $\rho_M = t_1 t_2 \dots t_\ell$ le chemin associé à ρ_G . Nous avons $\rho_G = \rho_1 \rho_2 \dots \rho_\ell$ tel que pour tout $i \in \llbracket 1, \ell \rrbracket$, ρ_i soit le chemin associé à t_i . Pour $j \in \llbracket 1, \ell \rrbracket$, nous appelons $\rho_{M,j} = t_1 \dots t_j$ le préfixe de longueur j de ρ_M , r_j l'exécution correspondant à $\rho_{M,j}$ et $\rho_{G,j} = \rho_1 \dots \rho_j$ le préfixe de ρ_G associé à $\rho_{M,j}$.

Hypothèse de récurrence : Pour tout $k \in \llbracket 0, \ell \rrbracket$ et pour $i \in \llbracket 1, 2 \rrbracket$, $c_i(\rho_{M,k}) \geq 0$ et $\phi(\rho_{G,k}, \lambda^*)_i = c_i(\rho_{M,k}) * \lambda^*$. La transition t_k est tirable si elle existe.

Initialisation : Dans le cas $k = 0$, pour $i \in \llbracket 1, 2 \rrbracket$ nous avons $\phi(\rho_{G,0}, \lambda^*)_i = 0$. Puisque la valeur initiale des compteurs est zéro nous avons $c_i(\rho_{M,0}) = 0$ et donc $\phi(\rho_{G,0}, \lambda^*)_i = c_i(\rho_{M,0}) * \lambda^*$ et $c_i(\rho_{M,0}) \geq 0$ pour $i \in \{1, 2\}$.

Hérédité : Supposons l'hypothèse de récurrence vraie pour un $k \geq 0$. Pour $i \in \llbracket 1, 2 \rrbracket$ nous avons donc $\phi(\rho_{G,k}, \lambda^*)_i = c_i(\rho_{M,k}) * \lambda^*$ et $c_i(\rho_{M,k}) \geq 0$. Soit t_{k+1} la $k+1$ -ème transition de ρ_M avec $\delta(t_{k+1}) = (C_{k+1}, \sim_{k+1})$. Il n'y a que 8 configurations possibles :

1. $\sim_{k+1} = -$ et $C_{k+1} = C_1$,
2. $\sim_{k+1} = +$ et $C_{k+1} = C_1$,
3. $\sim_{k+1} = =$ et $C_{k+1} = C_1$,
4. $\sim_{k+1} = >$ et $C_{k+1} = C_1$,
5. $\sim_{k+1} = -$ et $C_{k+1} = C_2$,
6. $\sim_{k+1} = +$ et $C_{k+1} = C_2$,
7. $\sim_{k+1} = =$ et $C_{k+1} = C_2$,
8. $\sim_{k+1} = >$ et $C_{k+1} = C_2$.

Nous pouvons remarquer comme précédemment la symétrie des différents cas entre C_1 et C_2 c'est-à-dire que les cas (5) à (8) sont symétriques aux cas (1) à (4). Dans la suite nous ne traiterons donc que les cas (1) à (4).

Cas (1). Nous avons $\delta(t_{k+1}) = (C_1, -)$, $\rho_{G,k+1}$ n'a qu'un arc e de plus que $\rho_{G,k}$. Comme nous avons $\alpha_e(\lambda^*)_2 = 0$, $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho_{G,k}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et comme $c_2(\rho_{M,k}) \geq 0$ nous avons $c_2(\rho_{M,k+1}) \geq 0$.

Comme nous avons $\alpha_e(\lambda^*)_1 = -\lambda^*$, $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k} - 1)$ et $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^*$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$. De plus comme $0 \leq \phi(\rho_{G,k+1}, \lambda^*)_1 \leq 1$ nous avons que $0 \leq c_1(\rho_{M,k+1}) * \lambda^*$ et donc $c_1(\rho_{M,k+1}) \geq 0$. Nous avons donc t_{k+1} tirable et l'hypothèse de récurrence est vraie pour $k + 1$.

Cas (2). Nous avons $\delta(t_{k+1}) = (C_1, +)$, $\rho_{G,k+1}$ n'a qu'un arc e de plus que $\rho_{G,k}$. Nous avons t_{k+1} tirable. Comme nous avons $\alpha_e(\lambda^*)_2 = 0$, $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho_{G,k}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et comme $c_2(\rho_{M,k}) \geq 0$ nous avons $c_2(\rho_{M,k+1}) \geq 0$.

Comme nous avons $\alpha_e(\lambda^*)_1 = \lambda^*$, $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k} + 1)$ et $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k}) * \lambda^*$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$. De plus comme $c_1(\rho_{M,k}) \geq 0$ et $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k}) + 1$, nous avons que $c_1(\rho_{M,k+1}) \geq 0$ et l'hypothèse de récurrence est vraie pour $k + 1$.

Cas (3). Nous avons $\delta(t_{k+1}) = (C_1, =)$, $\rho_{G,k+1}$ a deux arcs e_1 et e_2 de plus que $\rho_{G,k}$. Soient e_1 l'arc tel que $\alpha_{e_1}(\lambda^*)_1 = 1$ et e_2 l'arc tel que $\alpha_{e_2}(\lambda^*)_1 = -1$. Soit ρ' le chemin $\rho_{G,k}$ suivi de l'arc e_1 (c'est-à-dire $\rho_{G,k+1}$ sans son dernier arc : e_2). Comme nous avons $\alpha_{e_1}(\lambda^*)_2 = \alpha_{e_2}(\lambda^*)_2 = 0$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho_{G,k+1}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et puisque $c_2(\rho_{M,k}) \geq 0$ nous avons $c_2(\rho_{M,k+1}) \geq 0$.

Comme nous avons $\alpha_{e_1}(\lambda^*)_1 + \alpha_{e_2}(\lambda^*)_1 = 0$ et $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k})$ nous avons que $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$. De plus comme $0 \leq \phi(\rho_{G,k}, \lambda^*)_1 \leq 1$, $0 \leq \phi(\rho', \lambda^*)_1 \leq 1$ et $\alpha_{e_1}(\lambda)_1 = 1$ nous avons que $\phi(\rho_{G,k}, \lambda^*)_1 = 0$. Par conséquent $c_1(\rho_{M,k}) = 0$ ce qui signifie que $c_1(\rho_{M,k+1}) = 0$. Nous avons donc t_{k+1} tirable et l'hypothèse de récurrence est vraie pour $k + 1$.

Cas (4). Nous avons $\delta(t_{k+1}) = (C_1, >)$, $\rho_{G,k+1}$ a deux arcs e_1 et e_2 de plus que $\rho_{G,k}$. Soient e_1 l'arc tel que $\alpha_{e_1}(\lambda^*)_1 = -\lambda^*$ et e_2 l'arc tel que $\alpha_{e_2}(\lambda^*)_1 = \lambda^*$. Soit ρ' le chemin $\rho_{G,k}$ suivi de l'arc e_1 (c'est-à-dire $\rho_{G,k+1}$ sans son dernier arc : e_2). Comme nous avons $\alpha_{e_1}(\lambda^*)_2 = \alpha_{e_2}(\lambda^*)_2 = 0$ nous avons que $\phi(\rho_{G,k+1}, \lambda^*)_2 = \phi(\rho_{G,k+1}, \lambda^*)_2 = c_2(\rho_{M,k}) * \lambda^* = c_2(\rho_{M,k+1}) * \lambda^*$ et puisque $c_2(\rho_{M,k}) \geq 0$ nous avons $c_2(\rho_{M,k+1}) \geq 0$.

Comme nous avons $\alpha_{e_1}(\lambda^*)_1 + \alpha_{e_2}(\lambda^*)_1 = 0$ et $c_1(\rho_{M,k+1}) = c_1(\rho_{M,k})$ nous avons que $\phi(\rho_{G,k}, \lambda^*)_1 = c_1(\rho_{M,k+1}) * \lambda^*$. De plus comme $0 \leq \phi(\rho_{G,k}, \lambda^*)_1 \leq 1$, $0 \leq \phi(\rho', \lambda^*)_1 \leq 1$ et $\alpha_{e_1}(\lambda^*)_1 = -\lambda^*$ avec $\lambda^* > 0$ nous avons que $c_1(\rho_{M,k}) > 0$ ce qui signifie que $c_1(\rho_{M,k+1}) > 0$. Nous avons donc t_{k+1} tirable et l'hypothèse de récurrence est vraie pour $k + 1$.

Comme l'hypothèse de récurrence est vraie dans tous les cas nous avons que l'hypothèse de récurrence est vraie pour $k + 1$.

Nous avons donc ρ_M un chemin de M de q_0 à h dont toutes les transitions sont tirables et r son exécution correspondante dont la configuration initiale est $(q_0, 0, 0)$ et dont la configuration finale est $(h, c_1(\rho_M), c_2(\rho_M))$. Par conséquent si $\exists \lambda \in \mathbb{A}$ tel que $\lambda > 0$ et ρ un chemin de G de v à w tel que ρ soit borné par $(\vec{0}, \vec{1})$ nous avons un chemin valide dans M qui va de l'état initial à l'état d'arrêt et donc M s'arrête.

Comme nous avons déjà prouvé que si M s'arrête alors la réponse à $EnTot_{LU}(G, (\vec{0}, \vec{1}), \mathbb{A}, v, w)$ est oui. La décidabilité du problème $EnTot_{LU}(G, (\vec{0}, \vec{1}), \mathbb{A}, v, w)$ est donc au

moins aussi difficile que celle du problème de l'arrêt. Par conséquent, pour un graphe $G(V, E, d, n, (\mu_e, \alpha_e)_{e \in E})$, un ensemble \mathbb{A} de valeur des paramètres, des contraintes L et U dans \mathbb{R}^d , et u et v deux sommets de G , la question de savoir s'il existe $\vec{\lambda} \in \mathbb{A}^n$ et ρ un chemin de G de v à w tel que ρ soit borné par (L, U) est indécidable lorsque $n \geq 1$ pour tout \mathbb{A} tel que $\mathbb{Q} \subseteq \mathbb{A}$. \square

3.4 Conclusion

Nous avons considéré dans ce chapitre le problème LU total tout d'abord dans le cas où les paramètres prennent des valeurs entières et que l'énergie est de dimension quelconque, et dans un second temps lorsque les paramètres prennent des valeurs rationnelles et que l'énergie est de dimension supérieure ou égale à deux. Nous avons prouvé la décidabilité avec complexité PSPACE-complète du problème lorsque les paramètres sont à valeurs entières, quelle que soit la dimension de l'énergie, ainsi que l'indécidabilité du problème lorsque les paramètres prennent des valeurs dans tout ensemble qui comprend les nombres rationnels et que l'énergie est de dimension supérieure ou égale à deux, à partir du moment où au moins un paramètre intervient dans l'énergie des arcs.

Ces travaux pourraient être prolongés par l'étude de variantes de ces problèmes. Le fait que la preuve d'indécidabilité du problème LU total avec paramètres à valeurs rationnelles n'est plus valide lorsque l'énergie est de dimension 1 peut laisser penser que le problème pourrait être décidable dans ce cas. Nous pourrions aussi étudier les variations de ces problèmes cherchant à connaître les valeurs de paramètres pour que tous les chemins vérifient les conditions et non plus de chercher l'existence d'un chemin. Il serait également possible de se pencher sur d'autres types de problèmes, notamment lorsque l'on supprime la contrainte de la borne supérieure. En effet les gadgets utilisés pour tester l'égalité à zéro nécessitent que l'énergie des chemins soit bornée. Une variante de ce problème lorsque l'énergie est mono-dimensionnelle sera étudiée dans le chapitre suivant.

JEU DE PARITÉ À ÉNERGIE PARAMÉTRÉE

4.1 Introduction

Les jeux sur les graphes ont un grand nombre d'applications possibles, notamment les problèmes de synthèse de contrôleur pour les systèmes réactifs. Dans ce contexte, en plus de propriétés qualitatives comme la vivacité ou la sûreté, nous voulons habituellement nous assurer de certaines propriétés quantitatives, qui peuvent par exemple avoir un lien avec l'énergie ou le temps.

En parallèle, dans les phases de développement précoces d'un système, il existe beaucoup d'aspects quantitatifs qui ne sont pas encore finalisés à ce stade. Par exemple, le modèle précis d'un microcontrôleur qui sera utilisé n'a pas encore été choisi, et par conséquent la consommation d'énergie n'est pas précisément connue. Il est cependant bien connu qu'il est d'autant plus facile et économique de fixer le design d'un système que l'on est tôt dans le cycle de développement. Dans cette optique, nous voulons modéliser des fonctionnalités quantitatives inconnues avec des paramètres en nous demandant quelles valeurs de paramètres nous assurent de l'existence d'un contrôleur.

Dans ce chapitre, nous considérerons des jeux à deux joueurs avec un double objectif : un objectif quantitatif sur l'énergie qui consiste à vouloir que la valeur de l'énergie reste au dessus d'un seuil donné comme dans les précédentes sections, ainsi qu'un objectif qualitatif sur la parité qui consiste à attribuer un nombre entier à chaque sommet que l'on appelle parité et à imposer que la parité minimale apparaissant infiniment souvent soit paire, d'où le nom. Les jeux que nous considérerons seront donc sur des graphes orientés à énergie paramétrée.

Un problème similaire a été étudié par Chatterjee et Doyen dans [CD12] dans sa version non paramétrée. En plus de ces paramètres, notre modèle comporte quelques différences importantes comme le fait que le jeu débute avec un niveau fixe d'énergie (0), tandis qu'ils

cherchaient un niveau d'énergie initial qui permettait l'existence d'une stratégie gagnante. Nous pourrions aisément modéliser cela par un paramètre, mais c'est également possible dans un modèle non paramétré avec une boucle initial permettant d'augmenter l'énergie jusqu'à une valeur arbitrairement haute.

Plusieurs travaux de Ouaknine et al. considèrent des automates à un compteur paramétré, avec des propriétés d'accessibilité [Haa+09; BO14] ou des propriétés de logiques temporelles (CTL, LTL, μ -calcul) [Göl+10]. Bien que cela soit assez proche de notre modèle, avec le compteur représentant la valeur de l'énergie, une différence subtile bien qu'importante est qu'il ne semble pas immédiat de modéliser le test à zéro de la valeur de l'énergie dans notre cas.

Nous prouvons dans ce chapitre la décidabilité ainsi que des bornes de la complexité du problème dans les cas à un et à deux joueurs. Dans le cas à un joueur, qui consiste à étudier des propriétés des chemins dans un graphe comme nous l'avons fait dans les chapitres précédents, nous ajoutons au modèle des arcs qui fixent la valeur de l'énergie quelle qu'ait été sa valeur précédente. Nous montrons que le problème dans sa version à un joueur est NP-complet et que le problème à deux joueurs est PSPACE-difficile ainsi que dans NEXPTIME.

Dans ce chapitre, après avoir défini les concepts qui nous seront utiles pour traiter des jeux sur les graphes paramétrés dans la section 4.2 nous commencerons par considérer le problème dans le cas où il n'y a qu'un seul joueur dans la section 4.3 en prenant en compte des arcs qui fixent l'énergie à une valeur donnée indépendamment de sa valeur précédente. Nous aborderons ensuite le problème dans sa version avec deux joueurs dans la section 4.4 et enfin nous discuterons des résultats que nous avons sur ce problème dans la section 4.5.

4.2 Définitions

Nous ne considérerons dans ce chapitre que des graphes $G = (V, E, d, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ à une seule dimension, c'est-à-dire tels que $d = 1$. Nous rappelons qu'un arc $e \in E$ sera un *arc croissant* si $\mu_e \geq 1$ et sera un *arc constant* si $\mu_e = 0$.

Nous prendrons également $\mathbb{A} = \mathbb{Z}$. De plus comme nous allons considérer des jeux à deux joueurs sur des graphes, l'ensemble V sera partitionné en deux parties V_1 et V_2 représentant respectivement les sommets contrôlés par le joueur 1 et ceux contrôlés par le joueur 2.

Nous présenterons donc les jeux de parités à énergie paramétrée sous la forme d'un 7-uplet : $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$.

Un chemin sera dit *croissant* si tous les arcs qui le composent sont croissants.

Nous rappelons que pour tout $u \in \mathbb{Z}^\omega$, $\inf(u)$ est l'ensemble des entiers qui apparaissent infiniment souvent dans u .

Dans les problèmes que nous considérerons nous chercherons à savoir s'il existe une manière pour le joueur 1 d'être sûr de créer un chemin qui vérifie certaines propriétés. Comme ils sont l'objectif du joueur 1 nous appellerons les chemins qui vérifient ces propriétés des chemins gagnants.

Définition 4.1 (Chemin gagnant). *Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Pour tout $x \in \mathbb{Z}^n$, nous définissons $W^x \subseteq E^\omega$ l'ensemble des chemins gagnants par $\rho \in W^x$ si et seulement si $\min \circ \inf \circ \pi(\rho)$ est pair et pour tout $\rho' \in \text{Pref}(\rho)$ on a $0 \leq \phi^x(\rho')$. S'il existe $x \in \mathbb{Z}^n$ tel que $W^x \neq \emptyset$ alors on dit que le joueur 1 gagne.*

Nous pouvons remarquer que les propriétés vérifiées par un chemin gagnant sont la combinaison d'une condition sur la parité des chemins ainsi que d'une condition sur l'énergie totale des chemins qui est analogue au problème *EnTot_L* pour les chemins infinis.

4.3 Cas à un joueur

Dans cette section, comme il n'y a qu'un seul joueur nous avons $V_2 = \emptyset$.

4.3.1 Définitions

Nous définissons dans cette section des notions qui ne nous serviront que dans le cas des jeux à un joueur.

Définition 4.2 (Chemin simple pour les arcs constants). *Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur, soit $\rho = e_1 \dots e_k$ un chemin fini et soient e_{i_1}, \dots, e_{i_m} ses arcs constants par ordre d'occurrence. Le chemin ρ est un chemin simple pour les arcs constants si et seulement si ρ est le chemin vide ou si pour tout $(j, l) \in \llbracket 1, m \rrbracket^2$ tel que $j \neq l$ nous avons $e_{i_j} \neq e_{i_l}$.*

Intuitivement un chemin simple pour les arcs constants est un chemin où chaque arc constant apparaît au plus une fois.

Définition 4.3 (Schéma d'itération et chemin de schéma d'itération). Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Soit $\rho = e_1 \dots e_n$ un chemin fini, ρ est un chemin de schéma d'itération s'il est croissant, qu'il y a ρ_p et ρ_s , deux chemins simples de G et ρ_c un cycle simple de G ou le chemin vide, tels que $\rho = \rho_p \rho_c^k \rho_s$ avec $k \geq 1$. Le triplet (ρ_p, ρ_c, ρ_s) tel que ρ_p ait une longueur minimale parmi tous les ρ_p possibles est le schéma d'itération de ρ .

4.3.2 Transformation d'un chemin en schéma d'itération

Nous commençons par mettre en place des transformations permettant de créer un schéma d'itération à partir d'un chemin croissant et nous prouvons ensuite plusieurs propriétés des schémas d'itération ainsi créés.

Nous cherchons à construire des schémas d'itération car, comme le montre le lemme suivant, les schémas d'itération sont en nombre fini ce qui nous permet de pouvoir les énumérer.

Lemme 4.1. Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur, il y a un nombre fini de schémas d'itération.

Démonstration. Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur et soit $n = |V|$. Comme un chemin simple est au plus de longueur n , il n'y a qu'un nombre fini de chemins simples. De manière similaire, un cycle simple est de longueur inférieure ou égale à $n + 1$, il y a donc un nombre fini de cycles simples. Par conséquent, il n'y a qu'un nombre fini de schémas d'itérations dans G . \square

Nous commençons par définir la transformation θ qui, à partir d'un chemin croissant donné ρ et de $x \in \mathbb{Z}^n$, retire tous les cycles de ρ , à l'exception du premier cycle de ρ qui augmente strictement l'énergie. Nous utilisons ensuite θ pour définir Θ , qui construit un schéma d'itération depuis ρ grâce à $\theta(\rho)$.

Commençons par décrire le fonctionnement de la transformation θ avec un exemple avant de la définir formellement. Nous considérerons dans cet exemple le graphe G_θ^x représenté dans la figure 4.1.

Nous allons étudier la transformation du chemin ρ_θ , de longueur 9. Ce chemin a pour sommet initial s et sommet terminal t et est représenté dans la figure 4.2. Pour plus de fluidité nous appellerons dans la suite de cet exemple :

- C_1 le cycle simple $e_3 e_4$,

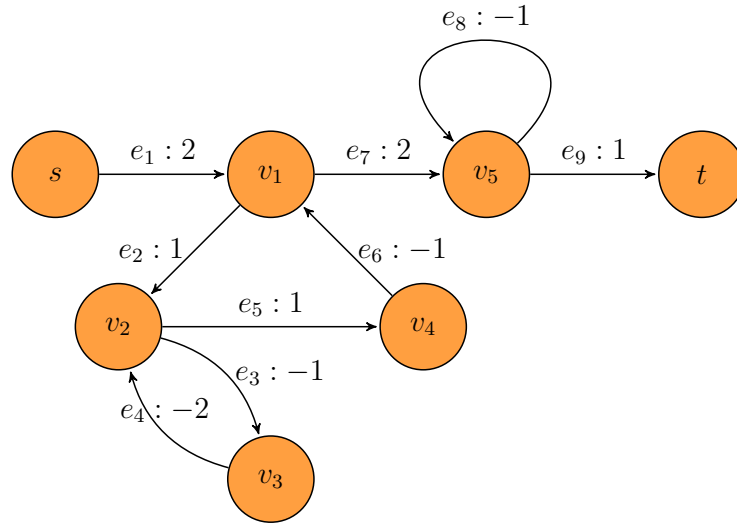


FIGURE 4.1 – Un graphe G_θ^x .

- C_2 le cycle simple $e_2e_5e_6$,
- C_3 le cycle simple e_8 ,
- C_4 le cycle $e_2e_3e_4e_5e_6$.

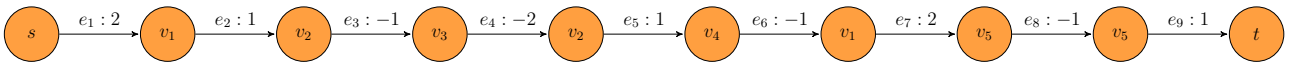


FIGURE 4.2 – Le chemin ρ_θ .

La transformation θ cherche à identifier un cycle simple qui augmente strictement l'énergie du chemin et à supprimer tous les autres cycles du chemin. Pour cela nous allons parcourir ρ_θ en commençant par son sommet initial.

Le premier cycle simple identifié est C_1 , après avoir considéré les quatre premiers arcs de ρ_θ . Nous regardons alors quel impact a C_1 sur l'énergie lorsqu'il est parcouru et puisqu'il diminue l'énergie de 3 à chaque emprunt la transformation θ supprime C_1 de ρ_θ . Nous avons alors le chemin ρ'_θ représenté sur la figure 4.3.

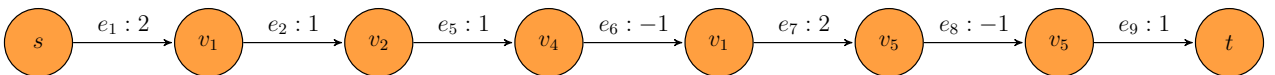


FIGURE 4.3 – Le chemin ρ'_θ .

Comme nous n'avons pas terminé de parcourir ρ'_θ suite à la suppression de C_1 nous

continuons à énumérer les arcs de ρ'_θ à la suite des arcs e_1 et e_2 que nous avons déjà considéré et que nous n'avons pas retiré.

Nous identifions un second cycle simple après avoir parcouru deux arcs supplémentaires, il s'agit de C_2 . Nous constatons alors qu'à chaque emprunt, C_2 augmente l'énergie du chemin de 1, nous avons donc trouvé un cycle augmentant strictement l'énergie. A partir de ce moment nous ne cherchons plus qu'à supprimer tous les éventuels cycles du suffixe de ρ'_θ commençant à la fin de C_2 .

Lors de cette étape nous identifions alors un troisième et dernier cycle simple, C_3 . Nous le supprimons indépendamment de son impact sur l'énergie puisque nous avons déjà identifié le cycle C_2 qui l'augmente strictement à chaque emprunt. Une fois que nous avons considéré tous les arcs de ρ_θ , le procédé se termine et nous obtenons alors $\theta(\rho_\theta)$, représenté sur la figure 4.4.

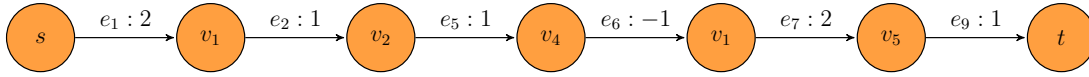


FIGURE 4.4 – Le chemin $\theta(\rho_\theta)$.

Nous pouvons constater que dans le chemin ρ_θ les cycles C_1 , C_3 et C_4 sont directement présents et ils diminuent tous trois l'énergie du chemin lorsqu'ils sont empruntés. La transformation θ , en ne considérant que les cycles simples permet cependant d'identifier tout de même le cycle C_2 qui augmente strictement l'énergie lorsqu'il est parcouru.

La transformation θ ayant identifié et isolé le premier cycle simple augmentant strictement l'énergie, nous pouvons ensuite créer un schéma d'itération avec Θ . Pour cela il suffit d'isoler le cycle conservé par θ et de prendre le préfixe de $\theta(\rho_\theta)$ se terminant au sommet initial du cycle ainsi que le suffixe de $\theta(\rho_\theta)$ commençant au sommet final du cycle. Ainsi pour notre exemple nous avons $\Theta(\rho_\theta) = (e_1, C_2, e_7e_9)$.

Formalisons à présent les transformations que nous venons de présenter sur l'exemple de ρ_θ avec les définitions suivantes.

Définition 4.4 (Transformation θ). *Fixons $x \in \mathbb{Z}^n$. Soit $\gamma = e_1 \dots e_\ell$ un chemin croissant fini de G^x avec $e_k = (v_k, v_{k+1})$ pour $k \in \llbracket 1, \ell \rrbracket$. Soit $y \in \mathbb{Z}$. Nous construisons le chemin $\theta^x(y, \gamma) = \theta_\ell^x(y, \gamma)$ de la manière itérative suivante :*

- $\theta_0^x(y, \gamma) = \varepsilon$ et $m_0 = \ell$,
- Pour $k \in \llbracket 1, \ell \rrbracket$, $\vartheta_k^x(y, \gamma) = \theta_{k-1}^x(y, \gamma)e_k = e_{k_1} \dots e_{k_p}$,
- Si $k \leq m_{k-1}$:
 - Si $\exists i \in \llbracket 1, p \rrbracket$ tel que $v_{k_i} = v_{k_p+1}$, soit $\vartheta_{k,s}^x(y, \gamma) = e_{k_1} \dots e_{k_{i-1}}$:

- Si $\phi^x(y, \vartheta_{k,s}^x(y, \gamma)) < \phi^x(y, \vartheta_k^x(y, \gamma))$, alors $\theta_k^x(y, \gamma) = \vartheta_k^x(y, \gamma)$ et $m_k = k$,
- Si $\phi^x(y, \vartheta_{k,s}^x(y, \gamma)) \geq \phi^x(y, \vartheta_k^x(y, \gamma))$, alors $\theta_k^x(y, \gamma) = \vartheta_{k,s}^x(y, \gamma)$ et $m_k = \ell$,
- Si $\forall i \in \llbracket 1, p \rrbracket, v_{k_i} \neq v_{k_{p+1}}$, alors $\theta_k^x(y, \gamma) = \vartheta_k^x(y, \gamma)$ et $m_k = \ell$,
- Si $k > m_{k-1}$ alors $m_k = m_{k-1}$ et on a $c = |\theta_{m_{k-1}}^x(y, \gamma)|$,
 - Si $\exists i \in \llbracket c+1, p \rrbracket$ tel que $v_{k_i} = v_{k_{p+1}}$, alors $\theta_k^x(y, \gamma) = e_{k_1} \dots e_{k_{i-1}}$,
 - Si $\forall i \in \llbracket c+1, p \rrbracket, v_{k_i} \neq v_{k_{p+1}}$, alors $\theta_k^x(y, \gamma) = \vartheta_k^x(y, \gamma)$.

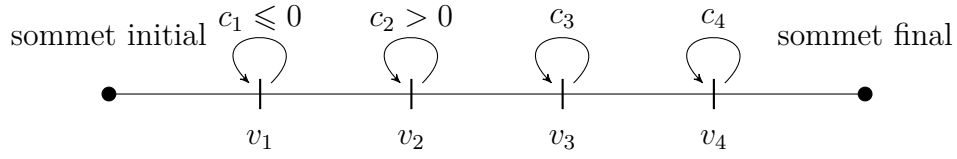
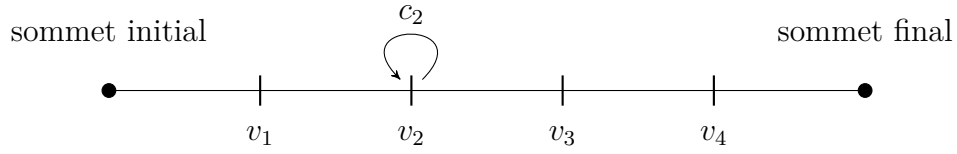

 (a) - Un chemin ρ

 (b) - Le chemin $\theta^x(\rho)$

 FIGURE 4.5 – Un chemin ρ et le chemin $\theta^x(\rho)$.

Nous résumons schématiquement l'action de la transformation θ avec la figure 4.5. Nous avons représenté dans la figure 4.5 (a) un exemple de chemin ρ ayant quatre cycles simples, les c_i pour $i \in \llbracket 1, 4 \rrbracket$. Nous appelons v_i le sommet initial (et final) du cycle c_i . Le cycle c_1 est tel qu'il diminue l'énergie totale du chemin : pour tout chemin γ et tout $y \in \mathbb{Z}$, $\psi^x(y, \gamma) \geq \psi^x(y, \gamma c_1)$. Le cycle c_2 quant à lui augmente strictement l'énergie totale du chemin ce qui signifie que, contrairement à c_1 , pour tout chemin γ et tout $y \in \mathbb{Z}$ nous avons $\psi^x(y, \gamma) \geq \psi^x(y, \gamma c_2)$. Les modifications de l'énergie provoquées par les cycles c_3 et c_4 peuvent être quelconques. En effet lors de la construction du chemin $\theta^x(\rho)$ le premier cycle identifié sera le cycle c_1 , or puisqu'il diminue l'énergie totale du chemin il ne sera pas conservé dans $\theta^x(\rho)$. Le second cycle considéré sera c_2 , qui augmente strictement l'énergie du chemin. Une fois c_2 identifié la fin de la construction de $\theta^x(\rho)$ consiste uniquement en la suppression des cycles dans le suffixe de ρ commençant en v_2 , le sommet final de c_2 .

Nous pouvons observer cela sur la figure 4.5 (b) qui représente le chemin $\theta^x(\rho)$ ainsi obtenu. Lors de la construction de $\theta^x(\rho)$ le cycle c_1 est bien supprimé puisqu'il diminue l'énergie totale. Nous trouvons bien c_2 dans le chemin $\theta^x(\rho)$ puisque c'est un cycle qui augmente l'énergie ce qui entraîne que les cycles c_3 et c_4 soient absents de $\theta^x(\rho)$ quels que soient les modifications qu'ils appliquent à l'énergie de ρ .

Définition 4.5 (Transformation Θ). *Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Fixons $x \in \mathbb{Z}^n$. Soient $\gamma = e_1 \dots e_\ell$ un chemin fini croissant de G^x et $y \in \mathbb{Z}$. Soient $\theta^x(y, \gamma) = e_1 \dots e_p$ et m_ℓ tels que définis précédemment. Nous définissons $\Theta^x(y, \gamma)$ de la manière suivante :*

- Si $m_\ell = \ell$ et $v_1 = v_{\ell+1}$, alors $\Theta^x(y, \gamma) = (\varepsilon, \varepsilon, \varepsilon)$. Avec ε le chemin vide.
- Si $m_\ell = \ell$ et $v_1 \neq v_{\ell+1}$, alors $\Theta^x(y, \gamma) = (\varepsilon, \varepsilon, \theta^x(y, \gamma))$.
- Si $m_\ell = k < \ell$ alors nous avons $c = |\theta_k^x(y, \gamma)|$ et $i \in \llbracket 1, c \rrbracket$ tel que $v_i = v_{c+1}$, soit $\Theta^x(y, \gamma) = (e_1 \dots e_{i-1}, e_i \dots e_c, e_{c+1} \dots e_p)$ (notons que selon les valeurs de i et c , il est possible d'avoir des chemins vides dans $\Theta^x(y, \gamma)$).

Nous commençons par démontrer plusieurs propriétés des transformations θ et Θ . Nous montrons que la transformation θ ne fait que retirer des cycles de γ sans ajouter d'arcs et qu'en notant $\Theta^x(y, \gamma) = (\gamma_p, \gamma_c, \gamma_s)$ nous avons que γ_p et γ_s sont des chemins simples et γ_c est un cycle simple qui augmente strictement l'énergie.

Lemme 4.2. *Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Fixons $x \in \mathbb{Z}^n$. Soient $\gamma = e_1 \dots e_\ell$ un chemin fini croissant de G^x et $y \in \mathbb{Z}$. Soient $\theta^x(y, \gamma)$ et $\Theta^x(y, \gamma) = (\gamma_p, \gamma_c, \gamma_s)$ tels que définis précédemment.*

Nous prouvons les propriétés suivantes :

- (a) - La transformation θ^x ne supprime que des cycles.
- (b) - Pour tout arc $e \in \theta^x(y, \gamma)$, $e \in \gamma$.
- (c) - γ_c est soit le chemin vide, soit un cycle simple qui augmente strictement l'énergie.
- (d) - γ_p et γ_s sont des chemins simples.

Démonstration. Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Fixons $x \in \mathbb{Z}^n$. Soient $\gamma = e_1 \dots e_\ell$ un chemin fini croissant de G^x et $y \in \mathbb{Z}$. Soient $\theta^x(y, \gamma) = e_1 \dots e_p$ et m_ℓ tels que définis précédemment avec $e_i = (v_i, v_{i+1})$ pour tout $i \in \llbracket 1, p \rrbracket$. Pour $k \in \llbracket 0, \ell \rrbracket$ nous aurons $\theta_k^x(y, \gamma) = e_{k_1} \dots e_{k_q}$. Soit $\Theta^x(y, \gamma) = (\gamma_p, \gamma_c, \gamma_s)$. Nous noterons ε le chemin vide.

Nous prouvons les résultats par récurrence :

Hypothèse de récurrence : Pour tout $k \in \llbracket 0, \ell \rrbracket$, $\theta_k^x(y, \gamma)$ vérifie :

- (1) - Si $k \geq 1$ et $|\theta_k^x(y, \gamma)| \leq |\theta_{k-1}^x(y, \gamma)|$ alors le chemin supprimé à l'étape k est un cycle.
- (2) - Pour tout arc $e \in \theta^x(y, \gamma)$, $e \in \gamma$.
- (3) - Si $m_k = \ell$ alors $\theta_k^x(y, \gamma)$ est un chemin simple.
- (4) - Si $m_k \neq \ell$, avec $c = |\theta_{m_k}^x(y, \gamma)|$ et $i_k \in \llbracket 1, c \rrbracket$ tels que $v_{i_k} = v_{c+1}$ nous avons :
 - (4.1) - $\gamma_{p,k} = e_{k_1} \dots e_{k_{i_k-1}}$ est un chemin simple.
 - (4.2) - $\gamma_{s,k} = e_{k_{c+1}} \dots e_{k_q}$ est un chemin simple.
 - (4.3) - $\gamma_{c,k} = e_{k_{i_k}} \dots e_{k_c}$ est un cycle simple.
 - (4.4) - $\gamma_{c,k}$ augmente strictement l'énergie.

Lorsque (4.1), (4.2), (4.3) et (4.4) seront vraies nous résumerons en notant que (4) est vraie.

Nous pouvons remarquer que d'après la définition de Θ , si (3) et (4) sont vraies pour l'étape ℓ alors (c) et (d) sont vraies.

Initialisation : Pour $k = 0$ nous avons (1) vraie. Puisque $\theta_0^x(y, \gamma) = \varepsilon$ (2) est vraie et comme $m_0 = \ell$ nous avons (4) vraie et (3) est également vraie car ε est un chemin simple. L'hypothèse de récurrence est donc vraie pour $k = 0$.

Hérédité : Supposons l'hypothèse de récurrence vraie pour un $k \geq 0$. Cela signifie que $\theta_k^x(y, \gamma)$ vérifie (1), (2), (3) et (4).

Soit $\vartheta_{k+1}^x(y, \gamma) = \theta_k^x(y, \gamma)e_{k+1} = e_{k+1} \dots e_{k+1_{q'}}$.

Dans le cas où il existe $j \in \llbracket 1, q' \rrbracket$ tel que $v_j = v_{q'+1}$, nous noterons $\vartheta_{k+1,s}^x(y, \gamma) = e_{k_1} \dots e_{k+1_{j-1}}$.

Cas $m_{k+1} = \ell$ Si $m_{k+1} = \ell$ alors cela signifie que $m_k = \ell$ et nous avons donc $k+1 \leq m_k$. Il n'y a que deux possibilités pour que cela se produise, la première lorsqu'il existe $j \in \llbracket 1, q' \rrbracket$ tel que $v_j = v_{q'+1}$ et $\phi^x(y, \vartheta_{k+1,s}^x(y, \gamma)) \geq \phi^x(y, \vartheta_{k+1}^x(y, \gamma))$, et la seconde lorsque pour tout $j \in \llbracket 1, q' \rrbracket$, $v_j \neq v_{q'+1}$.

Dans tous les cas nous avons (4) vérifiée par $\theta_{k+1}^x(y, \gamma)$ puisque $m_{k+1} = \ell$.

Supposons tout d'abord que pour tout $j \in \llbracket 1, q' \rrbracket$, $v_j \neq v_{q'+1}$. Nous avons alors $\theta_{k+1}^x(y, \gamma) = \theta_k^x(y, \gamma)e_{k+1}$ et (1) est vérifiée puisque aucun arc n'a été retiré lors de l'étape $k + 1$. De plus comme l'hypothèse de récurrence est vraie pour k , nous avons que $\theta_k^x(y, \gamma)$ est un chemin simple dont tous les arcs appartiennent à γ . Puisque

$e_{k+1} \in \gamma$ nous avons (2) vérifiée par $theta_{k+1}^x(y, \gamma)$. Enfin comme nous avons que pour tout $j \in \llbracket 1, q' \rrbracket$, $v_j \neq v_{q'+1}$, nous avons que $theta_{k+1}^x(y, \gamma)$ est un chemin simple et (3) est bien vérifiée.

Supposons maintenant qu'il existe $j \in \llbracket 1, q' \rrbracket$ tel que $v_j = v_{q'+1}$ et $\phi^x(y, \vartheta_{k+1,s}^x(y, \gamma)) \geq \phi^x(y, \vartheta_{k+1}^x(y, \gamma))$. Nous avons donc $theta_{k+1}^x(y, \gamma) = \vartheta_{k+1,s}^x(y, \gamma) = e_{k+1} \dots e_{k+1_{j-1}}$ et le chemin retiré à l'étape $k+1$ est $\gamma_{sup} = e_{k+1_j} \dots e_{k+1_{q'}}$. Hors comme $v_j = v_{q'+1}$ nous avons que γ_{sup} est un cycle et donc (1) est bien vérifiée. De plus comme $theta_{k+1}^x(y, \gamma)$ est un préfixe de $theta_k^x(y, \gamma)$, nous avons bien que $theta_{k+1}^x(y, \gamma)$ est un chemin simple dont tous les arcs appartiennent à γ et donc (2) et (3) sont vérifiées.

Par conséquent l'hypothèse de récurrence est vraie pour $k+1$ lorsque $m_{k+1} = \ell$.

Cas $m_{k+1} \neq \ell$ et $m_k = \ell$ Si $m_{k+1} \neq \ell$ et $m_k = \ell$ alors cela signifie qu'il existe $i_{k+1} \in \llbracket 1, q' \rrbracket$ tel que $v_{i_{k+1}} = v_{q'+1}$ et $\phi^x(y, \vartheta_{k+1,s}^x(y, \gamma)) < \phi^x(y, \vartheta_{k+1}^x(y, \gamma))$. Nous avons donc $m_{k+1} = k+1$ et cela signifie également que le cycle $\gamma_{c,k+1} = e_{k_{i_{k+1}}} \dots e_{k_{q'}}$ augmente strictement l'énergie et donc (4.4) est vérifiée. Puisque aucun arc n'est supprimé à cette étape nous avons également (1) vérifiée et comme $m_{k+1} \neq \ell$, (3) est également vérifiée. Puisque l'hypothèse de récurrence est vraie à l'étape k , que $e_{k+1} \in \gamma$ et que $\theta_{k+1}^x(y, \gamma) = theta_k^x(y, \gamma)e_{k+1}$ nous avons que (2) est vraie à l'étape $k+1$.

Puisque l'hypothèse de récurrence est vraie à l'étape précédente et que $m_k = \ell$, nous avons que $theta_k^x(y, \gamma)$ est un chemin simple. Nous avons donc $\gamma_{p,k+1} = e_{k+1_1} \dots e_{k+1_{i_{k+1}-1}}$ et $e_{k_{i_{k+1}}} \dots e_{k_{q'-1}}$ des chemins simples. Par conséquent (4.1) et (4.3) sont vérifiées à l'étape $k+1$. De plus nous avons $\gamma_{s,k+1}$ le chemin vide donc (4.2) est également vérifiée.

Par conséquent l'hypothèse de récurrence est vraie pour $k+1$ lorsque $m_{k+1} \neq \ell$ et $m_k = \ell$.

Cas $m_{k+1} \neq \ell$ et $m_k \neq \ell$ Soit $c = |\theta_{m_k}^x(y, \gamma)|$, si $m_{k+1} \neq \ell$ et $m_k \neq \ell$ alors soit il existe $j \in \llbracket c+1, q' \rrbracket$ tel que $v_{k+1_j} = v_{k+1_{q'+1}}$, soit pour tout $j \in \llbracket c+1, q' \rrbracket$ nous avons $v_{k+1_j} \neq v_{k+1_{q'+1}}$.

Dans les deux cas, puisque $m_{k+1} \neq \ell$, (3) est vérifiée à l'étape $k+1$.

Dans les deux cas, le préfixe de taille c de $theta_{k+1}^x(y, \gamma)$ et celui de $theta_k^x(y, \gamma)$ sont identiques et puisque l'hypothèse de récurrence est vraie pour k , nous avons donc que (4.1), (4.3) et (4.4) sont vérifiées à l'étape $k+1$.

Supposons tout d'abord qu'il existe $j \in \llbracket c+1, q' \rrbracket$ tel que $v_{k+1_j} = v_{k+1_{q'+1}}$. Nous avons alors $theta_{k+1}^x(y, \gamma) = e_{k+1_1} \dots e_{k+1_{j-1}}$, $\gamma_{s,k+1} = e_{k+1_{c+1}} \dots e_{k+1_{j-1}}$ et le chemin supprimé à

cette étape est $\gamma_{sup} = e_{k+1_j} \dots e_{k+1_{q'}}$. Puisque tous les arcs de $theta_{k+1}^x(y, \gamma)$ sont des arcs de $theta_k^x(y, \gamma)$ et que l'hypothèse de récurrence est vraie pour k , nous avons que tous les arcs de $theta_{k+1}^x(y, \gamma)$ appartiennent à γ et (2) est donc vérifiée à l'étape $k + 1$. De plus nous avons $v_{k+1_j} = v_{k+1_{q'+1}}$ donc γ_{sup} est un cycle et (1) est vérifiée. Enfin, puisque (4.2) est vraie à l'étape k nous avons que $\gamma_{s,k} = e_{k+1_{c+1}} \dots e_{k+1_{q'-1}}$ est un chemin simple, par conséquent $\gamma_{s,k+1}$ est un chemin simple et (4.2) est vérifiée à l'étape $k + 1$.

Supposons maintenant que pour tout $j \in \llbracket c + 1, q' \rrbracket$ nous avons $v_{k+1_j} \neq v_{k+1_{q'+1}}$. Puisque aucun arc n'est supprimé nous avons bien (1) vérifiée. Nous avons $theta_{k+1}^x(y, \gamma) = theta_k^x(y, \gamma)e_{k+1}$, comme (2) est vérifiée à l'étape k et $e_{k+1} \in \gamma$, (2) est bien vérifiée à l'étape $k + 1$. Enfin nous avons $\gamma_{s,k+1} = e_{k+1_{c+1}} \dots e_{k+1_{q'}}$ et d'après l'hypothèse de récurrence nous avons également que $\gamma_{s,k} = e_{k+1_{c+1}} \dots e_{k+1_{q'-1}}$ est un chemin simple, puisque pour tout $j \in \llbracket c + 1, q' \rrbracket$ nous avons $v_{k+1_j} \neq v_{k+1_{q'+1}}$ la propriété (4.2) est vérifiée à l'étape $k + 1$.

Par conséquent l'hypothèse de récurrence est vraie pour $k + 1$ lorsque $m_{k+1} \neq \ell$ et $m_k \neq \ell$.

Puisque l'hypothèse de récurrence est vraie pour $k + 1$ dans tous les cas nous avons que l'hypothèse de récurrence est vraie pour $k + 1$.

Soient $x \in \mathbb{Z}^n$, γ un chemin et $\Theta^x(y, \gamma) = (\gamma_p, \gamma_c, \gamma_s)$. Nous avons donc montré que la transformation θ^x ne supprime que des cycles de γ et n'ajoute aucun arc, que γ_p et γ_s sont des chemins simples et que γ_c est soit le chemin vide soit un cycle simple qui augmente strictement l'énergie. \square

Nous prouvons avec les lemmes suivants que pour $x \in \mathbb{Z}^n$ et $y \in \mathbb{Z}$, nous pouvons construire un schéma d'itération $\Theta^x(y, \rho)$ depuis un chemin fini croissant ρ (lemme 4.3). Nous prouvons également que grâce à $\Theta^x(y, \rho)$ nous pouvons construire un chemin qui augmente plus l'énergie que ρ tout en ayant les mêmes sommets initiaux et finaux. De plus si tous les préfixes de ρ ont une énergie positive alors ce sera également le cas du chemin construit de cette manière (lemme 4.4). Enfin nous prouvons avec le lemme 4.5 que le chemin construit de cette manière a une parité minimale supérieure ou égale à la parité minimale de ρ .

Pour les trois lemmes suivants ainsi que leurs preuves nous considérerons $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur, un chemin fini croissant γ , une valeur des paramètres $x \in \mathbb{Z}^n$ et un entier $y \in \mathbb{N}$. On note $\Theta^x(y, \gamma) = (\rho_p, \rho_c, \rho_s)$.

Lemme 4.3. *Nous avons que $\Theta^x(y, \gamma)$ est un schéma d'itération.*

Démonstration. Commençons par le cas où $\rho_c = \varepsilon$. Dans ce cas nous avons donc $m_\ell = \ell$ dans $\theta^x(y, \gamma)$ ce qui implique que $\Theta^x(y, \gamma) = (\varepsilon, \varepsilon, \varepsilon)$ ou $\Theta^x(y, \gamma) = (\varepsilon, \varepsilon, \theta^x(y, \gamma))$ en fonction de si $v_1 \neq v_{\ell+1}$ ou $v_1 = v_{\ell+1}$. Dans ces deux cas nous avons alors $\rho_p = \varepsilon$ et par conséquent ρ_p est nécessairement un chemin simple de longueur minimale. Quant à ρ_c il est bien soit un cycle soit le chemin vide puisqu'il est le chemin vide. Si nous avons $\rho_s = \varepsilon$ alors ρ_s est également un chemin simple et $\Theta^x(y, \gamma)$ est un schéma d'itération. Si au contraire nous avons $\rho_s \neq \varepsilon$ alors cela signifie que le sommet initial de $\theta^x(y, \gamma)$ n'est pas son sommet final et donc que $\rho_s = \theta^x(y, \gamma)$ est un chemin simple, par conséquent $\Theta^x(y, \gamma)$ est bien un schéma d'itération.

Considérons à présent le cas où $\rho_c \neq \varepsilon$ alors ρ_c est un cycle dont le sommet initial est le sommet final de ρ_p et est aussi le sommet initial de ρ_s . De plus par construction de $\theta^x(y, \gamma)$, nous avons que ρ_p est de taille i telle que v_i soit la première occurrence du sommet initial de ρ_c . Par conséquent nous avons que ρ_p est bien de taille minimale et donc nous avons que $\Theta^x(y, \gamma)$ est un schéma d'itération. \square

Lemme 4.4. *Si pour tout ρ préfixe de γ , nous avons $0 \leq \phi^x(y, \rho)$ alors il existe $i \in \mathbb{N}$ tel que $\rho_p \rho_c^i \rho_s$ est un chemin vide ou avec les mêmes sommets initiaux et finaux que γ et pour tout ρ' préfixe de $\rho_p \rho_c^i \rho_s$, nous avons $0 \leq \phi^x(y, \rho')$.*

Démonstration. Remarquons que si ρ_p , ρ_c et ρ_s sont tous les trois des chemins vides alors le reste de ce lemme est trivialement vrai. Dans la suite de cette preuve nous considérerons qu'au moins ρ_p , ρ_c ou ρ_s n'est pas le chemin vide.

Puisque la transformation θ ne supprime que des cycles de ρ nous avons que pour tout $\ell \in \mathbb{N}$, $\gamma' = \rho_p \rho_c^\ell \rho_s$ a les mêmes sommets initiaux et finaux que γ .

Supposons que pour tout préfixe ρ de γ , nous ayons $0 \leq \phi^x(y, \rho)$, alors pour tout préfixe ρ'_p de ρ_p nous avons que $0 \leq \phi^x(y, \rho'_p)$ puisque la transformation θ ne commence par retirer que les cycles qui diminuent l'énergie. Nous avons également $\phi^x(y, \gamma_p) \leq \phi^x(y, \rho'_p)$ avec γ_p le préfixe de γ dont le sommet final est la première occurrence du sommet initial de ρ_c . De plus, comme ρ_c augmente strictement l'énergie nous avons que pour tout $M \in \mathbb{N}$, il existe i_M tel que $M \leq \phi^x(y, \rho_p \rho_c^{i_M})$. En particulier nous pouvons choisir i tel que pour tout ρ'_s préfixe de ρ_s nous ayons $0 \leq \phi^x(y, \rho_p \rho_c^i \rho'_s)$. Par conséquent si pour tout ρ préfixe de γ , nous avons $0 \leq \phi^x(y, \rho)$ alors il existe $i \in \mathbb{N}$ tel que pour tout ρ' préfixe de $\rho_p \rho_c^i \rho_s$, nous ayons $0 \leq \phi^x(y, \rho')$. \square

Lemme 4.5. *Nous avons $\min(\pi(\gamma)) \geq \min(\pi(\Theta^x(y, \gamma)))$.*

Démonstration. Puisque la transformation Θ ne fait que retirer des arcs et n'en ajoute aucun, tous les sommets présents dans $\Theta^x(y, \gamma)$ sont présents dans γ et nous avons que $\min(\pi(\gamma)) \geq \min(\pi(\Theta^x(y, \gamma)))$. □

4.3.3 Transformation d'un chemin en schéma multiple d'itération

De la même manière qu'un chemin croissant peut être simplifié en un chemin de schéma d'itération avec les mêmes sommets initiaux et finaux grâce à la transformation Θ , tout chemin peut être (moins) simplifié en un chemin simple pour les arcs constants ayant les mêmes sommets initiaux et finaux avec la transformation κ . Nous allons ensuite définir la notion de schéma multiple d'itération, structure analogue au schéma d'itération prenant en compte les arcs constants, ainsi que la transformation ψ qui simplifie un chemin quelconque en un chemin de schéma multiple d'itération, à l'instar de la transformation θ qui permet de simplifier les chemins croissants.

Avant de définir la transformation κ qui va chercher à simplifier les chemins contenant des arcs constants, nous avons besoin de définir les facteurs croissants maximaux d'un chemin qui représentent les chemins croissants compris entre chaque arc constant de ce chemin.

Définition 4.6 (Facteur croissant maximal). *Soit ε le chemin vide. Soit $\rho = e_1 \dots e_n$ un chemin fini. Soient $(e_{i_j})_1^k$ ses arcs constants par ordre d'apparition. Soient $\rho_1 = e_1 \dots e_{i_1-1}$, $\rho_{k+1} = e_{i_k+1} \dots e_n$ et pour $j \in \llbracket 2, k \rrbracket$ soient $\rho_j = e_{i_{j-1}+1} \dots e_{i_j-1}$, on appelle facteurs croissants maximaux de ρ les ρ_j .*

Commençons par décrire le fonctionnement de la transformation κ avec un exemple avant de la définir formellement. Nous considérerons dans cet exemple le graphe G_κ^x représenté dans la figure 4.6.

Nous allons étudier la transformation du chemin $\rho_\kappa = e_1 e_2 e_3 e_2 e_4 e_5 e_6 e_7 e_5 e_8 e_9$, de longueur 11. Ce chemin a pour sommet initial s et sommet terminal t et est représenté dans la figure 4.7. L'énergie des arcs n'a pas été représentée car la transformation κ ne considère que le fait qu'un arc soit constant ou non. Afin d'expliciter les arcs constants nous les avons représentés en rouge sur les figures. Pour la suite de cet exemple nous appellerons les arcs de ρ_κ :

- e_{i_1} la première occurrence de l'arc constant e_2 ,

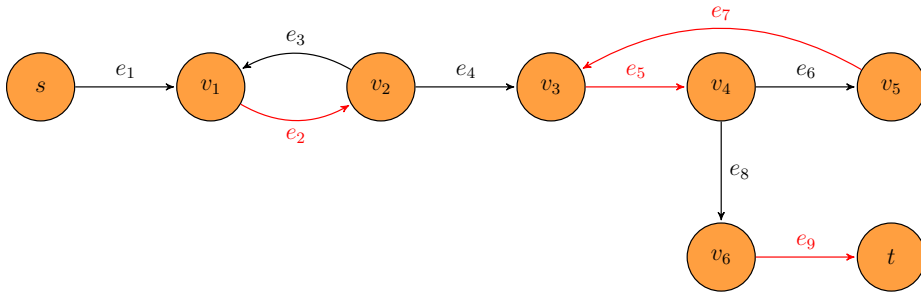


FIGURE 4.6 – Un graphe G_{κ}^x .

- e_{i_2} la seconde occurrence de l'arc constant e_2 ,
- e_{i_3} la première occurrence de l'arc constant e_5 ,
- e_{i_4} l'arc constant e_7 ,
- e_{i_5} la seconde occurrence de l'arc constant e_5 ,
- e_{i_6} l'arc constant e_9 .

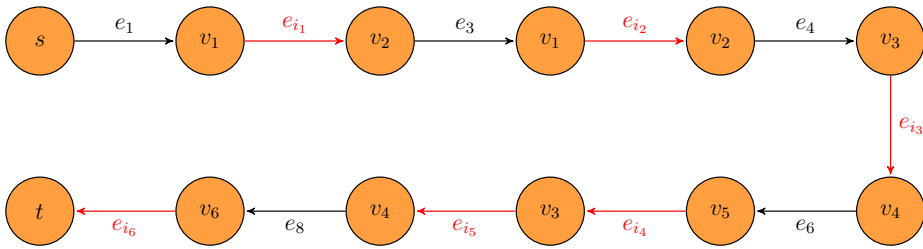


FIGURE 4.7 – Le chemin ρ_{κ} .

Le chemin ρ_{κ} a sept facteurs croissants maximaux que nous noterons :

- $f_1 = e_1$,
- $f_2 = e_3$,
- $f_3 = e_4$,
- $f_4 = e_6$,
- $f_5 = \varepsilon$,
- $f_6 = e_8$,
- $f_7 = \varepsilon$.

La transformation κ a pour objectif de rendre simple pour les arcs constants un chemin quelconque, pour cela nous allons parcourir ρ_{κ} en commençant par son sommet initial et nous supprimerons les cycles dont le sommet initial est un arc constant. Ainsi nous allons supprimer les arcs constants présents plusieurs fois ce qui rendra le chemin simple pour les arcs constants.

Le premier cycle identifié est $e_3e_{i_2}$, la transformation κ supprime le cycle de ρ_κ ce qui en retire le facteur f_2 . Nous avons alors le chemin ρ'_κ représenté sur la figure 4.8.

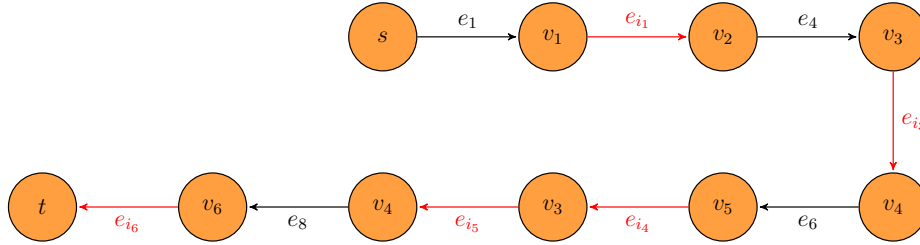


FIGURE 4.8 – Le chemin ρ'_κ .

Comme nous n'avons pas terminé de parcourir ρ'_κ suite à la suppression de ce premier cycle, nous continuons à énumérer les arcs de ρ'_κ . Nous identifions un second cycle qui est $e_6e_{i_4}e_{i_5}$ qui est alors retiré de ρ_κ , ce qui supprime alors les facteurs f_4 et f_5 du chemin. Une fois que nous avons considéré tous les arcs de ρ_κ , le procédé se termine et nous obtenons alors $\kappa(\rho_\kappa)$, représenté sur la figure 4.9.

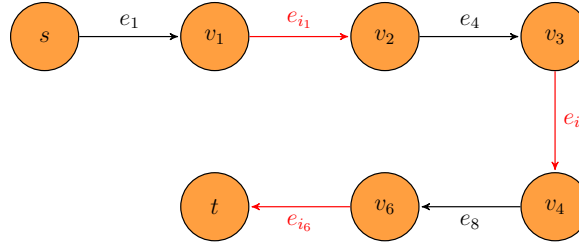


FIGURE 4.9 – Le chemin $\kappa(\rho_\kappa)$.

Nous pouvons constater que dans le chemin ρ_κ seuls les arcs constants e_2 et e_5 sont présents plusieurs fois. La transformation ρ supprime cependant également l'arc constant e_7 . Ainsi, bien que seuls deux doublons soient présents, la transformation κ a retiré trois facteurs croissants maximaux du chemin.

Formalisons à présent la transformation κ que nous venons de présenter sur l'exemple de ρ_κ avec la définition suivante.

Définition 4.7 (Transformation κ). *Soit γ un chemin fini quelconque. Soient $e_{i_1} \dots e_{i_k}$ ses arcs constants par ordre d'apparition et soient $\gamma_1 \dots \gamma_{k+1}$ ses facteurs croissants maximaux.*

Pour $a \in \llbracket 1, k \rrbracket$ nous appelons ρ_a le chemin $\gamma_a e_{i_a}$ qui est la concaténation du a -ième facteur croissant maximal de γ et du a -ième arc constant de γ . De plus, $\rho_{k+1} = \gamma_{k+1}$.

Soit $\kappa(\rho_1) = \rho_1$ et pour $j \in \llbracket 2, k \rrbracket$ définissons $\kappa(\rho_j)$ tel que $\kappa(\rho_j) = \varepsilon$ s'il y a $l < j$ et $l' \geq j$ tels que $e_{i_l} = e_{i_{l'}}$ et $\kappa(\rho_j) = \rho_j$ sinon. Enfin, $\kappa(\rho_{k+1})$ est toujours ρ_{k+1} .

Nous définissons $\kappa(\gamma) = \kappa(\rho_1) \dots \kappa(\rho_{k+1})$ le chemin simple pour les arcs constants issus de γ .

Nous pouvons remarquer que, pour un facteur croissant maximal donné, soit la transformation κ laisse le facteur inchangé, soit elle le supprime complètement. De plus si γ est un chemin croissant alors on a $\kappa(\gamma) = \gamma$ puisqu'il n'y a aucun arc constant dans γ .

Nous prouvons avec les lemmes suivants qu'à partir d'un chemin quelconque γ , la transformation κ permet d'obtenir un chemin $\kappa(\gamma)$ qui est simple pour les arcs constants et a les mêmes sommets initiaux et finaux que γ . De plus si tous les préfixes de γ ont une énergie positive alors ce sera également le cas du chemin construit avec κ (lemme 4.6). Enfin nous prouvons avec le lemme 4.7 que le chemin $\kappa(\gamma)$ a une parité minimale supérieure ou égale à la parité minimale de γ .

Lemme 4.6. *Considérons $x \in \mathbb{Z}^n$, $y \in \mathbb{N}$, et un chemin γ , nous avons $\kappa(\gamma)$ un chemin simple pour les arcs constants qui a les mêmes sommets initiaux et finaux que γ et est tel que si pour tout ρ_γ préfixe de γ , $0 \leq \phi^x(y, \rho_\gamma)$ alors pour tout ρ_κ préfixe de $\kappa(\gamma)$, $0 \leq \phi^x(y, \rho_\kappa)$.*

Démonstration. Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur, et soit $\gamma = e_1 \dots e_n$ un chemin de G . Soient les e_{i_1}, \dots, e_{i_k} les arcs constant de γ par ordre d'occurrence. Soient les γ_i , pour $i \in \llbracket 1, k+1 \rrbracket$ les facteurs croissants maximaux de γ , les ρ_i et $\kappa(\gamma) = \kappa(\rho_1) \dots \kappa(\rho_{k+1})$ comme définit précédemment.

Comme par définition $\kappa(\rho_1) = \rho_1$, nous avons que $\kappa(\gamma)$ a le même sommet initial que γ et comme par définition $\kappa(\rho_{k+1}) = \rho_{k+1}$, nous avons que $\kappa(\gamma)$ a le même sommet final que γ .

Supposons que $\kappa(\gamma)$ ne soit pas simple pour les arcs constants, cela signifie qu'il existe j et l tels que $e_{i_j} = e_{i_l}$ et $j \neq l$, tout en ayant $\kappa(\rho_j) \neq \varepsilon$ et $\kappa(\rho_l) \neq \varepsilon$. Supposons sans perte de généralité que $j < l$, nous avons alors par construction que $\kappa(\rho_l) = \varepsilon$, puisque c'est impossible cela signifie que $\kappa(\gamma)$ est simple pour les arcs constants.

Supposons ensuite que pour tout préfixe p_γ de γ , $0 \leq \phi^x(y, p_\gamma)$, alors pour tout préfixe p_ℓ de γ_ℓ on a $0 \leq \phi^x(y, p_\ell)$. En effet soit $p_{\gamma, \ell}$ le préfixe de γ se terminant par p_ℓ , nous avons alors $p_{\gamma, \ell} = \gamma_1 e_{i_1} \dots e_{i_{\ell-1}} p_\ell$. Étant donné que les arcs constants fixent l'énergie à

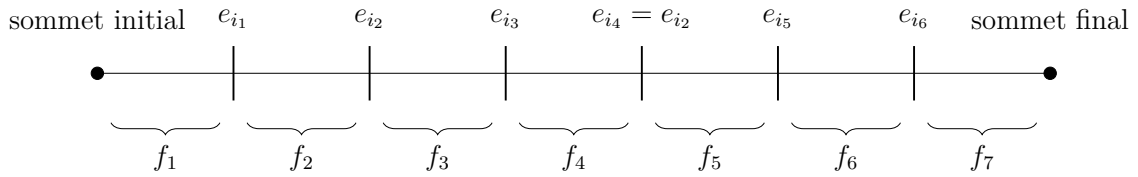
une valeur donnée indépendamment de la valeur précédente de l'énergie ($\mu_{e_{i_\ell}} = 0$), nous avons $\phi^x(y, p_{\gamma, \ell}) = \phi^x(y, p_\ell)$. Or $0 \leq \phi^x(y, p_{\gamma, \ell})$ puisque $p_{\gamma, \ell}$ est un préfixe de γ , nous avons donc bien $0 \leq \phi^x(y, p_\ell)$.

Comme nous avons que soit $\kappa(\rho_i) = \rho_i$, soit $\kappa(\rho_i) = \varepsilon$, alors pour tout préfixe $p_{\kappa, i}$ de $\kappa(\rho_i)$ nous avons $0 \leq \phi^x(y, p_{\kappa, i})$. De plus puisque les arcs constants réinitialisent l'énergie à 0 ($\mu_{e_{i_\ell}} = 0$ pour tout ℓ), pour tout préfixe p_κ de $\kappa(\gamma)$, nous avons $0 \leq \phi^x(y, p_\kappa)$. Par conséquent si pour tout préfixe p_γ de γ , nous avons $0 \leq \phi^x(y, p_\gamma)$ alors pour tout préfixe p_κ de $\kappa(\gamma)$, nous avons $0 \leq \phi^x(y, p_\kappa)$.

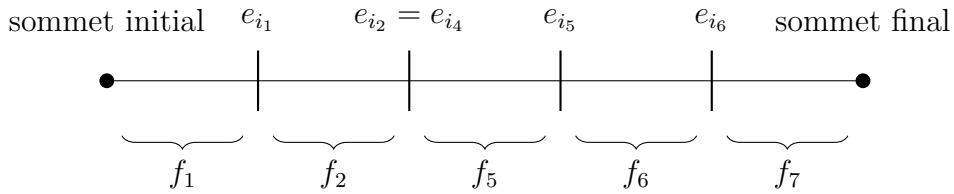
□

Lemme 4.7. Soit γ un chemin, nous avons que $\min(\pi(\gamma)) \geq \min(\pi(\kappa(y, \gamma)))$.

Démonstration. Puisque la transformation κ ne fait que potentiellement retirer des sommets sans en ajouter aucun nous avons que $\min(\pi(\gamma)) \geq \min(\pi(\kappa(y, \gamma)))$. □



(a) - Un chemin ρ



(b) - Le chemin $\kappa(\rho)$

FIGURE 4.10 – Un chemin ρ et le chemin $\kappa(\rho)$.

Dans la figure 4.10 (a) nous avons représenté un chemin ρ comme une ligne horizontale. Nous avons mis en valeur ses arcs constants qui sont les $e_{i_1} \dots e_{i_6}$ ainsi que ses facteurs croissants maximaux, les $f_1 \dots f_7$. Dans la figure 4.10 (b) nous avons représenté $\kappa(\rho)$ pour montrer comme la transformation κ fonctionne et en quoi elle laisse inchangé ou supprime complètement les facteurs croissants maximaux de ρ .

Nous nous intéressons maintenant à la notion de schéma multiple d'itération afin de créer une structure analogue aux schémas d'itération qui prend en compte les arcs constants. Nous définissons aussi la transformation ψ qui permet d'obtenir un schéma multiple d'itération à partir d'un chemin simple pour les arcs constants. Nous prouvons ensuite que nous pouvons créer des chemins ayant des propriétés intéressantes grâce à la transformation ψ .

Définition 4.8 (Chemin de schéma multiple d'itération). *Considérons un chemin $\rho = e_1 \dots e_n$. Soient les e_{i_1}, \dots, e_{i_k} ses arcs constants par ordre d'occurrence. Le chemin ρ est dit "de schéma multiple d'itération" s'il est simple pour les arcs constants et si ses facteurs croissants maximaux sont des chemins de schéma d'itération.*

Définition 4.9 (Schéma multiple d'itération). *Soit $\rho = e_1 \dots e_n$ un chemin de schéma multiple d'itération. Soient les e_{i_1}, \dots, e_{i_k} ses arcs constants par ordre d'occurrence. Soient les $(\rho_{p_j}, \rho_{c_j}, \rho_{s_j})$ pour $j \in \llbracket 1, k+1 \rrbracket$ les schémas d'itérations des facteurs croissants maximaux de ρ . Le $2(k+1)$ -uplet $(\rho_{p_1}, \rho_{c_1}, \rho_{s_1}, e_{i_1}, \dots, e_{i_k}, \rho_{p_{k+1}}, \rho_{c_{k+1}}, \rho_{s_{k+1}})$ est appelé le schéma multiple d'itération du chemin ρ .*

Nous cherchons à construire des schémas multiples d'itération car ils sont en nombre fini comme le montre le lemme 4.8. Cela nous permettra de les énumérer.

Lemme 4.8. *Soit G un jeu de parité à énergie paramétrée, il y a un nombre fini de schémas multiples d'itération.*

Démonstration. Il y a un nombre fini de schéma d'itération d'après le lemme 4.1 et puisque E est fini il y a également un nombre fini d'arcs constants. Un schéma multiple d'itération est une combinaison finie de différents arcs constants ainsi que de schémas d'itération. Comme il n'y a qu'un nombre fini de chacun de ces éléments il n'y a également qu'un nombre fini de schémas multiples d'itération. □

Commençons par décrire le fonctionnement de la transformation ψ avec un exemple avant de la définir formellement. Nous considérerons dans cet exemple le graphe G_ψ^x représenté dans la figure 4.11.

Nous allons étudier la transformation du chemin ρ_ψ , de longueur 11. Ce chemin a pour sommet initial s et sommet terminal t et est représenté dans la figure 4.12. Ce chemin contient un arc constant, e_{i_0} et a deux facteurs croissants maximaux :

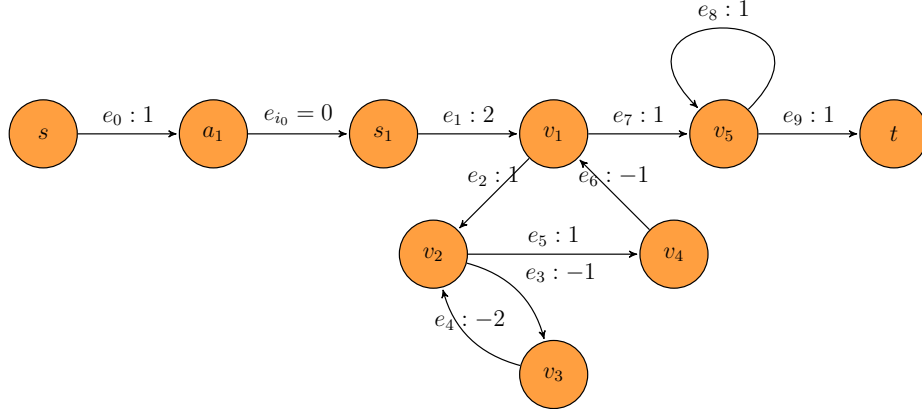


FIGURE 4.11 – Un graphe G_ψ^x .

- $f_1 = e_0$,
- $f_2 = e_1 \dots e_9$.

Nous pouvons remarquer que f_2 est identique au chemin ρ_θ qui nous a servi à illustrer le fonctionnement des transformations θ et Θ . En effet la transformation ψ fait appel à la transformation Θ comme nous le verrons dans cet exemple.

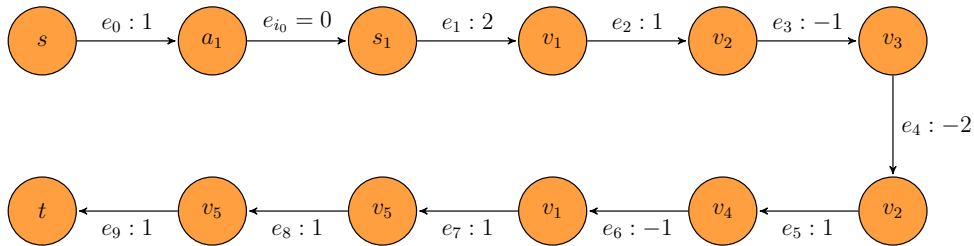


FIGURE 4.12 – Le chemin ρ_ψ .

La transformation ψ cherche à créer un schéma multiple d'itération à partir d'un chemin simple pour les arcs constants. Pour cela nous commençons par identifier les différents facteurs croissants maximaux qui sont f_1 et f_2 .

Nous avons donc une première décomposition de notre chemin qui nous servira de base pour créer $\psi(\rho_\psi) : \rho'_\psi = (f_1, e_{i_0}, f_2)$. Cependant nous souhaitons obtenir un schéma multiple d'itération et pour cela nous avons besoin de remplacer f_1 et f_2 par des schémas d'itération. Pour cela nous utilisons la transformation θ qui va associer des schémas d'itération $\Theta(f_1)$ et $\Theta(f_2)$ aux chemins croissants que sont les facteurs croissants maximaux de ρ_ψ .

La transformation Θ ayant construit les schémas d'itérations associés aux facteurs

croissants maximaux nous pouvons ensuite créer un schéma multiple d'itération avec ψ . Nous obtenons alors $\psi(\rho_\psi) = (\Theta(f_1), e_{i_0}, \Theta(f_2))$, c'est-à-dire $\psi(\rho_\psi) = ((\varepsilon, \varepsilon, 0), e_{i_0}, (e_1, e_2 e_5 e_6, e_7 e_9))$.

Formalisons à présent la transformation que nous venons de décrire sur l'exemple de ρ_ψ avec la définition suivante.

Définition 4.10 (Transformation ψ). *Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Fixons $x \in \mathbb{Z}^n$ et $y \in \mathbb{N}$. Soit $\gamma = e_1 \dots e_n$ un chemin simple pour les arcs constants de G^x . Nous construisons un schéma multiple d'itération $\psi^x(y, \gamma)$ en créant un schéma d'itération pour chaque facteur croissant maximal de γ . Plus spécifiquement, soient e_{i_1}, \dots, e_{i_k} les arcs constants de γ par ordre d'occurrence et les f_i ses facteurs croissants maximaux. Nous décomposons le chemin γ comme $f_1 e_{i_1} f_2 e_{i_2} \dots e_{i_k} f_{k+1}$.*

Enfinement nous avons $\psi^x(y, \gamma) = (\Theta^x(y, f_1), e_{i_1}, \Theta^x(\alpha_{e_{i_1}}(x), f_2), e_{i_2}, \dots, e_{i_k}, \Theta^x(\alpha_{e_{i_k}}(x), f_{k+1}))$.

Nous prouvons avec les lemmes suivants plusieurs propriétés de la transformation ψ . Tout d'abord nous montrons qu'à partir d'un chemin simple pour les arcs constants γ , la transformation ψ permet d'obtenir un schéma multiple d'itération dont les sommets initiaux et finaux sont les mêmes que ceux de γ avec le lemme 4.9. De plus nous montrons avec le lemme 4.10 que si tous les préfixes de γ ont une énergie positive alors il est possible de construire un chemin de schéma multiple d'itération qui vérifie également cette propriété.

Lemme 4.9. *Considérons $x \in \mathbb{Z}^n$, $y \in \mathbb{N}$, et un chemin γ simple pour les arcs constants. Nous avons $\psi(\gamma)$ un schéma multiple d'itération tel que les chemins de schéma multiple d'itération associés à $\psi(\gamma)$ aient les mêmes sommets initiaux et finaux que γ .*

Démonstration. Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Soient $x \in \mathbb{Z}^n$ et $\gamma = e_1 \dots e_n$ un chemin. Soient e_{i_1}, \dots, e_{i_k} les arcs constants de γ par ordre d'occurrence. Soient γ_j , pour $j \in \llbracket 1, k+1 \rrbracket$ les facteurs croissants maximaux de γ et $\psi^x(y, \gamma) = (\Theta^x(y, \gamma_1), e_{i_1}, \Theta^x(\alpha_{e_{i_1}}(x), \gamma_2), e_{i_2}, \dots, e_{i_k}, \Theta^x(\alpha_{e_{i_k}}(x), \gamma_{k+1}))$ comme défini précédemment.

Pour tout $j \in \llbracket 1, k+1 \rrbracket$, soit $z_j = y$ si $j = 1$ et $z_j = \alpha_{e_{i_{j-1}}}(x)$ sinon. D'après le lemme 4.3, nous avons $\Theta^x(z_j, u_j)$ un schéma d'itération avec les mêmes sommets initiaux et finaux que u_j . Par conséquent nous avons $\psi(\gamma)$ un schéma multiple d'itération tel que

les chemins de schéma multiple d'itération associés aient les mêmes sommets initiaux et finaux que γ . \square

Lemme 4.10. *Considérons $x \in \mathbb{Z}^n$, $y \in \mathbb{N}$, et un chemin γ simple pour les arcs constants. Si pour tout préfixe ρ_γ de γ , nous avons $0 \leq \phi^x(y, \rho_\gamma)$ alors il existe p un chemin de schéma multiple d'itération associé à $\psi(\gamma)$ tel que pour tout préfixe ρ_p de p , nous ayons $0 \leq \phi^x(y, \rho_p)$.*

Démonstration. Soit $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur. Soient $x \in \mathbb{Z}^n$ et $\gamma = e_1 \dots e_n$ un chemin. Soient e_{i_1}, \dots, e_{i_k} les arcs constants de γ par ordre d'occurrence. Soient γ_j , pour $j \in \llbracket 1, k+1 \rrbracket$ les facteurs croissants maximaux de γ et $\psi^x(y, \gamma) = (\Theta^x(y, \gamma_1), e_{i_1}, \Theta^x(\alpha_{e_{i_1}}(x), \gamma_2), e_{i_2}, \dots, e_{i_k}, \Theta^x(\alpha_{e_{i_k}}(x), \gamma_{k+1}))$ comme défini précédemment.

Nous avons $\psi(\gamma)$ un schéma multiple d'itération d'après le lemme précédent.

Supposons que pour tout préfixe ρ_γ de γ , nous ayons $0 \leq \phi^x(y, \rho_\gamma)$, alors pour tout préfixe ρ_ℓ de γ_ℓ nous avons $0 \leq \phi^x(y, \rho_\ell)$ car $\phi^x(y, \rho_\ell) = \phi^x(y, \gamma_1 e_{i_1} \dots e_{i_{\ell-1}} \rho_\ell)$ puisque $e_{i_{\ell-1}}$ est un arc constant et que le chemin $\gamma_1 e_{i_1} \dots e_{i_{\ell-1}} \rho_\ell$ est un préfixe de γ .

Pour $j \in \llbracket 2, k+1 \rrbracket$, soit $z_j = \alpha_{e_{i_j}}(x)$ et soit $z_1 = y$. Comme pour tout j , $\Theta^x(z_j, \gamma_j)$ est un schéma d'itération, d'après le lemme 4.4 il existe un chemin p_j ayant les mêmes sommets initiaux et finaux que γ_j et tel que pour tout préfixe p'_j de p_j , nous avons $0 \leq \phi^x(z_j, p'_j)$. Soit $p = p_1 e_{i_1} \dots e_{i_{k+1}} p_{k+1}$, p est un chemin de schéma multiple d'itération associé à $\psi(\gamma)$ et pour tout préfixe ρ_p de p , nous avons $0 \leq \phi^x(y, \rho_p)$. Par conséquent si pour tout préfixe ρ_γ de γ , nous avons $0 \leq \phi^x(y, \rho_\gamma)$ alors il existe p un chemin de schéma multiple d'itération associé à $\psi(\gamma)$ tel que pour tout préfixe ρ_p de p , nous ayons $0 \leq \phi^x(y, \rho_p)$. \square

4.3.4 Chemins gagnants

Maintenant que nous avons défini toutes ces transformations nous nous intéressons ici aux chemins gagnants. Nous commençons par prouver que s'il existe au moins un chemin gagnant, alors il existe un chemin gagnant qui est composé de deux schémas multiples d'itération. Nous montrons ensuite les propriétés que doivent avoir ces schémas multiples d'itération ce qui nous donne une méthode pour chercher un chemin gagnant qui consiste à énumérer les schémas multiples d'itération en testant si certains vérifient ces propriétés.

Le lemme suivant montre l'existence d'un chemin gagnant ayant une certaine forme dès l'instant où il existe au moins un chemin gagnant.

De manière informelle, nous montrons que si nous avons un chemin gagnant ρ alors nous pouvons construire un chemin qui a une structure donnée ce qui nous assure que nous n'avons qu'à tester certaines propriétés du graphe pour dire s'il existe un chemin gagnant ou non.

Pour cela nous commençons par chercher un sommet responsable de la parité minimale qui apparaît infiniment souvent, nous l'avons nommé s_1 dans la figure 4.13 qui représente un chemin gagnant quelconque.

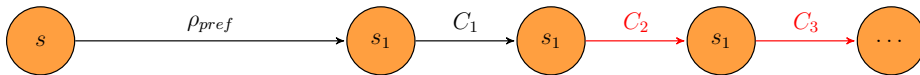


FIGURE 4.13 – Un chemin gagnant ρ .

Puisque nous avons un chemin infini dont l'énergie est toujours positive, il y a au moins un cycle ne diminuant pas l'énergie qui commence (et se finit) en s_1 . Disons que le cycle C_{i_0} ayant pour sommet initial s_1 vérifie cette propriété. Nous pouvons alors simplifier le chemin ρ en ρ' comme illustré sur la figure 4.14.

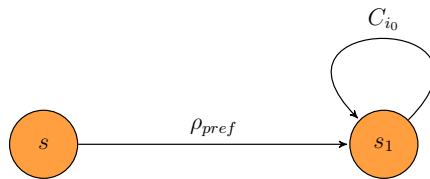
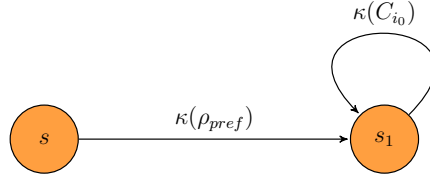


FIGURE 4.14 – Un chemin gagnant ρ' .

Nous pouvons alors simplifier le cycle C_{i_0} avec les transformations κ et ψ en retirant les parties superflues comme les cycles qui diminuent l'énergie ou les cycles commençant par un arc constant comme illustré sur la figure 4.15. Cela fait nous avons v , la partie du chemin que nous allons répéter indéfiniment.

Nous avons également besoin d'atteindre le point de départ de v donc nous prenons le préfixe ρ qui se termine à la première apparition de s_1 . Nous simplifions également ρ_{pref} avec les transformations κ et ψ pour obtenir u , la partie du chemin qui nous permet d'atteindre v . Enfin, nous construisons le chemin gagnant $\rho_g = uv^\omega$ tel que u et v soient des chemins de schéma multiple d'itération, ce qui nous garantit que lorsque nous cherchons un chemin gagnant il nous suffit de vérifier les chemins de schéma multiple d'itération pour savoir s'il en existe un ou non.

FIGURE 4.15 – Un chemin gagnant ρ_g .

Lemme 4.11. Soient $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur et $x \in \mathbb{Z}^n$. S'il existe un chemin gagnant dans G^x , alors il existe également un chemin gagnant uv^ω tel que u et v soient des chemins de schéma multiple d'itération.

Démonstration. Soit $\rho = e_1 e_2 \dots$ un tel chemin gagnant avec pour tout $i \geq 1$, $e_i = (v_i, v_{i+1})$. Soit $I := \inf(\pi(\rho))$ l'ensemble des priorités qui apparaissent infiniment souvent dans ρ . Soit ν un sommet apparaissant infiniment souvent dans ρ et tel que sa parité soit minimale parmi celles apparaissant infiniment souvent dans ρ , c'est-à-dire $\pi_\nu = \min(I)$.

Puisque le chemin est un chemin gagnant, l'énergie du chemin est toujours positive ou nulle. Par conséquent il y a au moins deux occurrences consécutives d'arcs dirigés vers ν dans ρ , aux positions $i < j$, telles que $\phi^x(0, e_1 \dots e_{i-1}) \leq \phi^x(0, e_1 \dots e_{j-1})$. Soient $u := \psi^x(0, \kappa(e_1 \dots e_{i-1}))$, et $y := \phi^x(0, e_1 \dots e_{i-1})$, et soit $v := \psi^x(y, \kappa(e_i \dots e_{j-1}))$. D'après le lemme 4.6, nous avons que $\kappa(e_1 \dots e_{i-1})$ et $\kappa(e_i \dots e_{j-1})$ sont simples pour les arcs constants et donc d'après le lemme 4.9 nous avons que u et v sont des chemins de schéma multiple d'itération. Par construction nous avons que $\inf(\pi(uv^\omega)) \subseteq I$. De plus comme le sommet initial de v est ν nous avons que $\pi_\nu \in \inf(\pi(uv^\omega))$ et que $\min(\inf(\pi(uv^\omega))) = \min(I)$. Puisque ρ est un chemin gagnant nous savons donc que $\min(\inf(\pi(uv^\omega)))$ est pair.

D'après le lemme 4.6 nous avons que pour tout préfixe p de $\kappa(e_1 \dots e_{i-1})$ (resp. $\kappa(e_i \dots e_{j-1})$), il existe un préfixe p' de $e_1 \dots e_{i-1}$ (resp. $e_i \dots e_{j-1}$) tel que $\phi^x(0, p) = \phi^x(0, p')$ (resp. $\phi^x(y, \kappa(e_i \dots e_{j-1})) = \phi^x(y, e_i \dots e_{j-1})$) et comme ρ est un chemin gagnant l'énergie de ses préfixes est toujours positive. Par conséquent l'énergie des préfixes de u et de v est positive et nous avons que uv^ω est un chemin gagnant tel que u et v soient des chemins de schéma multiple d'itération. \square

Nous définissons à présent la notion de sureté pour une valeur donnée qui indique si un schéma d'itération peut permettre de créer des chemins de schéma d'itération dont l'énergie reste positive si l'énergie de départ prend la valeur considérée. Nous étendons

également cette notion aux schémas multiples d'itération qui sont sûrs lorsque tous les schémas d'itération les composant le sont pour certaines valeurs.

Définition 4.11 (Sureté). Soient $x \in \mathbb{Z}^n$ et $y \in \mathbb{N}$. Soit $f = (\rho_p, \rho_c, \rho_s)$ un schéma d'itération. Si ρ_c est le chemin vide, nous disons que f est sûr pour y si $0 \leq \phi^x(y, w)$ pour tout préfixe w de $\rho_p \rho_s$. Si ρ_c est non-vide nous disons que f est sûr pour y si $0 \leq \phi^x(y, w)$ pour tout préfixe w de $\rho_p \rho_c$, et si $\phi^x(y, \rho_p) < \phi^x(y, \rho_p \rho_c)$. Considérons maintenant un schéma multiple d'itération $f = ((\rho_{p_1}, \rho_{c_1}, \rho_{s_1}), e_{i_1}, \dots, e_{i_k}, (\rho_{p_{k+1}}, \rho_{c_{k+1}}, \rho_{s_{k+1}}))$. Nous disons que f est sûr pour y si $(\rho_{p_1}, \rho_{c_1}, \rho_{s_1})$ est sûr pour y , et chaque $(\rho_{p_{j+1}}, \rho_{c_{j+1}}, \rho_{s_{j+1}})$ est sûr pour $\alpha_{e_{i_j}}(x)$.

Théorème 4.1. Soient $G = (V_1, \emptyset, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée à un joueur et $x \in \mathbb{Z}^n$. Les affirmations suivantes sont équivalentes.

1. Il existe un chemin gagnant dans G^x qui a pour sommet initial s .
2. Il existe un schéma multiple d'itération $f = ((\rho_{p_1}, \rho_{c_1}, \rho_{s_1}), e_{i_1}, \dots, e_{i_k}, (\rho_{p_{k+1}}, \rho_{c_{k+1}}, \rho_{s_{k+1}}))$ avec s le sommet initial de ρ_{p_1} et un schéma multiple d'itération $f' = ((\rho'_{p_1}, \rho'_{c_1}, \rho'_{s_1}), e_{i'_1}, \dots, e_{i'_{k'}}, (\rho'_{p_{k'+1}}, \rho'_{c_{k'+1}}, \rho'_{s_{k'+1}}))$ commençant et terminant à la fin de $\rho_{s_{k+1}}$, et vérifiant les énoncés suivants :
 - f est sûr pour 0.
 - Si $\rho_{c_{k+1}}$ est le chemin vide et $k = 0$, alors f' est sûr pour $\phi^x(0, \rho_{p_1} \rho_{s_1})$.
 - Si $\rho_{c_{k+1}}$ est le chemin vide et $k \neq 0$, alors f' est sûr pour $\phi^x(\alpha_{e_{i_k}}(x), \rho_{p_{k+1}} \rho_{s_{k+1}})$.
 - Si $\rho_{c_{k+1}}$ est non-vide, il existe $y \in \mathbb{N}$ tel que f' est sûr y .
 - La parité minimale apparaissant dans f' est paire.

Démonstration. Pour le sens $1 \Rightarrow 2$, s'il existe un chemin gagnant ayant pour sommet initial s dans G^x , d'après le lemme 4.11 nous avons u et v des chemins de schéma multiple d'itération tels que uv^ω soit un chemin gagnant de G^x avec pour sommet initial s . Soit f le schéma multiple d'itération de u et f' le schéma multiple d'itération de v . Soient $f = (\rho_{p_1}, \rho_{c_1}, \rho_{s_1}, e_{i_1}, \dots, e_{i_k}, \rho_{p_{k+1}}, \rho_{c_{k+1}}, \rho_{s_{k+1}})$ et $f' = (\rho'_{p_1}, \rho'_{c_1}, \rho'_{s_1}, e_{i'_1}, \dots, e_{i'_{k'}}, \rho'_{p_{k'+1}}, \rho'_{c_{k'+1}}, \rho'_{s_{k'+1}})$.

Puisque uv^ω est un chemin gagnant, f est sûr pour 0.

Si $\rho_{c_{k+1}}$ est le chemin vide et $k = 0$, alors $u = \rho_{p_1} \rho_{s_1}$ et comme uv^ω est un chemin gagnant, f' est sûr pour $\phi^x(0, \rho_{p_1} \rho_{s_1})$.

Si $\rho_{c_{k+1}}$ est le chemin vide et $k \neq 0$, alors le dernier chemin de schéma d'itération de u est $\rho_{p_{k+1}} \rho_{v_{k+1}}$ et f' est sûr pour $\phi^x(\alpha_{e_{i_k}}(x), \rho_{p_{k+1}} \rho_{v_{k+1}})$ puisque uv^ω est un chemin gagnant.

Pour les mêmes raisons si $\rho_{c_{k+1}}$ est le chemin non vide, alors f' est sûr pour $y = \phi^x(0, u)$.

De plus comme uv^ω est un chemin gagnant, nous avons que $\min \circ \inf \circ \pi(uv^\omega)$ est pair. Par conséquent la priorité minimale apparaissant dans v est paire et donc la priorité minimale apparaissant dans f' est paire.

Dans tous les cas l'implication est vraie donc nous avons bien que $1 \Rightarrow 2$.

Pour le sens $2 \Rightarrow 1$, soient $f = (\rho_{p_1}, \rho_{c_1}, \rho_{s_1}, e_{i_1}, \dots, e_{i_k}, \rho_{p_{k+1}}, \rho_{c_{k+1}}, \rho_{s_{k+1}})$ et $f' = (\rho'_{p_1}, \rho'_{c_1}, \rho'_{s_1}, e_{i'_1}, \dots, e_{i'_{k'}}, \rho'_{p_{k'+1}}, \rho'_{c_{k'+1}}, \rho'_{s_{k'+1}})$ des schémas multiples d'itération tels que f soit sûr pour 0, que la priorité minimale apparaissant dans f' soit paire, que s soit le sommet initial de ρ_{p_1} et que le sommet initial de $\rho'_{p_{k'+1}}$ soit le sommet final de $\rho_{s_{k+1}}$ ainsi que celui de $\rho'_{s_{k'+1}}$.

Commençons par considérer le cas où $\rho_{c_{k+1}}$ est le chemin vide, avec $k = 0$, et où f' est sûr pour $\phi^x(0, \rho_{p_1}\rho_{s_1})$. Soient $u = \rho_{p_1}\rho_{s_1}$ et $v = \rho'_{p_1}\rho'_{c_1}{}^{j'_1}\rho'_{s_1}e_{i'_1} \dots e_{i'_{k'}}\rho'_{p_{k'+1}}\rho'_{c_{k'+1}}{}^{j'_{k'+1}}\rho'_{s_{k'+1}}$ avec pour tout $\ell \in \llbracket 1, k+1 \rrbracket$, $j'_\ell \in \mathbb{N}$ minimal tel que $\phi^x(y, \rho'_{p_\ell}) < \phi^x(y, \rho'_{p_\ell}\rho'_{c_\ell}{}^{j'_\ell}\rho'_{s_\ell})$. Nous avons alors uv^ω un chemin tel que l'énergie de tous ses préfixes soit positive puisque f est sûr pour 0 et que f' est sûr pour $\phi^x(0, u)$. De plus, comme la priorité minimale apparaissant dans f' est paire, la parité minimale apparaissant dans v est également paire. Par conséquent, uv^ω est un chemin gagnant de G^x dont le sommet initial est s .

Considérons maintenant le cas où $\rho_{c_{k+1}}$ est le chemin vide, avec $k \neq 0$ et où f' est sûr pour $\phi^x(\alpha_{e_{i_k}}(x), \rho_{p_{k+1}}\rho_{s_{k+1}})$. Soit $u = \rho_{p_1}\rho_{c_1}{}^{j_1}\rho_{s_1}e_{i_1} \dots e_{i_k}\rho_{p_{k+1}}\rho_{s_{k+1}}$ avec pour tout $\ell \in \llbracket 1, k+1 \rrbracket$, $j'_\ell \in \mathbb{N}$, $j_\ell \in \mathbb{N}$ minimal tel que l'énergie de tous les préfixes de $\rho_{p_\ell}\rho_{c_\ell}{}^{j_\ell}\rho_{s_\ell}$ soit positive. Un tel j_ℓ existe car pour tout $y \in \mathbb{Z}$, nous avons $\phi^x(y', \rho_{p_\ell}) < \phi^x(y', \rho_{p_\ell}\rho_{c_\ell}{}^{j_\ell}\rho_{s_\ell})$. Soit $v = \rho'_{p_1}\rho'_{c_1}{}^{j'_1}\rho'_{s_1}e_{i'_1} \dots e_{i'_{k'}}\rho'_{p_{k'+1}}\rho'_{c_{k'+1}}{}^{j'_{k'+1}}\rho'_{s_{k'+1}}$ avec $j'_\ell \in \mathbb{N}$ minimal tel que $\phi^x(y, \rho'_{p_\ell}) < \phi^x(y, \rho'_{p_\ell}\rho'_{c_\ell}{}^{j'_\ell}\rho'_{s_\ell})$. Nous avons uv^ω un chemin tel que l'énergie de tous ses préfixes soit positive puisque f est sûr pour 0. Nous avons également que f' est sûr pour $\phi^x(0, u)$ puisque $\phi^x(0, u) = \phi^x(\alpha_{e_{i_k}}(x), \rho_{p_{k+1}}\rho_{s_{k+1}})$. De plus comme la priorité minimale apparaissant dans f' est paire, nous avons que la priorité minimale apparaissant dans v est également paire. Par conséquent uv^ω est un chemin gagnant de G^x dont le sommet initial est s .

Enfin, considérons le cas où $\rho_{c_{k+1}}$ est non vide et où f' est sûr pour un $y \in \mathbb{N}$. Soit $j_\ell \in \mathbb{N}$ pour $\ell \in \llbracket 1, k \rrbracket$, minimal tel que l'énergie de tous les préfixes de $\rho_{p_\ell}\rho_{c_\ell}{}^{j_\ell}\rho_{s_\ell}$ soit positive, un tel j_ℓ existe car pour tout $y' \in \mathbb{Z}$, nous avons $\phi^x(y', \rho_{p_\ell}) < \phi^x(y', \rho_{p_\ell}\rho_{c_\ell}{}^{j_\ell}\rho_{s_\ell})$. Soit $j_{k+1} \in \mathbb{N}$ minimal tel que l'énergie de tous les préfixes de $\rho_{p_{k+1}}\rho_{c_{k+1}}{}^{j_{k+1}}\rho_{s_{k+1}}$ soit positive et $\phi^x(0, \rho_{p_{k+1}}\rho_{c_{k+1}}{}^{j_{k+1}}\rho_{s_{k+1}}) \geq y$, un tel j_{k+1} existe car pour tout $y' \in \mathbb{Z}$, nous avons $\phi^x(y', \rho_{p_{k+1}}) < \phi^x(y', \rho_{p_{k+1}}\rho_{c_{k+1}}{}^{j_{k+1}}\rho_{s_{k+1}})$. Soit $u = \rho_{p_1}\rho_{c_1}{}^{j_1}\rho_{s_1}e_{i_1} \dots e_{i_k}\rho_{p_{k+1}}\rho_{c_{k+1}}{}^{j_{k+1}}\rho_{s_{k+1}}$. Soit $v = \rho'_{p_1}\rho'_{c_1}{}^{j'_1}\rho'_{s_1}e_{i'_1} \dots e_{i'_{k'}}\rho'_{p_{k'+1}}\rho'_{c_{k'+1}}{}^{j'_{k'+1}}\rho'_{s_{k'+1}}$ avec $j'_\ell \in \mathbb{N}$ minimal tel que $\phi^x(y, \rho'_{p_\ell}) <$

$\phi^x(y, \rho'_{p_\ell} \rho'_{c_\ell} \rho'_{s_\ell})$. Nous avons uv^ω un chemin tel que l'énergie de tous ses préfixes soit positive puisque f est sûr pour 0 et f' est sûr pour $\phi^x(0, u)$ puisque $\phi^x(0, u) = \phi^x(\alpha_{e_{i_k}}(x), \rho_{p_{k+1}} \rho_{c_{k+1}} \rho_{s_{k+1}}) \geq y$. Comme la priorité minimale apparaissant dans f' est paire, la priorité minimale apparaissant dans v est également paire et par conséquent uv^ω est un chemin gagnant de G^x dont le sommet initial est s .

Dans tous les cas il y a un chemin gagnant dans G^x dont le sommet initial est s et donc l'implication $2 \Rightarrow 1$ est vraie.

Finalement comme nous avons $1 \Rightarrow 2$ et $2 \Rightarrow 1$, le théorème est vrai. \square

4.3.5 Complexité

Nous nous concentrons maintenant sur la décidabilité et la complexité du problème des jeux de parité à énergie paramétrée dans le cas à un joueur.

Décidabilité D'après le lemme 4.8 nous pouvons énumérer les schémas multiples d'itération en temps fini et nous pouvons vérifier en temps fini les propriétés listées dans le théorème 4.1 sur chaque schéma multiple d'itération en temps fini. Le problème consistant à trouver s'il existe $x \in \mathbb{Z}^n$ tel qu'il existe un chemin gagnant dans un jeu de parité à énergie paramétrée avec un seul joueur, est décidable.

Corollaire 4.1. *Considérons le problème suivant.*

- *Entrée : un jeu de parité à énergie paramétrée à un joueur.*
- *Sortie : l'existence de $x \in \mathbb{Z}^n$ tel qu'il existe un chemin gagnant dans G^x dont le sommet initial est s .*

Ce problème est NP-complet.

Démonstration. Commençons par l'appartenance à NP.

Considérons la machine de Turing non déterministe qui effectue les actions suivantes :

1. Deviner les schémas multiples d'itération f et f' du Théorème 4.1 ainsi que les écrire sur la bande.
2. Ecrire les inégalités sur les paramètres traduisant que f et f' satisfassent les conditions de Théorème 4.1.
3. Décider si ces inégalités sont satisfiables ou non et si la priorité minimale apparaissant dans f' est paire ou non.

D'après le Théorème 4.1, nous avons bien que si la machine termine, elle répond "oui" si et seulement si il existe un chemin gagnant dont le sommet initial est s .

Considérons maintenant les différentes étapes plus en détails.

Tout d'abord, chaque schéma d'itération multiple peut être décrit comme une chaîne de symboles qui est linéaire en la taille de G : nous avons au plus $3|V| + 1$ sommets dans un schéma d'itération, et considérons que nous utilisons deux marques spéciales pour séparer les deux chemins simples et le cycle simple. Le même raisonnement s'applique aux schémas multiples d'itération : nous avons tout au plus $|E|$ arcs constants par conséquent nous avons besoin d'écrire au plus $|E| + 1$ schémas d'itération et $|E|$ arcs pour les séparer afin de décrire un schéma multiple d'itération. Par conséquent, l'étape 1 est faite en temps polynomial par rapport à $|G|$.

Ensuite, décider si c_{k+1} est vide ou non et si k est égal à zéro ou non est également fait en temps linéaire. Calculer les expressions affines correspondant à l'énergie dépensée sur chaque chemin et sur chaque cycle du schéma multiple d'itération est fait en temps polynomial. Par conséquent les contraintes de sûreté du théorème 4.1 sont écrites en temps polynomial. Notons également qu'elles forment une conjonction de contraintes affines.

Trouver la priorité minimale apparaissant dans f' est fait en temps linéaire et décider de la satisfiabilité d'une conjonction de contraintes affines peut être effectué avec la programmation linéaire en nombres entiers (ILP, integer linear programming) [Sch98], en temps polynomial non déterministe.

En prenant tout cela en compte, la machine que nous avons construite termine toujours et opère dans NP.

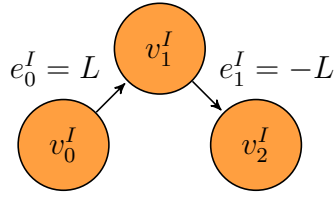
Considérons maintenant l'aspect NP-difficile, nous prouvons que nous pouvons décider la satisfiabilité d'une conjonction de contraintes affines sur des variables entières avec notre problème.

Considérons une inégalité affine $I = (L \geq 0)$, avec L une fonction affine des variables x_1, \dots, x_m . Nous construisons un graphe paramétré correspondant qui contient :

- Trois sommets v_0^I, v_1^I et v_2^I ,
- Un arc e_0^I de v_0^I à v_1^I avec $\mu_{e^I} = 1$ et $\alpha_{e^I} = L$,
- Un arc e_1^I de v_1^I à v_2^I avec $\mu_{e^I} = 1$ et $\alpha_{e^I} = -L$.

La figure 4.16 illustre un graphe pour une inégalité L .

Nous constatons que nous pouvons aller de v_0^I à v_1^I si et seulement si $L \geq 0$. Puis nous pouvons aller à v_2^I avec aucune contrainte supplémentaire et avoir la même énergie que nous avons à v_0^I .

FIGURE 4.16 – Un gadget qui teste si L est positive.

Si nous avons une conjonction I_1, \dots, I_p de telles contraintes affines nous n'avons besoin que de concaténer les graphes correspondants avec des arcs de $v_2^{I_j}$ à $v_0^{I_{j+1}}$ et d'énergie 0 ($\mu = 1$ et α la fonction constante à zéro).

Dans le graphe correspondant que nous avons construit il n'y a qu'un seul chemin pour atteindre le dernier sommet. Nous pouvons atteindre ce dernier sommet si et seulement si la conjonction des contraintes affines est satisfaite. En prenant les x_i comme variables entières, ceci est équivalent à ILP qui est NP-difficile.

□

Nous pouvons remarquer que si les paramètres prennent des valeurs dans \mathbb{Q} et non dans \mathbb{Z} alors les preuves sont toujours valides à l'exception du fait que nous n'avons plus qu'une borne inférieure pour la complexité.

4.4 Cas à deux joueurs

D'après le lemme 3 de l'article [CD12], si le joueur 2 peut gagner dans un jeu de parité à énergie traditionnel, alors il peut gagner avec des stratégies sans mémoire. Ce résultat est toujours valide pour nos jeux légèrement plus généraux, où l'énergie est également multipliée par un facteur non négatif. Nous utiliserons une preuve à la structure similaire à celle du lemme 5 de [CD12]. Avec une preuve analogue nous montrons que les stratégies à mémoire finie suffisent au joueur 1 pour gagner.

4.4.1 Définitions

Nous commençons par poser quelques définitions pour compléter la notion de chemin en ajoutant le fait que les joueurs ne contrôlent qu'une partie des sommets et non la totalité du graphe comme dans le cas à un joueur.

Définition 4.12 (Partie). *Un jeu de parité à énergie paramétrée sur G commençant*

depuis un sommet $v_1 \in V$ est joué en tours comme suit. Si le jeu est dans un sommet appartenant au joueur 1, alors le joueur 1 choisit quel arc sera emprunté parmi l'ensemble des arcs partant du sommet. Sinon le jeu est dans un sommet appartenant au joueur 2, et c'est donc le joueur 2 qui choisit le prochain arc à être emprunté.

Le jeu génère une partie commençant en v_1 qui est un chemin infini $\rho = e_1 e_2 \dots$ tel que $e_i = (v_i, v_{i+1})$ pour tout $i \geq 1$.

Définition 4.13 (Décomposition en cycles). Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soit $\rho = e_1 e_2 \dots$ un chemin infini de G . La décomposition en cycles de ρ est une suite infinie de cycles simples C_1, C_2, \dots obtenue comme suit :

- On débute avec une liste et une décomposition en cycles vides,
- On ajoute successivement à la liste les arcs de ρ en commençant par $e_1, e_2 \dots$,
- Dès que l'on ajoute un arc dont le sommet final est le sommet initial d'un arc qui est déjà dans la liste alors on vient de former un cycle,
- On supprime le cycle de la liste et on l'ajoute à la décomposition en cycle.

Nous pouvons remarquer que la décomposition en cycle contient exactement tous les cycles simples présents dans ρ et que le contenu de la liste est toujours un chemin simple de longueur maximale $|V|$.

Notre modèle contient maintenant deux joueurs et la notion de chemin ne suffit plus à exprimer les possibilités d'un joueur. Nous définissons les stratégies sur un jeu et quelques types spécifiques de stratégies qui nous seront utiles pour résoudre notre problème.

Définition 4.14 (Stratégie). Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soit $P_1 = \{e_1 \dots e_n, v_{n+1} \in V_1\}$ l'ensemble des chemins finis dont le sommet final appartient à V_1 . Une stratégie pour le joueur 1 est une fonction $\sigma : P_1 \rightarrow E$ telle que $\sigma(\rho)$ parte du sommet final de ρ , pour tout $\rho \in P_1$.

Une exécution de σ est une partie $e_1 e_2 \dots$ telle que $\sigma(e_1 \dots e_i) = e_{i+1}$ pour tout $i \geq 0$ tel que $e_1 \dots e_i \in P_1$.

Nous définissons les stratégies pour le joueur 2 de manière analogue.

Définition 4.15 (Stratégie sans mémoire). Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Une stratégie σ_0 est dite sans mémoire si elle peut être représentée par une fonction $f : V_1 \rightarrow E$.

En d'autres termes cela signifie que l'arc choisi par une stratégie sans mémoire depuis un sommet $v \in V_1$ est toujours le même et ne dépend pas du chemin qui a amené à v .

Définition 4.16 (Stratégie à mémoire finie). Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Une stratégie est dite à mémoire finie si elle peut être représentée par un 4-uplet (M, m_0, v_u, v_n) où :

- M est un ensemble fini qui est la mémoire de la stratégie,
- $m_0 \in M$ est la valeur initiale de la mémoire,
- $v_u : M \times V \rightarrow M$ est une fonction d'actualisation,
- $v_n : M \times V_1 \rightarrow E$ est une fonction de prochain mouvement.

La taille de la stratégie est le nombre $|M|$ de valeurs possibles pour la mémoire. Si le jeu est dans un sommet v du joueur 1 et que m est la valeur courante de la mémoire, alors l'arc choisi par la stratégie est $e = v_n(m, v)$ et la valeur de la mémoire est actualisée en $v_u(m, v)$.

Définition 4.17 (Restriction d'un graphe à une stratégie sans mémoire). Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soit σ une stratégie sans mémoire. Le graphe $G_\sigma = (V_1, V_2, E_\sigma, s, d, (\mu_{e,\sigma}, \alpha_{e,\sigma})_{e \in E_\sigma}, (\pi_{v,\sigma})_{v \in V})$ est la représentation en graphe de σ telle que :

- $E_\sigma \subseteq E$ tel que $(v, v') \in E_\sigma$ si et seulement si $v \in V_2$ et $(v, v') \in E$, ou $v \in V_1$ et $(v, v') = \sigma(v)$.
- Pour $e \in E_\sigma$, $\mu_{e,\sigma} = \mu_e$, et pour tout $x \in \mathbb{Z}^n$, nous avons $\alpha_{e,\sigma}(x) = \alpha_e(x)$.

Définition 4.18 (Stratégie gagnante). Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Une stratégie σ du joueur 1 est dite gagnante depuis un sommet $v \in V$ pour $x \in \mathbb{Z}^n$, si pour toutes les exécutions ρ de σ depuis v , nous avons $\rho \in W^x$ avec W^x l'ensemble des chemins gagnants pour le joueur 1 pour x . Nous disons également que le joueur 1 gagne depuis v pour x ou que v est un sommet gagnant pour x .

Le dernier type de stratégie que nous souhaitons définir est celui des stratégies sans danger pour l'énergie. Ce sont des stratégies qui assurent que l'énergie restera positive tant qu'elles sont suivies. De plus ces stratégies assurent que dans le cas où l'énergie reste constante alors la parité minimale est paire. Cette notion est très proche de la définition des stratégies "good-for-energy" utilisées dans [CD12] et nous étudierons les différences entre ces deux notions dans la section suivante.

Définition 4.19 (Stratégie sans danger pour l'énergie).

Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Une stratégie σ du joueur 1 est sans danger pour l'énergie dans un sommet $v \in V$ pour $x \in \mathbb{Z}^n$ si pour toute exécution ρ de σ telle que son sommet initial de ρ soit v nous avons :

- Pour tout cycle C appartenant à la décomposition en cycles de ρ , soit $\phi^x(C) > 0$, soit $\phi^x(C) = 0$ et la priorité minimale des sommets apparaissant dans C soit paire.
- Pour tout préfixe ρ' de ρ , nous avons $\phi^x(\rho') \geq 0$.

4.4.2 Stratégies particulières

Les deux lemmes suivants montrent les ressources nécessaires pour vérifier si une stratégie donnée est sans danger pour l'énergie ou non ou si elle peut forcer la visite d'un sommet de priorité 0 ou non.

En effet nous allons manipuler des stratégies ayant de telles propriétés afin de construire une stratégie gagnante, ce qui nous permettra de ne vérifier que ces propriétés pour savoir s'il existe une stratégie gagnante pour le joueur 1.

Lemme 4.12. *Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soient $v \in V$ un sommet, et σ une stratégie sans mémoire. Le problème de décider si σ est sans danger pour l'énergie dans v pour un $x \in \mathbb{Z}^n$ ou non, peut être résolu en temps exponentiel non déterministe.*

Démonstration. Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soient $x \in \mathbb{Z}^n$, $v \in V$, et une stratégie σ sans mémoire. Soit G_σ la représentation en graphe de σ . Soit $G' = (V'_1, V'_2, E', s, d, (\mu'_e, \alpha'_e)_{e \in E}, (\pi_v)_{v \in V'})$ la restriction de G_σ aux sommets accessibles depuis v . Pour vérifier si la stratégie σ est sans danger pour l'énergie nous avons seulement besoin de vérifier que pour tout γ exécution de σ :

- (A) pour tout ρ apparaissant dans γ tel que ρ soit un chemin simple dont le sommet initial est v , nous avons $\phi^x(\rho) \geq 0$,
- (B) pour tout cycle simple c apparaissant dans la décomposition en cycle de γ , nous avons $\phi^x(c) > 0$, ou $\phi^x(c) = 0$ et la priorité minimale des sommets apparaissant dans c est paire.

Nous commençons par prouver que si nous avons (A) et (B) alors pour tout cycle c apparaissant dans γ nous avons $\phi^x(c) > 0$, ou $\phi^x(c) = 0$ et la priorité minimale des sommets apparaissant dans c est paire.

Soit c un cycle apparaissant dans une exécution de σ . Soit c_i pour $i \in \llbracket 1, k \rrbracket$ les cycles simples formant c . Puisque pour tout c_i nous avons $\phi^x(c_i) \geq 0$ d'après (B), nous avons que $\phi^x(c) \geq 0$. De plus si $\phi^x(c) = 0$ cela signifie que pour tout i , nous avons $\phi^x(c_i) = 0$

d'après (B) et la priorité minimale apparaissant dans chaque c_i est paire. Par conséquent la priorité minimale apparaissant dans c est également paire.

Nous prouvons maintenant que pour toute exécution de σ à partir de v , l'énergie de tous ses préfixes est positive si (A) et (B) sont vraies.

Toute exécution de σ est un chemin $\rho = e_1 e_2 \dots$ de G' dont le sommet initial est v . Il peut être décomposé en une succession de chemins simples potentiellement vides ρ_i et de cycles simples ou des chemins vides c_i tels que $\rho = \rho_1 c_1 \rho_2 c_2 \dots$. Supposons qu'il y ait $j \in \mathbb{N}$ tel que $\phi^x(e_1 \dots e_j) < 0$. Soit i_0 tel que $e_j \in \rho_{i_0} c_{i_0}$, nous avons deux possibilités :

- (1) $e_j \in \rho_{i_0}$,
- (2) $e_j \in c_{i_0}$.

Supposons (1) (resp. (2)), soit γ le préfixe de ρ_{i_0} se terminant avec e_j (resp. le préfixe de c_{i_0}), nous avons $e_1 \dots e_j = \rho_1 c_1 \dots c_{i_0-1} \gamma$ (resp. $e_1 \dots e_j = \rho_1 c_1 \dots \rho_{i_0} \gamma$). Soit $\rho' = \rho_1 \dots \rho_{i_0-1} \gamma$ (resp. $\rho' = \rho_1 \dots \rho_{i_0} \gamma$), comme l'énergie de tout c_i est supérieure ou égale à zéro d'après (B), nous avons $\phi^x(\rho') \leq \phi^x(e_1 \dots e_j)$. Soit ρ_0 le chemin ρ' sans aucun cycle, nous avons $\phi^x(\rho_0) \leq \phi^x(\rho') \leq \phi^x(e_1 \dots e_j)$. Comme il n'y a aucun cycle dans ρ_0 nous avons que ρ_0 est un chemin simple dont le sommet initial est v et donc $\phi^x(\rho_0) \geq 0$ d'après (A). Ainsi $\phi^x(e_1 \dots e_j) \geq 0$ ce qui prouve que dans les deux cas l'énergie de toute exécution de σ est positive si nous avons (A) et (B).

Par conséquent si nous avons σ telle que les propriétés (A) et (B) soient vraies nous avons que σ est sans danger pour l'énergie.

Pour savoir s'il existe un $x \in \mathbb{Z}^n$ tel que la stratégie σ soit sans danger pour l'énergie nous avons besoin de savoir s'il y a un x tel que (A) et (B) soient vraies. Ainsi pour chaque ρ qui est un chemin simple ou un cycle simple dont la priorité minimale des sommets y apparaissant est paire, nous générons une contrainte sur x qui est $\phi^x(\rho) \geq 0$. Pour chaque cycle simple c dont la priorité minimale des sommets y apparaissant est impaire, nous générons une contrainte $\phi^x(c) > 0$. Enfin nous devons vérifier si la zone définie par les contraintes ainsi générées est vide ou non.

Comme nous avons au plus $|V|!$ chemins simples dont le sommet initial est v et au plus $2^{|V|}$ cycles simples, nous avons au plus $|V|! + 2^{|V|}$ contraintes.

Savoir si la conjonction de contraintes affine est satisfiable ou non peut être fait avec la programmation linéaire en nombres entiers (ILP) [Sch98], en temps polynomial non déterministe. Nous avons donc montré que décider si une stratégie σ est sans danger pour l'énergie ou non dans v pour un $x \in \mathbb{Z}^n$ peut être fait au moins en temps exponentiel non déterministe. \square

Lemme 4.13. *Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soient $x \in \mathbb{Z}^n$ et une stratégie σ sans mémoire. Le problème de savoir si σ peut forcer ou non la visite d'un sommet v tel que $\pi_v = 0$ depuis tous les sommets de V , peut être résolu en temps polynomial.*

Démonstration. Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Soit σ une stratégie sans mémoire. Soit $G_\sigma = (V_1, V_2, E_\sigma, s, d, (\mu_{e,\sigma}, \alpha_{e,\sigma})_{e \in E_\sigma}, (\pi_v)_{v \in V})$ la représentation en graphe de σ . Soit $S_0 = \{v \in V, \pi_v = 0\}$ l'ensemble des sommets de parité nulle.

Nous cherchons à vérifier que pour tout v sommet de G_σ nous pouvons atteindre un sommet de S_0 . Soit $v \in V$, en notant avec un léger abus de notations S_0 la propriété des sommets qui est vraie lorsque le sommet considéré appartient à S_0 et qui est fausse sinon, la propriété que nous voulons vérifier peut s'exprimer sous la forme de la formule CTL (Computation Tree Logic) : $AF(S_0)$. En effet nous pouvons représenter les chemins dont le sommet initial est v par des arbres et nous pouvons donc bien chercher à savoir si ces arbres vérifient des propriétés CTL. Comme nous voulons que quel que soit le chemin (A) nous puissions visiter à un moment (F) un sommet appartenant à S_0 , pour chaque $v \in V$ nous voulons bien vérifier que l'arbre représentant les chemins dont le sommet initial est v vérifie la propriété $AF(S_0)$.

Nous avons donc V formules CTL à vérifier (une par sommet) afin de savoir si σ peut forcer ou non la visite d'un sommet v tel que $\pi_v = 0$ depuis tous les sommets de V . Comme les propriétés CTL peuvent être vérifiées en un temps polynomial nous pouvons bien vérifier que σ peut forcer la visite de S_0 depuis tous les sommets de V en temps polynomial. \square

Par la suite nous allons avoir besoin d'utiliser plusieurs lemmes de [CD12] dans nos preuves. Cependant comme nous avons une définition légèrement différente d'une stratégie sans danger pour l'énergie nous prouvons avec le lemme suivant que les conditions utilisées dans [CD12] pour être certain de l'existence d'une stratégie sans danger pour l'énergie sont toujours valides dans notre modèle.

Nous commençons par rappeler la définition d'une stratégie sans danger pour l'énergie présente dans [CD12] que nous allons appeler *stratégie GFE* dans la définition suivante car elle est appelée "good-for-energy" dans [CD12] :

Définition 4.20 (Stratégie GFE). *Soit $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée. Une stratégie σ pour le joueur 1 est dite GFE dans un*

sommet $v \in V$ pour $x \in \mathbb{Z}^n$ si pour toute exécution ρ de σ telle que le sommet initial de ρ soit v , nous avons que pour tout cycle C de la décomposition en cycles de ρ , $\phi^x(C) > 0$, ou $\phi^x(C) = 0$ et la priorité minimale des sommets apparaissant dans C est paire.

Une fois qu'une valeur a été fixée pour les paramètres, la seule différence entre notre modèle de jeu de parité à énergie et le modèle de [CD12] est la présence d'un crédit initial dans [CD12]. Notre définition de stratégie sans danger pour l'énergie peut être vue comme celle d'une stratégie GFE avec un crédit initial de zéro pour le modèle de [CD12].

Puisque notre définition de stratégie sans danger pour l'énergie (pour notre modèle) implique la définition de stratégie sans danger pour l'énergie du modèle de [CD12] (GFE), si une stratégie σ est sans danger pour l'énergie dans un sommet $v \in V$ pour une valeur de paramètre $x \in \mathbb{Z}^n$ alors σ est GFE dans $v \in V$ pour $x \in \mathbb{Z}^n$.

Le prochain lemme montre l'existence d'une stratégie sans danger pour l'énergie ayant une certaine forme à partir du moment où il existe au moins une stratégie gagnante pour le joueur 1. Cela nous est utile pour prouver la complexité du problème que nous considérons.

Lemme 4.14. *Soient $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée et $x \in \mathbb{Z}^n$. Soit Win l'ensemble des sommets gagnants pour le joueur 1 dans G^x , il existe une stratégie sans mémoire pour le joueur 1 qui est sans danger pour l'énergie dans chaque $v \in Win$.*

Démonstration. Tout d'abord, la définition d'une stratégie sans danger pour l'énergie dans un sommet q peut être vue comme celle d'une stratégie gagnante dans un jeu fini de création de cycle à partir de q où le jeu s'arrête lorsqu'un cycle C est formé pendant une partie ρ . Le vainqueur est ensuite déterminé comme suit :

- Le joueur 1 perd s'il existe un préfixe ρ' de ρ tel que nous ayons $\phi^x(\rho') < 0$,
- Si ce n'est pas le cas alors le joueur 1 gagne si $\phi^x(C) > 0$ ou si $\phi^x(C) = 0$ et $\min(\{\pi_v, v \in C\})$ est pair (c'est-à-dire si la priorité minimale apparaissant dans C est paire) et perd sinon.

D'après le théorème 5.1 de [BSV04], qui dit que dans ce que nous avons appelé jeu fini de création de cycle, si l'un des joueurs a une stratégie gagnante, alors il a une stratégie sans mémoire gagnante.

Soient $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée et $x \in \mathbb{Z}^n$. Supposons que le joueur 1 gagne dans G^x depuis un sommet $v \in V$. Si le joueur 1 n'a pas de stratégie sans danger pour l'énergie depuis v , cela signifie que le joueur 2 a une

stratégie sans mémoire gagnante σ_2 dans le jeu fini de création de cycle. Si nous fixons cette stratégie dans G^x cela signifie que tous les cycles ont soit une énergie négative, soit une énergie nulle et une priorité minimale impaire, ou enfin que le joueur 2 arrive à faire prendre une valeur strictement négative à l'énergie. Ainsi le joueur 1 perd G^x depuis v , ce qui est impossible puisque nous avons que $v \in Win$. Par conséquent le joueur 1 a une stratégie gagnante dans le jeu fini de création de cycle, cela signifie qu'il a une stratégie gagnante sans mémoire de ce jeu. Par conséquent le joueur 1 a donc une stratégie σ_v sans danger pour l'énergie depuis v .

Finalement, nous pouvons définir une stratégie σ pour le joueur 1 qui est sans danger pour l'énergie dans chaque sommet de Win :

Nous fixons un ordre total sur les sommets de Win : nous avons alors $v_1 < v_2 < \dots < v_n$. Soit $Out(v_i)$ l'ensemble des sommets apparaissant dans les exécutions de σ_{v_i} . Nous avons alors $\sigma(v_i) = \sigma_{v_j}(v_i)$ avec $j = \min\{k | v_i \in Out(v_k)\}$. Nous avons ainsi une stratégie σ pour le joueur 1 qui est sans danger pour l'énergie dans chaque sommet de Win .

Par conséquent nous avons bien qu'il existe une stratégie sans mémoire qui est sans danger pour l'énergie dans chaque sommet $v \in Win$. \square

4.4.3 Complexité

Les prochains théorèmes prouvent des bornes sur la complexité du problème : il est dans NEXPTIME et PSPACE-difficile.

Théorème 4.2. Soient $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée et $v \in V$.

Le problème de décider s'il existe une valeur de paramètres $x \in \mathbb{Z}^n$ telle que le joueur 1 gagne dans G depuis v est dans NEXPTIME.

Démonstration. Soient $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée et un sommet $v \in V$.

Considérons la machine de Turing non déterministe effectuant les actions suivantes :

1. Deviner Win l'ensemble des sommets gagnants pour le joueur 1 et l'écrire sur la bande.
2. Deviner les deux stratégies sans mémoire σ_{GFE} et σ_0 et les écrire sur la bande.
3. Ecrire les inégalités sur les paramètres qui sont énumérées dans la preuve du lemme 4.12 traduisant que σ_{GFE} est sans danger pour l'énergie dans chaque sommet v de Win (une telle stratégie existe d'après le lemme 4.14).

4. Décider si ces inégalités sont satisfiables ou non et si la stratégie σ_0 peut forcer ou non la visite d'un sommet w tel que $\pi_w = 0$ depuis chaque sommet de Win .

D'après le lemme 10 de [CD12] nous avons que le problème de décider s'il existe un crédit initial tel que le joueur 1 ait une stratégie gagnante depuis un sommet q dans un jeu de parité à énergie (non paramétrée) est dans NP. Il prouve également que le joueur 1 a une stratégie gagnante si et seulement s'il existe deux stratégies sans mémoire, l'une étant GFE et l'autre pouvant forcer la visite d'un sommet de parité nulle. De plus, si une stratégie est sans danger pour l'énergie pour une valeur $y \in \mathbb{Z}^n$ alors cette stratégie est également GFE pour ce y . Nous avons donc que si la machine termine alors il existe un $x \in \mathbb{Z}^n$ tel que le joueur 1 gagne dans G si et seulement si la machine répond oui.

A présent considérons les différentes étapes plus en détails :

- Chaque stratégie peut être décrite en un espace qui est linéaire dans la taille de V_1 . En effet nous avons seulement besoin d'indiquer quel arc doit être choisi par le joueur 1 pour chaque sommet de V_1 .
- Pour écrire le nombre exponentiel de contraintes qui sont générées par le lemme 4.12 nous avons besoin d'un temps exponentiel.
- D'après le lemme 4.13 nous pouvons vérifier si σ_0 peut forcer la visite d'un sommet w tel que $\pi_w = 0$ depuis chaque sommet de Win en temps polynomial et nous pouvons décider si la conjonction de contraintes affine est satisfiable en temps exponentiel non déterministe avec la programmation linéaire en nombres entiers (ILP) [Sch98].

Tout cela pris en compte, la machine que nous avons construite termine toujours et opère dans NEXPTIME. \square

Théorème 4.3. Soient $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ un jeu de parité à énergie paramétrée et le sommet $v \in V$.

Le problème de décider s'il existe une valeur de paramètre $x \in \mathbb{Z}^n$ telle que le joueur 1 gagne dans G depuis v est PSPACE-difficile.

Démonstration. Soit $QBF = q_1 x_1 \dots q_n x_n f(x_1 \dots x_n)$ une formule booléenne quantifiée en forme prénex. C'est-à-dire, pour tout $i \in \llbracket 1, n \rrbracket$ nous avons $q_i \in \{\forall, \exists\}$, avec les x_i des variables booléennes, et f une formule booléenne non quantifiée en forme normale conjonctive : $f(x_1 \dots x_n) = \bigwedge_{j=1}^k c_j$, avec $k \in \mathbb{N}$ et pour tout j , nous avons c_j une clause. Pour toute clause c nous avons $c = \bigvee_{j=1}^l p^{i_j}$ avec $l \in \mathbb{N}$ et pour tout j , nous avons p^{i_j} la variable booléenne x^{i_j} ou sa négation.

Le problème de savoir si une formule booléenne quantifiée est vraie ou non est connu pour être dans PSPACE [GJ79].

Nous allons construire un graphe $G = (V_1, V_2, E, s, n, (\mu_e, \alpha_e)_{e \in E}, (\pi_v)_{v \in V})$ tel qu'il existe une valeur des paramètres $x \in \mathbb{Z}^n$ telle que $W^x \neq \emptyset$ si et seulement si la formule QBF est vraie. Le graphe G aura autant de paramètres qu'il y a de variables booléennes différentes dans QBF . Avec un léger abus de notations nous noterons x_i le i -ème paramètre de x , qui représentera la variable booléenne x_i . Tous les sommets auront la même priorité paire : pour tout $v \in V$, nous avons $\pi_v = 0$, par conséquent le seul critère qu'il reste est celui portant sur l'énergie.

Nous voulons tout d'abord limiter les valeurs possibles de x_i pour qu'elles soient dans $\{0, 1\}$, avec $x_i = 0$ signifiant que x_i est fausse et $x_i = 1$ signifiant que x_i est vraie. Pour cela nous utilisons le gadget décrit dans la figure 4.17. Nous modélisons le fait que $q_i \in \{\forall, \exists\}$ en donnant le contrôle de v_i au joueur 1 ($v_i \in V_1$) si $q_i = \exists$ et au joueur 2 ($v_i \in V_2$) si $q_i = \forall$. Nous avons représenté cela dans la figure par le fait que les sommets du joueur 1 sont en orange, que les sommets du joueur 2 sont en cyan et que les sommets qui sont dans V_1 ou V_2 en fonction des q_i sont oranges avec des hachures cyans.

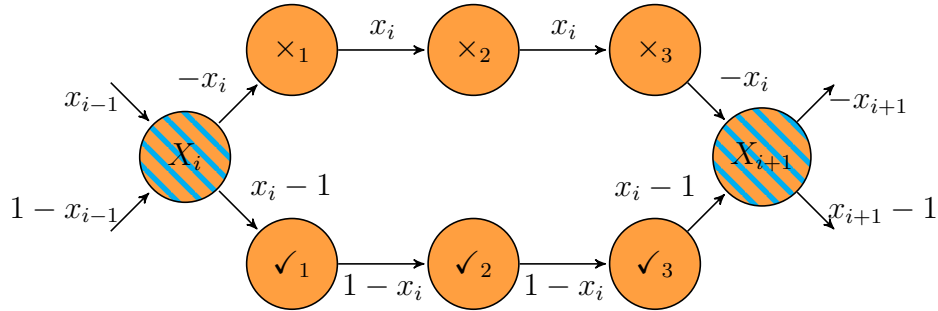


FIGURE 4.17 – Un gadget pour assurer que $x_i \in \{0, 1\}$.

Nous pouvons commencer en remarquant que les arcs d'un gadget ne modifient pas l'énergie et qu'initialement la valeur de l'énergie est nulle. De plus pour le joueur 1 l'énergie doit rester positive ou nulle et la seule possibilité pour que ce soit le cas en empruntant le premier arc est d'avoir :

- $x_i \leq 0$ si l'arc \times_1 est emprunté, et ensuite $x_i \geq 0$ lorsque nous prenons l'arc entre \times_2 et \times_3 .
- $x_i \geq 1$ si l'arc \checkmark_1 est emprunté, et ensuite $x_i \leq 1$ lorsque nous prenons l'arc entre \checkmark_2 et \checkmark_3 .

Ainsi lorsque nous sommes au sommet X_{i+1} nous avons que $x_i \in \{0, 1\}$ et l'énergie est

nulle à nouveau.

Nous montrons dans la figure 4.17 comment nous mettons les gadgets pour chaque dimensions les uns à la suite des autres en commençant par $s = X_1$ et en terminant par un sommet X_{n+1} qui n'est associé à aucune dimension des paramètres. Ce sommet est dans V_2 et est utilisé pour permettre au joueur 2 de choisir parmi toutes les clauses laquelle sera testée comme nous pouvons l'observer sur la figure 4.18. En effet s'il existe au moins une clause qui est fausse, les stratégies gagnantes pour le joueur 2 consistent à tester l'une de ces clauses.

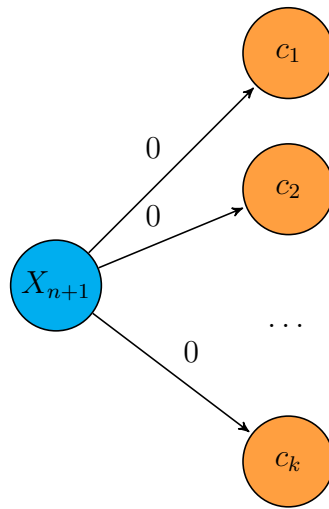


FIGURE 4.18 – Le joueur 2 choisit la clause à tester.

Nous définissons ensuite un gadget pour chaque clause comme montré dans la figure 4.19, dans lequel e_{j_s} dépend du littéral p_{j_s} :

- Si p_{j_s} est la variable booléenne x_i alors $e_{j_s} = x_{j_s} - 1$. En effet p_{j_s} est vrai si x_{j_s} est vraie et nous voulons donc tester si $x_{j_s} = 1$. Comme $x_{j_s} \in \{0, 1\}$ nous n'avons besoin que d'un seul arc et comme l'énergie est nulle nous avons que si $x_{j_s} = 0$ alors le joueur 1 perd bien si cet arc est emprunté.
- Si p_{j_s} est la négation de la variable booléenne x^{j_s} alors $e_{j_s} = -x_{j_s}$. En effet p_{j_s} est vrai si x_{j_s} est fausse et nous voulons donc tester si $x_{j_s} = 0$. Ainsi si $x_{j_s} = 1$ le joueur 1 perd si cet arc est emprunté.

Comme les clauses sont des conjonctions, nous avons que $c_j \in V_1$. En effet s'il existe un littéral p_{j_s} dans c_j qui soit vrai alors c'est une stratégie gagnante pour le joueur 1 d'emprunter l'arc e_{j_s} .

Pour conclure si le littéral p_{j_s} est vrai alors le joueur 1 ne perd pas si l'arc associé est

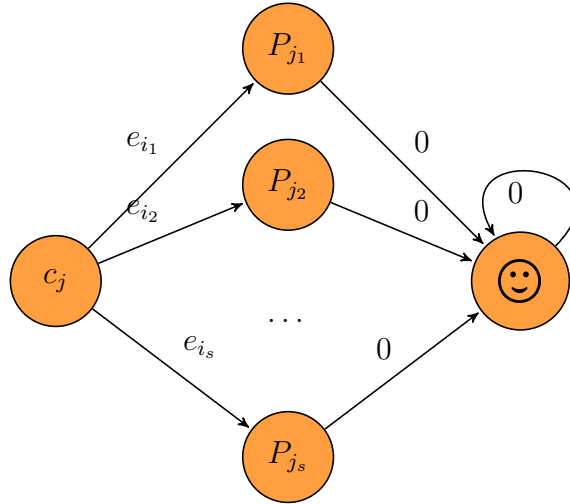


FIGURE 4.19 – Un gadget qui teste si une clause est vraie.

emprunté et le jeu arrive dans un sommet de priorité 0 avec un arc qui part et arrive de ce sommet. Cela permet au joueur 1 de gagner comme montré sur la figure 4.20 dans laquelle les différents gadgets ne sont pas détaillés. En effet si nous arrivons dans le sommet \odot avec une énergie positive cela signifie qu'après avoir contraint la valeur de chaque x_i à être dans $\{0, 1\}$ dans la figure 4.17, le joueur 2 n'a pas pu choisir une clause dans la figure 4.18 telle que tous ses littéraux soient faux ce qui aurait contraint le joueur 1 à choisir d'emprunter un arc représentant un littéral qui est faux dans la figure 4.19. Par conséquent si le joueur 1 gagne dans G depuis s cela signifie que QBF est vraie.

Réciproquement, si QBF est vraie alors il existe une valuation de $(x_i)_1^n$ telle que pour tout i tel que $q_i = \exists$, le joueur 1 ait une stratégie qui force les valeurs correspondantes des x_i telles que toutes les clauses soient vraies. Cela signifie que lorsque le jeu est dans X_{n+1} , le joueur 2 ne peut pas forcer la visite d'un gadget d'une clause fautive, ce qui implique que le joueur 1 a une stratégie gagnante dans chaque gadget correspondant à une clause. Par conséquent si QBF est vraie alors le joueur 1 gagne dans G depuis s .

En conclusion, le joueur 1 gagne dans G depuis s si et seulement si QBF est vraie ce qui prouve que le problème de décider s'il existe une valeur de paramètre $x \in \mathbb{Z}^n$ telle que le joueur 1 gagne dans G depuis v est PSPACE-difficile.

□

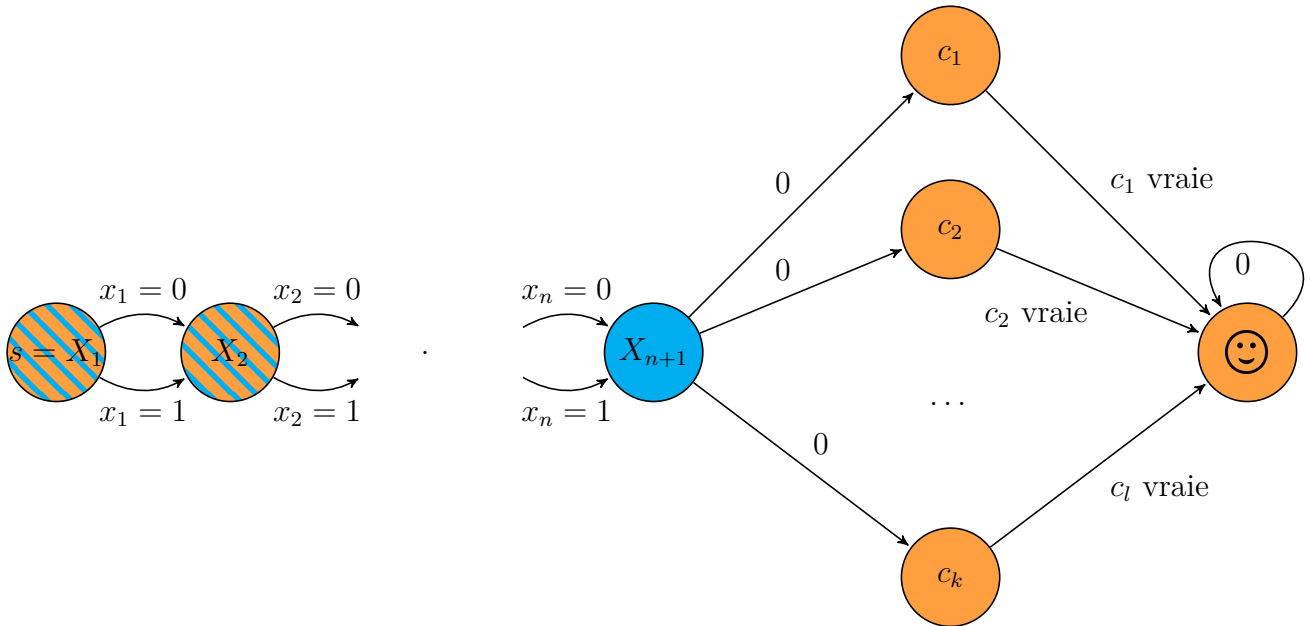


FIGURE 4.20 – Une vue d’ensemble du graphe construit.

4.5 Conclusion

Nous avons étudié le problème de savoir s’il existe une stratégie gagnante pour le joueur 1 dans un jeu de parité à énergie paramétrée dans les cas où il y a un ou deux joueurs, en ajoutant des arcs pouvant fixer l’énergie à une valeur donnée dans le cas à un joueur. Dans ces jeux l’énergie des arcs implique un nombre arbitraire de paramètres à valeurs entières. Nous avons proposé des preuves de décidabilité du problème dans les deux cas ainsi que des bornes sur la complexité du problème. Ce chapitre généralise un travail précédent réalisé par Chatterjee et Doyen dans lequel la version non paramétrée du problème est considérée [CD12]. Nous avons prouvé la décidabilité du problème dans les deux cas et nous avons montré que le problème à un joueur est NP-complet et que le problème à deux joueurs est PSPACE-difficile et NEXPTIME.

De futurs travaux pourraient consister à affiner les bornes sur la complexité ou encore à implémenter notre méthode dans un algorithme afin d’évaluer son efficacité sur des études de cas réels. Nous pourrions également considérer des paramètres prenant des valeurs rationnelles ou réelles. En effet, les preuves concernant la borne supérieure de la décidabilité font appel à la programmation linéaire en nombres entiers à la fois pour le cas à un joueur mais aussi pour celui à deux joueurs. Il serait aussi envisageable d’étudier un modèle de jeux de parité à énergie paramétrée à deux joueurs qui prend en considération

les arcs constants ce qui forcerait à complexifier la notion de stratégie sans danger pour l'énergie. Une autre piste pourrait être d'essayer de généraliser les notions de schéma d'itération et de schéma multiple d'itération aux stratégies afin d'avoir des stratégies ne considérant pas les arcs constants.

CONCLUSION

5.1 Principaux résultats

Nous avons tout d'abord proposé un algorithme visant à trouver les chemins optimaux pour aller d'une unique source à tous les autres sommets d'un graphe paramétré dans lequel les énergies des arcs font appel à un nombre arbitraire de paramètres à valeurs réelles. Puisque ces chemins optimaux changent avec les valeurs des paramètres, le résultat de notre algorithme est un ensemble fini d'arbres, chacun de ces arbres étant associé par l'algorithme à une zone de l'espace des paramètres sur laquelle il est optimal. Ces zones couvrent l'espace des paramètres pour lequel il n'y a pas de cycle d'énergie négative au sein du graphe. Cet algorithme permet de généraliser des travaux préalablement réalisés par Karp et Orlin dans lesquels seul un unique paramètre était considéré [KO81].

Nous avons ensuite considéré, dans le chapitre 3, le problème LU total tout d'abord dans le cas où les paramètres prennent des valeurs entières et que l'énergie est de dimension quelconque, et dans un second temps lorsque les paramètres prennent des valeurs rationnelles et que l'énergie est de dimension supérieure ou égale à deux. Nous avons prouvé la décidabilité avec complexité PSPACE-complète du problème lorsque les paramètres sont à valeurs entières, quelle que soit la dimension de l'énergie. Nous apportons également la preuve de l'indécidabilité du problème lorsque les paramètres prennent des valeurs dans tout ensemble qui comprend les nombres rationnels et que l'énergie est de dimension supérieure ou égale à deux, à partir du moment où au moins un paramètre intervient dans l'énergie des arcs.

Enfin dans le dernier chapitre, nous avons étudié le problème de savoir s'il existe une stratégie gagnante pour le joueur 1 dans un jeu de parité à énergie paramétrée dans les cas où il y a un ou deux joueurs, en ajoutant des arcs pouvant fixer l'énergie à une valeur donnée dans le cas à un joueur. Dans ces jeux l'énergie des arcs implique un nombre arbitraire de paramètres à valeurs entières. Nous avons proposé des preuves de décidabilité du problème dans les deux cas ainsi que des bornes sur la complexité du

problème. Ce chapitre généralise un travail précédent réalisé par Chatterjee et Doyen dans lequel la version non paramétrée du problème est considérée [CD12]. Nous avons prouvé la décidabilité du problème dans les deux cas et nous avons montré que le problème à un joueur est NP-complet et que le problème à deux joueurs est PSPACE-difficile et NEXPTIME.

5.1.1 Perspectives et futurs travaux

De futurs travaux visant à prolonger le travail sur les plus courts chemins d'une source à tous les autres sommets d'un graphe paramétré pourraient consister en une implémentation de l'algorithme ainsi qu'une évaluation de son efficacité sur des études de cas réels. Il serait également possible de s'intéresser au cas où les paramètres prennent des valeurs entières. En effet dans ce cas la notion d'arbres voisins ne permettrait plus d'énumérer les arbres de manière exhaustive car certains arbres voisins qui sont optimaux pour certaines valeurs rationnelles ne sont optimaux pour aucune valeur entière ce qui implique des modifications importantes de l'algorithme si nous voulons qu'il puisse prendre en compte ce cas. Nous pourrions également essayer de généraliser la notion d'arbre voisin aux cas où l'énergie a plusieurs dimensions afin de chercher à étendre notre algorithme aux graphes multi-dimensionnels.

Les travaux sur les problèmes LU totaux pourraient être prolongés par l'étude de leurs variantes. Le fait que la preuve d'indécidabilité du problème LU total avec paramètres à valeurs rationnelles n'est plus valide lorsque l'énergie est de dimension 1 peut laisser penser que le problème pourrait être décidable dans ce cas. Nous pourrions aussi étudier les variations de ces problèmes consistant à connaître les valeurs de paramètre pour que tous les chemins vérifient les conditions. Contrairement aux problèmes que nous avons considérés dans lesquels nous cherchons l'existence d'un chemin qui vérifie des conditions. Il serait également possible de se pencher sur d'autres types de problèmes, notamment lorsque l'on supprime la contrainte de la borne supérieure. En effet les gadgets utilisés pour tester l'égalité à zéro nécessitent que l'énergie des chemins soit bornée.

D'autres travaux pourraient consister à affiner les bornes de la complexité de nos résultats sur les jeux à un ou deux joueurs, ou encore à implémenter notre méthode dans un algorithme afin d'évaluer son efficacité sur des études de cas réels. Nous pourrions également considérer des paramètres prenant des valeurs rationnelles ou réelles. En effet les preuves concernant la borne supérieure de la décidabilité font appel à la programmation linéaire en nombres entiers à la fois pour le cas à un joueur mais aussi pour celui à

deux joueurs. Il serait aussi envisageable d'étudier un modèle de jeux de parité à énergie paramétrée à deux joueurs qui prend en considération les arcs constants ce qui forcerait à complexifier la notion de stratégie bonne pour l'énergie.

Une autre approche possible serait d'essayer d'adapter nos preuves à de la recherche de chemins vérifiant des propriétés portant sur d'autres critères que l'énergie totale. Nous pourrions par exemple nous demander si les méthodes que nous avons utilisées s'appliquent lorsque nous ne cherchons plus un chemin dont l'énergie reste bornée, ou supérieure à une borne donnée, mais lorsque c'est l'énergie moyenne qui est soumise à ces bornes. Nous pourrions également nous poser ces questions concernant la variation moyenne d'énergie, dont les différents problèmes sans paramètre ont été largement étudiés. Le principal obstacle à l'utilisation de nos méthodes sur des critères comme la variation d'énergie moyenne ou l'énergie moyenne est la nature même du critère qui nécessite de prendre en compte la longueur du chemin, critère que nous n'avons pas eu besoin de considérer lorsque nous travaillions sur l'énergie totale des chemins.

Nous pouvons aussi penser à combiner ces différentes contraintes sur les chemins pour créer de nouveaux problèmes à étudier. Nous pourrions chercher à savoir s'il est possible de rendre décidable le problème LU total avec des paramètres rationnels lorsque l'énergie est de dimension supérieure ou égale à deux, en ajoutant des contraintes sur l'énergie moyenne ou sur la variation d'énergie moyenne. Cela pourrait peut être, en fonction des relations entre les bornes sur l'énergie et les bornes sur l'énergie moyenne ou sur la variation d'énergie moyenne, limiter par exemple le nombre de test à zéro qu'il est possible d'effectuer sur un chemin donné.

Il est également envisageable de généraliser les problèmes sur l'énergie moyenne ou sur la variation moyenne d'énergie ainsi que les problèmes étant une combinaison de plusieurs contraintes évoquées précédemment à des jeux à deux joueurs et de chercher s'il existe une stratégie pour que le joueur 1 puisse forcer le chemin à respecter les contraintes. Chaque jeu ainsi créé aurait lui-même une variante qui consisterait à ajouter une contrainte sur la parité des chemins comme nous l'avons fait dans le dernier chapitre.

BIBLIOGRAPHIE

- [AHV93] Rajeev ALUR, Thomas A. HENZINGER et Moshe Y. VARDI, « Parametric Real-time Reasoning », in : *ACM Symposium on Theory of Computing*, 1993, p. 592-601.
- [And+19a] Étienne ANDRÉ et al., « Minimal-Time Synthesis for Parametric Timed Automata », in : *TACAS, Part II*, sous la dir. de Tomáš VOJNAR et Lijun ZHANG, t. 11428, Lecture Notes in Computer Science, Prague, Czech Republic : Springer, 2019, p. 211-228.
- [And+19b] Étienne ANDRÉ et al., « Parametric Analyses of Attack-fault Trees », in : *19th International Conference on Application of Concurrency to System Design (ACSD 2019)*, sous la dir. de Jörg KELLER et Wojciech PENCZEK, Aachen, Germany : IEEE Computer Society, juin 2019.
- [And10] Etienne ANDRÉ, « An inverse method for the synthesis of timing parameters in concurrent systems », Theses, École normale supérieure de Cachan - ENS Cachan, déc. 2010, p. 170-185, URL : <https://tel.archives-ouvertes.fr/tel-00595268>.
- [BBL08] Patricia BOUYER, Ed BRINKSMA et Kim G. LARSEN, « Optimal Infinite Scheduling for Multi-Priced Timed Automata », in : *Formal Methods in System Design* 32.1 (fév. 2008), p. 2-23, DOI : 10.1007/s10703-007-0043-4, URL : <http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BBL-FMSD04.pdf>.
- [BJK10] Tomáš BRÁZDIL, Petr JANČAR et Antonín KUČERA, « Reachability games on extended vector addition systems with states », in : *International Colloquium on Automata, Languages, and Programming*, Springer, 2010, p. 478-489.
- [BO14] Daniel BUNDALA et Joël OUAKNINE, « Advances in parametric real-time reasoning », in : *International Symposium on Mathematical Foundations of Computer Science*, Springer, 2014, p. 123-134.

-
- [BO17] Daniel BUNDALA et Joël OUAKNINE, « On parametric timed automata and one-counter machines », in : *Inf. Comput.* 253 (2017), p. 272-303.
- [Bou+08] Patricia BOUYER et al., « Infinite Runs in Weighted Timed Automata with Energy Constraints », in : *Formal Modeling and Analysis of Timed Systems*, sous la dir. de Franck CASSEZ et Claude JARD, Berlin, Heidelberg : Springer Berlin Heidelberg, 2008, p. 33-47, ISBN : 978-3-540-85778-5.
- [Bou+17] Patricia BOUYER et al., *Bounding Average-energy Games*, 2017, arXiv : 1610.07858 [cs.LG].
- [Bou+18] Patricia BOUYER et al., « Average-energy games », in : *Acta Informatica* 55.2 (mars 2018), p. 91-127, ISSN : 1432-0525, DOI : 10.1007/s00236-016-0274-1, URL : <https://doi.org/10.1007/s00236-016-0274-1>.
- [BSV04] Henrik BJÖRKLUND, Sven SANDBERG et Sergei VOROBYOV, « Memoryless determinacy of parity and mean payoff games : A simple proof », in : *THEORETICAL COMPUTER SCIENCE* 310 (2004), p. 365-378.
- [Cay89] Arthur CAYLEY, « A theorem on trees », in : *Quart. J. Math.* 23 (1889), p. 376-378.
- [CD12] Krishnendu CHATTERJEE et Laurent DOYEN, « Energy parity games », in : *Theor. Comput. Sci.* 458 (2012), p. 49-60, DOI : 10.1016/j.tcs.2012.07.038, URL : <https://doi.org/10.1016/j.tcs.2012.07.038>.
- [CES86] E. M. CLARKE, E. A. EMERSON et A. P. SISTLA, « Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications », in : *ACM Trans. Program. Lang. Syst.* 8.2 (avr. 1986), p. 244-263, ISSN : 0164-0925, DOI : 10.1145/5397.5399, URL : <https://doi.org/10.1145/5397.5399>.
- [Cha+10a] Sourav CHAKRABORTY et al., « Two-phase Algorithms for the Parametric Shortest Path Problem », in : *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, sous la dir. de Jean-Yves MARION et Thomas SCHWENTICK, t. 5, LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, p. 167-178.
- [Cha+10b] Krishnendu CHATTERJEE et al., « Generalized mean-payoff and energy games », in : *arXiv preprint arXiv :1007.1669* (2010).

-
- [CNM06] Yang C.C., Liu N. et Sageman M., « Analyzing the Terrorist Social Networks with Visualization Tools. », in : *Intelligence and Security Informatics*, sous la dir. de Mehrotra S. et al., t. 3975, Lecture Notes in Computer Science, Berlin, Heidelberg : Springer, 2006.
- [Dij59] E. W. DIJKSTRA, « A note on two problems in connexion with graphs », in : *Numerische Mathematik*, t. 1, 1, 1959, p. 269-271, DOI : 10.1007/BF01386390.
- [Fah+11] U. FAHRENBERG et al., « Energy Games in Multiweighted Automata », in : *Proceedings of the 8th International Colloquium on Theoretical Aspects of Computing (ICTAC'11)*, t. 6916, LNCS, Springer-Verlag, 2011, p. 95-115.
- [FJ15] John FEARNLEY et Marcin JURDZIŃSKI, « Reachability in two-clock timed automata is PSPACE-complete », in : *Information and Computation* 243 (2015), 40th International Colloquium on Automata, Languages and Programming (ICALP 2013), p. 26-36, ISSN : 0890-5401, DOI : <https://doi.org/10.1016/j.ic.2014.12.004>, URL : <https://www.sciencedirect.com/science/article/pii/S0890540114001564>.
- [Flo62] Robert W. FLOYD, « Algorithm 97 : Shortest Path », in : t. 5, 6, New York, NY, USA : Association for Computing Machinery, juin 1962, p. 345, DOI : 10.1145/367766.368168, URL : <https://doi.org/10.1145/367766.368168>.
- [For56] L. R. FORD, *Network Flow Theory*. Santa Monica, CA : RAND Corporation, 1956.
- [GH21] Stefan GÖLLER et Mathieu HILAIRE, « Reachability in Two-Parametric Timed Automata with One Parameter Is EXPSPACE-Complete », in : *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, sous la dir. de Markus BLÄSER et Benjamin MONMEGE, t. 187, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany : Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 36 :1-36 :18, ISBN : 978-3-95977-180-1, DOI : 10.4230/LIPIcs.STACS.2021.36, URL : <https://drops.dagstuhl.de/opus/volltexte/2021/13681>.
- [GJ79] Michael R GAREY et David S JOHNSON, *Computers and intractability*, t. 174, freeman San Francisco, 1979.

-
- [GL81] Peter GÁCS et Laszlo LOVÁSZ, « Khachiyan's algorithm for linear programming », in : *Mathematical Programming studies*, 1981, p. 61-68.
- [Göl+10] Stefan GÖLLER et al., « Model checking succinct and parametric one-counter automata », in : *International Colloquium on Automata, Languages, and Programming*, Springer, 2010, p. 575-586.
- [Gou19] GOUBAULT-LARRECQ, *Classes de complexité en espace, alternances, et tutti quanti*, Cours d'informatique, 2019.
- [Haa+09] Christoph HAASE et al., « Reachability in Succinct and Parametric One-Counter Automata », in : *CONCUR 2009 - Concurrency Theory*, sous la dir. de Mario BRAVETTI et Gianluigi ZAVATTARO, Berlin, Heidelberg : Springer Berlin Heidelberg, 2009, p. 369-383, ISBN : 978-3-642-04081-8.
- [HNR68] Peter E. HART, Nils J. NILSSON et Bertram RAPHAEL, « A Formal Basis for the Heuristic Determination of Minimum Cost Paths », in : t. 4, 2, 1968, p. 100-107, DOI : 10.1109/TSSC.1968.300136.
- [HNR72] Peter E. HART, Nils J. NILSSON et Bertram RAPHAEL, « Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths" », in : *SIGART Bull.* 37 (déc. 1972), p. 28-29, ISSN : 0163-5719, DOI : 10.1145/1056777.1056779, URL : <https://doi.org/10.1145/1056777.1056779>.
- [Jez+on] Loïg JEZEQUEL et al., « Parametric energy-parity games », in : En cours de soumission.
- [JGR13] Line JUHL, Kim GULDSTRAND LARSEN et Jean-François RASKIN, « Optimal Bounds for Multiweighted and Parametrised Energy Games », in : *Theories of Programming and Formal Methods : Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*, sous la dir. de Zhiming LIU, Jim WOODCOCK et Huibiao ZHU, Berlin, Heidelberg : Springer Berlin Heidelberg, 2013, p. 244-255, ISBN : 978-3-642-39698-4, DOI : 10.1007/978-3-642-39698-4_15, URL : https://doi.org/10.1007/978-3-642-39698-4_15.
- [JLS21a] Loïg JEZEQUEL, Didier LIME et Bastien SÉRÉE, « A Lazy Query Scheme for Reachability Analysis in Petri Nets », in : *Application and Theory of Petri Nets and Concurrency*, sous la dir. de Didier BUCHS et Josep CARMONA, Cham : Springer International Publishing, 2021, p. 360-378, ISBN : 978-3-030-76983-3.

-
- [JLS21b] Loïc JEZEQUEL, Didier LIME et Bastien SÉRÉE, « An Algorithm for Single-Source Shortest Paths Enumeration in Parameterized Weighted Graphs », in : *Language and Automata Theory and Applications*, sous la dir. d'Alberto LEPORATI et al., Cham : Springer International Publishing, 2021, p. 279-290, ISBN : 978-3-030-68195-1.
- [KO81] Richard M. KARP et James B. ORLIN, « Parametric shortest path algorithms with an application to cyclic staffing », in : *Discrete Applied Mathematics* 3.1 (1981), p. 37-45, ISSN : 0166-218X.
- [Lim+09] Didier LIME et al., « Romeo : A Parametric Model-Checker for Petri Nets with Stopwatches », in : *Tools and Algorithms for the Construction and Analysis of Systems*, sous la dir. de Stefan KOWALEWSKI et Anna PHILIPPOU, Berlin, Heidelberg : Springer Berlin Heidelberg, 2009, p. 54-57, ISBN : 978-3-642-00768-2.
- [LR05] Kim Guldstrand LARSEN et Jacob Illum RASMUSSEN, « Optimal Conditional Reachability for Multi-priced Timed Automata », in : *Foundations of Software Science and Computational Structures*, sous la dir. de Vladimiro SASSONE, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 234-249, ISBN : 978-3-540-31982-5.
- [LR08] Kim Guldstrand LARSEN et Jacob Illum RASMUSSEN, « Optimal reachability for multi-priced timed automata », in : *Theoretical Computer Science* 390.2 (2008), Foundations of Software Science and Computational Structures, p. 197-213, ISSN : 0304-3975, DOI : <https://doi.org/10.1016/j.tcs.2007.09.021>, URL : <https://www.sciencedirect.com/science/article/pii/S0304397507007050>.
- [LRS19] Didier LIME, Olivier H. ROUX et Charlotte SEIDNER, « Parameter Synthesis for Bounded Cost Reachability in Time Petri Nets », in : *40th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2019)*, sous la dir. de Susanna DONATELLI et Stefan HAAR, t. 11522, Lecture Notes in Computer Science, Aachen, Germany : Springer, juin 2019, p. 406-425.
- [Min67] Marvin MINSKY, « Computation : Finite and infinite machines prentice hall », in : *Inc., Engelwood Cliffs, NJ* (1967).

-
- [MLA20] Nicolas MARKEY, Didier LIME et Étienne ANDRÉ, « Language preservation problems in parametric timed automata », in : *Logical Methods in Computer Science* 16 (2020).
- [NMD18] Gaumont N., Panahi M. et Chavalarias D., « Reconstruction of the socio-semantic dynamics of political activist Twitter networks—Method and application to the 2017 French presidential election », in : *PLoS ONE* 13(9) : e0201879, 2018.
- [NPR16] Reino NISKANEN, Igor POTAPOV et Julien REICHERT, « Undecidability of Two-dimensional Robot Games », in : *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, sous la dir. de Piotr FALISZEWSKI, Anca MUSCHOLL et Rolf NIEDERMEIER, t. 58, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany : Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 73 :1-73 :13, ISBN : 978-3-95977-016-3, DOI : 10.4230/LIPIcs.MFCS.2016.73, URL : <http://drops.dagstuhl.de/opus/volltexte/2016/6483>.
- [Sch98] Alexander SCHRIJVER, *Theory of linear and integer programming*, John Wiley & Sons, 1998.
- [War62] Stephen WARSHALL, « A Theorem on Boolean Matrices », in : t. 9, 1, New York, NY, USA : Association for Computing Machinery, jan. 1962, p. 11-12, DOI : 10.1145/321105.321107, URL : <https://doi.org/10.1145/321105.321107>.
- [Woo+09] Jim WOODCOCK et al., « Formal methods : Practice and experience », in : *ACM Comput. Surv.*, 41(4), October 2009.
- [YTO91] Neal YOUNG, Robert TARJAN et James ORLIN, « Faster Parametric Shortest Path and Minimum Balance Algorithms », in : *Networks* 21.2 (1991), p. 205-221.

Titre : Problèmes d'optimisation dans les graphes paramétrés

Mot clés : automate, graphes paramétrés, plus court chemin, jeux sur les graphes, objectifs de parité, contraintes sur l'énergie, model-checking

Résumé : Nous considérons des graphes orientés pondérés dont l'énergie est paramétrée. Nous proposons dans un premier temps un algorithme qui, étant donné un graphe et un de ses sommets, renvoie des arbres, chaque arbre représentant les plus courts chemins depuis la source vers tous les autres sommets du graphe pour une zone particulière de l'espace des paramètres. De plus l'union de ces zones couvre l'espace des paramètres. Nous considérons ensuite l'accessibilité dans les graphes à énergie multi-dimensionnelle, avec un type de contraintes plus absolues qui imposent que l'énergie reste entre des bornes. Nous montrons la décidabilité et la complexité du problème quel que soit le nombre de paramètres et de dimensions lorsque les paramètres prennent des valeurs entières. Nous montrons également l'indécidabilité de ce problème avec au moins un paramètre lorsque la dimension est supérieure ou égale à deux. Nous étudions enfin des jeux de parité à un et deux joueurs sur les graphes paramétrés dont l'objectif est la conjonction d'une condition qualitative sur la parité et d'une condition quantitative : l'énergie doit rester positive. Nous montrons la décidabilité et prouvons des bornes de la complexité du problème de la recherche d'une stratégie gagnante dans les cas à un et à deux joueurs.

Title: Optimisation problems in parametrized graphs

Keywords: automata, parametrized graphs, shortest paths, games on graphs, parity objectives, energy constraints, model-checking

Abstract: We are considering weighted oriented graphs with parametrized energy. Firstly we propose an algorithm that, given a graph and one of its vertices, returns trees, every tree representing shortest-paths from the source to every other vertex for a particular zone of the parameter space. Moreover, union of these zones is a covering of the parameter space. Then we consider reachability in graphs with multi-dimensional energy, with stricter constraints that enforce the energy to stay between bounds. We prove decidability and complexity of this problem regardless of the dimension and the number of parameters when parameters take integer values. We also prove the undecidability of this problem when there is at least one parameter and the dimension is at least two. Finally we study parity games on parametrized graphs with one and two players whose objective is the conjunction of a qualitative condition on the parity and quantitative one : energy must stay positive. We show the decidability and prove bounds on the complexity of the problem of searching a winning strategy in both cases with one and two players.