

N° d'ordre 19/2018-D/MT

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET
DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE
FACULTÉ DES MATHÉMATIQUES



Thèse

Présentée pour l'obtention du grade de **Docteur en Sciences**

En : **Mathématiques**

Spécialité : **Recherche Opérationnelle : Mathématiques Discrètes et
Optimisation**

Par : **KHEFFACHE Rezika**

Sujet

**Algorithmes d'approximation pour des variantes
du problème du voyageur de commerce**

Soutenue publiquement le 09/07/2018 devant le jury composé de :

M. AIDER Méziane ;	Professeur ;	à l'USTHB ;	Président.
M. OUAFI Rachid ;	Professeur ;	à l'USTHB ;	Directeur de thèse.
M. HIFI Mhand ;	Professeur ;	à l'UPJV ;	Co-Directeur de thèse.
M. SEMRI Ahmed ;	Professeur ;	à l' USTHB ;	Examineur.
M. AIDENE Mohamed ;	Professeur ;	à l' UMMTO ;	Examineur.
M. SADI Bachir ;	Maître de Conférences A	à l'UMMTO ;	Examineur.

À la mémoire de mon père.

À ma mère.

À ma très chère fille Zira.

À toute ma famille et belle famille.

Remerciements

Je tiens tout d'abord à remercier chaleureusement mes directeurs de thèse : Monsieur OUAFI Rachid, Professeur à l'USTHB et Monsieur HIFI Mhand, Professeur à l'université de Picardie JULES VERNE. Ils ont su diriger mes travaux avec beaucoup de disponibilité, de tact et d'intérêt. Ils m'ont toujours accordé le temps nécessaire pour partager avec moi leurs idées et leur grande expérience. Je leur suis également reconnaissante de m'avoir assuré un encadrement rigoureux tout au long de ces années. Qu'ils trouvent ici l'expression de ma profonde gratitude.

Mes sincères remerciements s'adressent également à Monsieur GIANNAKOS Aristotelis, Maître de Conférences à l'université de Picardie JULES VERNE, pour son intérêt, sa grande disponibilité, ses remarques pertinentes et ses nombreux conseils durant la réalisation de ma thèse.

Je tiens à exprimer mes plus vifs remerciements à Monsieur AIDER Méziane, Professeur à l'USTHB, de m'avoir fait l'honneur de présider mon jury de thèse, ainsi qu'aux membres du jury, Messieurs SEMRI Ahmed, Professeur à l'USTHB, AIDENE Mohamed, Professeur à l'UMMTO et SADI Bachir, Maître de Conférences à l'UMMTO de m'avoir fait l'honneur d'accepter de juger ce travail.

Je voudrais remercier aussi tous les membres de l'unité EPROAD, plus particulièrement Mondher, Dhikra et Labib, pour l'accueil chaleureux qu'ils m'ont témoigné et l'ambiance scientifique et de convivialité qu'ils ont su créer, sans oublier la secrétaire du laboratoire Madame CARON Anne et Lei Wu pour leur gentillesse.

Enfin, j'adresse toute mon affection à ma famille et en particulier à ma mère et à mes sœurs, pour leur attention et encouragements tout au long de ces années, pour leur soutien moral, leur réconfort et leur amour. Mes sincères remerciements à mon mari Madjid pour son soutien, sa patience et son aide. Un clin d'œil à ma fille Zira que j'adore beaucoup.

Je ne saurais terminer sans souligner le soutien amical et chaleureux de mes copines et mes collègues durant ce parcours doctoral. Je m'abstiens de les nommer tellement la liste est longue. Je les remercie tous.

Table des matières

Résumé	6
Introduction générale	7
1 Optimisation combinatoire et complexité	14
1.1 Introduction	14
1.2 Optimisation combinatoire	14
1.3 Complexité	15
1.3.1 Concepts de base	15
1.3.2 Algorithme	16
1.3.3 Complexité d'un algorithme	16
1.3.4 Complexité dans le pire cas	16
1.3.5 Algorithme efficace	16
1.3.6 Complexité d'un problème	16
1.3.7 Les classes de problèmes	17
1.3.8 Classe des problèmes PO	19
1.3.9 Classe des problèmes NPO	19
1.4 Méthodes de résolution	19
1.4.1 Méthodes exactes	19
1.4.2 Méthodes approchées	20
1.5 Approximation polynomiale	20
1.6 Rapports d'approximation	21
1.6.1 Rapport standard	21
1.6.2 Rapport différentiel	22

1.7	Algorithmes approchés	22
1.8	Classes d'approximation	23
1.9	Inapproximabilité	26
1.10	Conclusion	26
2	État de l'art sur l'approximation du TSP et ses variantes	27
2.1	Introduction	27
2.2	Définition du problème général du voyageur de commerce	28
2.3	Variantes du TSP	29
2.3.1	TSP asymétrique	29
2.3.2	TSP symétrique	29
2.3.3	TSP métrique	29
2.3.4	TSP graphique	30
2.4	Variante chaîne du TSP	30
2.5	État de l'art sur l'approximation du TSP et ses variantes	31
2.6	Généralisation au problème du TSP multiple	32
2.7	Inapproximabilité	34
2.8	Conclusion	35
3	Un algorithme d'approximation pour le problème de la chaîne hamiltonienne à trois dépôts	36
3.1	Introduction	36
3.2	Définition du problème	36
3.3	Algorithme d'approximation pour le 3DHPP	37
3.4	Exemple illustratif	44
3.5	Conclusion	46
4	Algorithmes d'approximation pour le kDHPP et le TSP graphique	47
4.1	Introduction	47
4.2	Notations et définitions	47
4.3	Problème de la chaîne hamiltonienne à k dépôts	50
4.3.1	Quelques résultats	50

4.3.2	Algorithme d'approximation pour le k DHPP	51
4.4	Exemple pour $k = 3$	53
4.5	Le TSP graphique dans un graphe facteur critique et 2-sommet connexe . .	56
4.5.1	Un algorithme d'approximation	56
4.5.2	Application de l'algorithme A_{HT} à un exemple	59
4.6	Conclusion	61
5	Approximation du problème de k voyageurs de commerce	62
5.1	Introduction	62
5.2	Définitions et préliminaires	62
5.2.1	Un multigraphe	62
5.2.2	T -join	63
5.3	Définition du problème	63
5.4	Algorithme	65
5.5	Application de l'algorithme pour $k = 2$	66
5.6	Conclusion	67
	Conclusion et perspectives	68
	Annexe	70
	Bibliographie	75

Table des figures

2.1	Exemple d'une tournée $:(L,P,B,Y,S,R,L)$	29
2.2	Exemple d'un TSP graphique	30
3.1	Décomposition des sommets de V_0 en deux couplages disjoints, Cas 1 : $ \mathcal{H}_1 $ impair, $ \mathcal{H}_2 = \mathcal{H}_3 = 0$	41
3.2	Décomposition des sommets de V_0 en deux couplages disjoints, Cas 2 : $ \mathcal{H}_1 $ impair, $ \mathcal{H}_2 $ pair et $ \mathcal{H}_3 = 0$	42
3.3	Décomposition des sommets de V_0 en deux couplages disjoints, Cas 3 : $ \mathcal{H}_1 $ impair, $ \mathcal{H}_2 $ et $ \mathcal{H}_3 $ sont pairs	42
3.4	Décomposition des sommets de V_0 en deux couplages disjoints, Cas 4 : $ \mathcal{H}_1 $, $ \mathcal{H}_2 $ et $ \mathcal{H}_3 $ sont impairs	43
3.5	Étape 1 :(a) La forêt de coût minimum	44
3.6	Étape 1 :(b) En rouge, les destinations de degré impair, $ V_0 =$ $ O_1 \cup O_2 \cup O_3 = 9$	44
3.7	Étape 2 : Le couplage partiel de coût minimum de cardinal $\frac{ V_0 -3}{2} = 3$	45
3.8	Étape 3 et 4 : Ajout des arêtes à la forêt pour obtenir une chaîne eulérienne $E_r = \{m_1, 7, 6, 5, 4, 2, 3, 2, 1, d_1, d_2, m_1, 9, 10, 11, 12, m_2, m_3, 14, m_2, d_3\}$	45
3.9	Étape 5 : Suppression des arêtes pour obtenir une chaîne hamiltonienne $E_{sc} = \{m_1, 7, 6, 5, 4, 2, 3, 1, d_1, d_2, 9, 10, 11, 12, m_2, m_3, 14, d_3\}$	45
3.10	Étape 5 : Suppression des arêtes de coût 0 pour obtenir les trois chaînes hamiltoniennes $\{d_1, 1, 3, 2, 4, 5, 6, 7, m_1\}, \{d_2, 9, 10, 11, 12, m_2\}, \{d_3, 14, m_3\}$	46
4.1	Une décomposition en oreilles	49
4.2	Une instance du graphe G	54

4.3	Illustration de l'étape 1 (à gauche, notée(a)) et de l'étape 2 (à droite, notée (b)) de l'algorithme A_{kD}	54
4.4	Illustration de l'étape 3 (à gauche, notée (a)) et étape 4 (à droite, notée (b)) de l'algorithme A_{kD} sur le dernier graphe résultant de la Fig. 4.3 . . .	55
4.5	La solution finale obtenue par l'algorithme A_{kD}	56
4.6	Une instance du problème.	59
4.7	Illustration de l'étape 1 (à gauche, notée (a)) et de l'étape 2 (à droite, notée (b)) de A_{HT}	60
4.8	Illustration de l'étape 3 (à gauche, notée (a)) et de l'étape 4 (à droite, notée (b)) de A_{HT}	60
5.1	Un 4-join connexe	63
5.2	Graphe 2-arête connexe	66
5.3	Décomposition en oreilles du graphe G	66
5.4	Suppression des oreilles triviales	66
5.5	\emptyset -join	66
5.6	2-join	66
5.7	4-join	66
5.8	Le graphe eulérien obtenu	67
5.9	La solution finale	67

Résumé

De nombreux problèmes d'optimisation combinatoire sont NP-complets, la recherche de solutions optimales pour ces problèmes, peut donc être prohibitive. Les algorithmes d'approximation offrent un compromis intéressant : par définition, ils s'exécutent en temps polynomial et fournissent des solutions dont la qualité est garantie.

Cette thèse s'inscrit dans le domaine de l'étude de l'approximation polynomiale de quelques variantes du problème du voyageur de commerce à savoir : le problème de la chaîne hamiltonienne à trois dépôts dans un graphe métrique, puis une généralisation à k dépôts dans un graphe cubique, ensuite le problème du cycle hamiltonien et enfin le problème de k voyageurs de commerce dans un graphe 2-sommet connexe.

Mots clés : Optimisation combinatoire, Problème du voyageur de commerce, Complexité, Approximation polynomiale.

Introduction générale

Une des parties essentielles de la Recherche Opérationnelle est l'optimisation combinatoire, qui s'est développée en réponse au grand nombre de problèmes pratiques qu'elle peut modéliser et qu'elle se propose de résoudre. La plupart de ces problèmes appartiennent à la classe NP-difficile et ne possèdent donc de solutions algorithmiques efficaces valables pour toutes les données. Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle et en intelligence artificielle.

Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés dans leurs applications et comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante. Par conséquent, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes dites : approchées.

Ces méthodes visent à résoudre des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant. Et depuis ces dernières années, on constate que l'intérêt porté à ces méthodes augmente continuellement en recherche opérationnelle et en intelligence artificielle. Ce succès tient pour une grande part à leur capacité de fournir des solutions d'excellente qualité au prix d'une consommation en ressources réduite. La perte du caractère optimal se voit donc compensée par la diminution du temps de calcul.

Une classe bien particulière de méthodes approchées, celles qui garantissent la qualité de leurs solutions, c'est-à-dire capables de calculer des solutions dont la distance en valeur à la solution optimale est petite ou dont la position dans le rang des valeurs objectives réalisables est assez proche de la valeur optimale et assez loin de la pire valeur réalisable. Ces deux critères de mesure de la qualité des solutions d'un algorithme approché sont appelés, respectivement, approximation standard ou classique et approximation différentielle.

La plupart des problèmes réels sont tels : leurs meilleurs algorithmes connus sont de complexité exponentielle. La majorité de ces problèmes sont classés NP-difficiles. C'est notamment le cas de la plupart des problèmes de routage (exemple : le voyageur de commerce). La recherche d'algorithmes approchés est une démarche principalement adressée à cette classe de problèmes.

Le problème du voyageur de commerce (en anglais Traveling Salesman Problem(TSP)), étudié depuis le 19ème siècle, est l'un des plus connus dans le domaine de la recherche opérationnelle. William Rowan Hamilton a posé ce problème sous forme de jeu pour la première fois en 1859. Sous sa forme la plus classique, son énoncé est le suivant : «Un voyageur de commerce doit visiter une et une seule fois un nombre fini de villes et revenir à son point d'origine. Trouver l'ordre de visite des villes qui minimise la distance totale parcourue par le voyageur».

Les domaines d'application sont nombreux : les problèmes de logistique, les problèmes de transport de marchandises ou de personnes, et plus largement toutes sortes de problèmes d'ordonnancement. Certains problèmes rencontrés dans l'industrie se modélisent sous la forme d'un problème du voyageur de commerce.

L'approche de résolution la plus naïve de ce problème est l'énumération de tous les cycles possibles et de trouver le plus court. Sachant que le temps mis par un tel

algorithme est prohibitif dès que le nombre de villes commence à augmenter.

Par conséquent, on ne connaît pas d'algorithme permettant de trouver une solution exacte en un temps polynomial pour ce problème. De plus, la version décisionnelle : étant donnée une distance d , existe-t-il une chaîne plus petite que d passant par toutes les villes ? est connue comme étant un problème NP-complet.

Dans notre travail, nous nous sommes intéressés à l'approximation polynomiale de quelques variantes du problème du voyageur de commerce à savoir : Le problème de la chaîne hamiltonienne à trois dépôts, puis à k dépôts, le problème du cycle hamiltonien et enfin le problème de k voyageurs de commerce.

En accord avec ce que nous venons de présenter, ce document est organisé comme suit :

Le premier chapitre est consacré aux définitions et aux notions de base sur la complexité algorithmique, les problèmes d'optimisation et les méthodes de résolution de ces problèmes, en particulier les méthodes approchées.

Dans le second chapitre, un état de l'art est présenté sur l'approximation du problème du voyageur de commerce et ses variantes.

Nous étudions dans le troisième chapitre, l'approximation de l'une des variantes du problème du voyageur de commerce qui est le problème de la chaîne hamiltonienne à trois dépôts noté (3DHPP). Pour ce problème, nous présentons un algorithme d'approximation de rapport $\frac{3}{2}$ dans le cas où les coûts sont symétriques et vérifient l'inégalité triangulaire.

Le quatrième chapitre, traite deux autres variantes : la première étant une généralisation du problème de la chaîne hamiltonienne à trois dépôts appelée le problème de la chaîne hamiltonienne à k dépôts (k DHPP). Pour cette variante, un algorithme polynomial de rapport de $\frac{5}{3}$ qui donne une solution avec au plus $(\frac{5}{3}n - \frac{4k-2}{3})$ arêtes est présenté. Cet algorithme est basé sur la recherche d'une forêt et d'un couplage parfait. La deuxième est le problème du cycle hamiltonien dans un graphe facteur critique et

2-sommet connexe. Dans ce cas, nous introduisons la notion de décomposition en oreilles d'un graphe et nous présentons un algorithme polynomial de rapport de $\frac{7}{6}$.

Enfin, le cinquième chapitre est consacré à l'étude du problème de k voyageurs de commerce dans un graphe métrique, 2-arête connexe et les voyageurs doivent parcourir toutes les chaînes en passant par tous les sommets sans spécifier les extrémités. Dans ce cas, nous montrons l'existence d'un algorithme d'approximation asymptotique de rapport de $\frac{3}{2}$.

Une annexe est donnée à la fin de ce manuscrit où les principales notions de la théorie des graphes utilisées dans ce document sont définies.

Liste de travaux scientifiques

Publications internationales

1. A. Giannakos, M. Hifi, R. Kheffache, R. Ouafi, An approximation algorithm for the k - fixed depots problem, Computers & Industrial Engineering 111, 2017, pp 50-55.
2. ARISTOTELIS GIANNAKOS, MHAND HIFI, REZIKA KHEFFACHE, RACHID OUAFI, Using T-joins to approximate solutions for min graphic k-path TSP, 21th BALCOR, Belgrade, Serbia, September 7-11, 2013. Proceeding (N.Mladenovic, G. Savic, eds.) (ISBN :978-86-7680-285-2) University of Belgrade Press, 2014, pp 33-37.
3. Aristotelis Giannakos, Mhand Hifi, ReziKa Kheffache, Rachid Ouafi, An Approximation Algorithm for the Three Depots Hamiltonian Path Problem ; In book,(ISBN : 978-1-4614-5133-4) Optimization Theory, Decision Making, and Operations Research Applications, Publisher : Springer 2013, pp 351-359.

Conférences internationales

1. R. Kheffache, A. Giannakos, M. Hifi, R. Ouafi, « Using Nice Ears Decomposition to Approximate TSP », International Conference on Recent Advances in Pure and Applied Mathematics «ICRAPAM14», 6-9 November, 2014, Antalya, Turkey.
2. R. Kheffache, A. Giannakos, M. Hifi, R. Ouafi, « An approximation algorithm for metric TSP and optimally connected », 24th British Combinatorial Conference

- 2013 «**BCC2013**», du 30 Juin au 6 Juillet 2013, Royal Holloway University of London.
3. R. Kheffache, A. Giannakos , M. Hifi, R. Ouafi, « Approximation du problème de kDHPP dans un graphe cubique », Colloque International sur l'Optimisation et Système d'Information, «**COSI'13**» Alger, du 9 au 11 Juin 2013 .
 4. R. Kheffache, A. Giannakos, M. Hifi, R. Ouafi, « An approximation algorithm for graphic TSP », International Conference : Mathematical Science& Applications, Abu Dhabi, UAE, December 26-30, 2012.
 5. R. Kheffache, « Approximation polynomiale : cas du problème du chemin Hamiltonien à trois dépôts », séminaire organisé par le laboratoire EPROAD de l'université de Picardie Jules Verne à Amiens, France, 26 Octobre 2012.
 6. R. Kheffache, A. Giannakos, M. Hifi, R. Ouafi, « A $\frac{3}{2}$ -Approximation Algorithm for Three Depots Hamiltonian Path Problem », Forth International Conference on Mathematical Sciences, «**ICM12**», Al Ain University, UAE, 11-14 March 2012.
 7. R. Kheffache, M. Hifi, R. Ouafi, « An Approximation Algorithm for Three Depots Hamiltonian Path Problems », 1st International Symposium & 10th Conference on Operational Research «**BALCOR11**», September 22-24, 2011, Thessaloniki, Greece.
 8. R. Kheffache et R. Ouafi, « Un algorithme d'approximation pour le problème du chemin Hamiltonien à trois dépôts ». Cinquième Conférence Internationale en Recherche Opérationnelle «**CIRO10**», Marrakech, Maroc 24-27 Mai 2010.

Conférences nationales

1. R. Kheffache, M. Hifi, R. Kheffache, R. Ouafi, « An Upper Bound for k Depots Hamiltonian Path Problem », Congrès des Mathématiciens Algériens, «**CMA14**», Tlemcen du 11 au 13 Mai 2014.
2. R. Kheffache, A.Giannakos, M. Hifi, R. Ouafi, « A $\frac{3}{2}$ -Approximation Algorithm for Three Depots Hamiltonian Path Problem », Congrès des Mathématiciens Algériens, «**CMA12**», Annaba du 07 au 08 Mars 2012.

Chapitre 1

Optimisation combinatoire et complexité

1.1 Introduction

Ce chapitre est une collecte de définitions et de notions de base indispensables à la compréhension du travail réalisé. Ces notions sont centrées sur la complexité algorithmique, les différentes méthodes de résolution des problèmes d'optimisation, en particulier, les méthodes approchées. Enfin, pour plus de détails et compréhension, nous renvoyons le lecteur à quelques ouvrages sur la théorie de la complexité, citons par exemple : le livre de Garey et Johnson[40], l'article de Papadimitriou et Yannakakis[74] : Optimization, approximation and complexity classes ou celui de Krentel[58] : The complexity of optimization problems ou : Complexité et approximation polynomiale de Paschos [76].

1.2 Optimisation combinatoire

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématique discrète et en informatique. Son importance se justifie d'une part, par la grande difficulté des problèmes d'optimisation et d'autre part, par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire.

Un problème d'optimisation combinatoire est défini par un ensemble d'instances. A chaque instance du problème est associé un ensemble discret de solutions L , un sous-ensemble S de L représentant les solutions admissibles (réalisables) et une fonction du coût f (ou fonction objectif) qui assigne à chaque solution $s \in S$ le nombre réel (ou entier) $f(s)$. Résoudre un tel problème (plus précisément une telle instance du problème) consiste à trouver une solution $s^* \in S$ optimisant la valeur de la fonction du coût f . Une telle solution s^* est appelée solution optimale ou un optimum global. Nous avons donc la définition suivante :

Définition 1.2.1. [75] Une instance I d'un problème de minimisation est un couple (S, f) où $S \subseteq L$ est un ensemble fini de solutions admissibles, et f une fonction du coût ou (objectif) à minimiser $f : S \rightarrow \mathbb{R}$, il s'agit de déterminer s^* tel que $f(s^*) \leq f(s)$, pour tout élément s de S .

Notons que d'une manière similaire, on peut également définir les problèmes de maximisation en remplaçant simplement \leq par \geq .

1.3 Complexité

Afin de mesurer la difficulté d'un problème donné et la comparer avec celles d'autres problèmes pour pouvoir dire qu'un tel problème est plus facile à résoudre que l'autre, une théorie de la complexité a été développée et permet de classer les problèmes faciles et difficiles.

1.3.1 Concepts de base

- Un problème est une question générale qui possède des paramètres dont la valeur n'est pas connue.
- Une instance d'un problème est obtenue en affectant une valeur à chacun de ses paramètres.

- A toute instance I d'un problème (P), on associe un nombre $m(I)$ qui mesure la longueur des données de cette instance et qu'on appelle la taille de l'instance I .

1.3.2 Algorithme

Un algorithme est une procédure de calcul bien définie qui prend en entrée une valeur, ou un ensemble de valeurs, et qui donne en sortie une valeur, ou un ensemble de valeurs. Un algorithme est donc une séquence d'étapes de calcul qui transforment l'entrée en sortie.

Définition 1.3.1. Si A est l'algorithme de résolution d'un problème (P) et I une instance de ce problème, on associe au couple (A, I) un entier $\mu(A, I)$ représentant le nombre d'opérations élémentaires (addition, multiplication, comparaison...) effectuées par l'algorithme A pour la résolution de l'instance I du problème (P).

1.3.3 Complexité d'un algorithme

La complexité d'un algorithme est le nombre d'opérations élémentaires qu'il doit effectuer pour mener à bien un calcul en fonction de la taille des données d'entrée.

1.3.4 Complexité dans le pire cas

Le plus grand nombre $\mu(A, I)$ sur l'ensemble de toutes les instances ayant la même taille est appelé complexité dans le pire cas de l'algorithme A .

1.3.5 Algorithme efficace

Un algorithme est efficace si sa complexité est majorée par un polynôme en la taille des données.

1.3.6 Complexité d'un problème

La complexité d'un problème (P) est la complexité du meilleur algorithme qui résout (P).

1.3.7 Les classes de problèmes

Pour pouvoir exposer la notion de classes des problèmes, il est tout d'abord nécessaire de distinguer les problèmes de décision des problèmes d'optimisation combinatoire.

Un problème de décision est un problème pour lequel la réponse est «oui» ou «non».

Notons qu'il est possible d'associer à chaque problème d'optimisation un problème de décision en introduisant un seuil α correspondant à la fonction objectif f . Le problème de décision devient : « existe-t-il une solution réalisable s telle que : $f(s) \leq$ ou $(\geq) \alpha$? »

Bien que la théorie de la complexité se concentre sur des problèmes de décision, elle peut être étendue aux problèmes d'optimisation.

Définition 1.3.2. Soient g et f deux fonctions définies sur une partie des nombres réels.

L'ensemble $O(g)$ est défini ainsi :

$$O(g) = \{f / \exists a > 0, \exists n_0 \in \mathbb{N} \text{ tels que } \forall x \geq n_0, |f(x)| \leq a|g(x)|\}$$

La classe des problèmes P

La classe P regroupe les problèmes de décision résolus par des algorithmes polynomiaux. Un algorithme polynomial est défini comme un algorithme dont le temps d'exécution est en $O(h(x))$ où h est un polynôme et x est la longueur d'entrée. Les problèmes appartenant à cette classe sont dits faciles [81].

La classe des problèmes NP et les algorithmes non déterministes

Définition 1.3.3. Un algorithme non déterministe est caractérisé par l'utilisation d'une instruction «choix» opérant sur un ensemble fini et en sélectionnant un élément sans que la manière dont le choix est effectué ne soit précisée. Il est caractérisé par le fait que s'il existe une manière d'effectuer le choix qui conduit à la réponse «oui» alors c'est selon cette manière que le choix est effectué.

Définition 1.3.4. Un problème de décision appartient à la classe NP (non deterministic polynomial problem) s'il existe un algorithme non déterministe polynomial pour sa résolution.

Définition 1.3.5. Un problème de reconnaissance (ou de décision) appartient à la classe NP si étant donné une proposition d'une solution au problème, il est possible de vérifier en un temps polynomial que cette solution a pour réponse vrai.

Les algorithmes polynomiaux sont des cas particuliers des algorithmes non déterministes. Aussi tout problème de décision qui peut être résolu par un algorithme polynomial, et qui donc appartient à la classe P, appartient aussi à la classe NP, d'où $P \subseteq NP$.

La classe des problèmes NP-complets

Définition 1.3.6. Un problème (P_1) se réduit polynomialement à un problème (P_2) s'il existe un algorithme polynomial de résolution de (P_1) qui utilise un algorithme résolvant (P_2) .

La classe des problèmes NP-complets contient les problèmes les plus difficiles de NP. Ces problèmes semblent ne pas être dans P (bien qu'on ne possède pas actuellement, une réponse définitive à la question $P=NP$). S'il existe un algorithme polynomial pour résoudre efficacement un problème NP-complet, nous pouvons alors utiliser cet algorithme pour résoudre tous les problèmes de NP. En effet,

Définition 1.3.7. Un problème (P) est NP-complet s'il est dans NP et si tout problème de NP se réduit polynomialement à (P) .

Par conséquent, il est très utile de connaître une variété de problèmes NP-complets. Le premier problème qui a été prouvé comme étant NP-complet par Cook en 1971[24], est le problème de satisfaisabilité (SAT) qui peut être défini comme suit : étant donnée une expression booléenne contenant uniquement les opérations (et, ou, non, les variables et les parenthèses), existe-t-il une affectation en «vrai» et «faux» de ses variables de manière à ce que l'expression booléenne prenne la valeur «vrai»? Dans le cas où la réponse est oui, l'expression sera dite satisfiable.

Définition 1.3.8. Un problème d'optimisation combinatoire est dit NP-difficile si le problème de décision associé est NP-complet.

1.3.8 Classe des problèmes PO

La classe PO est la classe des problèmes d'optimisation qui admettent un algorithme permettant de trouver une solution optimale.

1.3.9 Classe des problèmes NPO

La classe NPO est la classe des problèmes d'optimisation qui admettent un algorithme non déterministe de complexité polynomiale en fonction de la taille des données. Cette classe a pour vocation de traiter les problèmes d'optimisation sans référence aux problèmes décisionnels associés.

1.4 Méthodes de résolution

Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles.

Étant donnée l'importance de ces problèmes, de nombreuses méthodes ont été développées en recherche opérationnelle et en intelligence artificielle. Ces méthodes peuvent être classées en deux grandes catégories : méthodes exactes et méthodes approchées.

1.4.1 Méthodes exactes

Les méthodes de résolution exactes sont nombreuses et se caractérisent par le fait qu'elles permettent d'obtenir une ou plusieurs solutions optimales.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi ces méthodes, on trouve les méthodes de Séparation et Evaluation Progressive (SEP) ou Branch and Bound (B& B).

Les méthodes exactes permettent de trouver des solutions optimales pour des problèmes de tailles raisonnables. Ces méthodes rencontrent généralement des difficultés face aux applications de tailles importantes car le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème. Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille.

1.4.2 Méthodes approchées

Pour les problèmes de grande taille, les méthodes exactes ne sont pas envisageables de par leur temps de calcul qui est exponentiel et donc sans aucun intérêt en pratique. Dans ce cas, il est possible d'utiliser des méthodes approchées qui fournissent une solution réalisable aussi bonne que possible. Ce sont souvent des méthodes heuristiques, très satisfaisantes dans le cas où l'obtention d'une solution optimale n'est pas absolument requise.

Il est naturel de s'intéresser alors à la qualité des solutions fournies par l'algorithme utilisé et avoir une garantie quant à cette qualité. C'est le rôle de l'approximation polynomiale.

1.5 Approximation polynomiale

L'approximation polynomiale consiste à construire des algorithmes qui fournissent en un temps polynomial des solutions réalisables ayant une certaine qualité pour des problèmes NP-difficiles. Cette solution garantit une certaine qualité qu'on mesure avec un rapport dit : rapport d'approximation.

La problématique de l'approximation polynomiale est apparue peu de temps après le théorème fondamental de la théorie de la complexité des algorithmes de Cook qui affirme que le problème de satisfaisabilité (SAT) est NP-complet. L'article fondateur du domaine est l'article « Approximation algorithms for combinatorial problem » écrit par

Johnson en 1974[50]. Par la suite, le domaine de recherche de l'approximation polynomiale s'est développé. Une variété de problèmes d'optimisation NP-complets dans leur version décisionnelle ont été étudiés [10].

1.6 Rapports d'approximation

L'objectif de l'approximation polynomiale est de trouver des solutions ayant une garantie de bonne qualité, donc il est utile de savoir ce qu'est une bonne qualité de la solution, par conséquent définir une mesure de qualité d'une solution, c'est à dire : mesurer la distance entre la solution obtenue et la valeur optimale.

Étant donné un problème (P) , soit A un algorithme qui fournit sur toute instance I de (P) une solution $A(I)$ telle que $|m(I, A(I)) - opt(I)| \leq \beta$ pour une certaine constante β . Une telle mesure est adaptée dans certains cas très particuliers, car peu de problèmes sont approximables dans ce sens. Une mesure comparant la valeur de la solution à la valeur optimale est alors introduite. Cette mesure consiste à calculer l'erreur relative. Elle est appelée rapport d'approximation standard de la solution.

1.6.1 Rapport standard

Soit A un algorithme polynomial pour le problème (P) et s une solution réalisable d'une instance I du problème (P) .

Le rapport d'approximation standard de s sur I est :

$$\rho(I) = \frac{s(I)}{opt(I)}$$

Ce rapport est dans $[0, 1]$ si (P) est à maximiser et dans $[1, \infty[$ si (P) est à minimiser. Dans les deux cas, plus il est proche de 1, meilleure est la solution.

Ce rapport a été utilisé pour mesurer la qualité d'une solution dans plusieurs travaux sur l'approximation [76, 47, 89, 91].

Il semble aussi qu'il est intéressant de ne pas mesurer la qualité d'une solution par

rapport uniquement à la valeur optimale, mais on peut aussi analyser la solution vis-à-vis de l'ensemble des valeurs réalisables. Pour cela, Ausiello et al.[8] ont introduit un autre rapport dit rapport différentiel.

1.6.2 Rapport différentiel

Le rapport différentiel a été introduit en 1977 [31]. Dans ce rapport, on mesure la qualité d'une solution non seulement en fonction de la valeur d'une solution optimale mais aussi en fonction de la valeur d'une pire solution notée $\lambda(I)$. En d'autres termes, $\lambda(I)$ est la valeur d'une solution optimale du problème obtenu en changeant le but du problème (maximiser au lieu de minimiser, ou inversement). Le rapport d'approximation différentiel de s sur I est défini comme suit :

$$\delta(I) = \frac{|\lambda(I) - s(I)|}{|opt(I) - \lambda(I)|}$$

Plus les rapports sont proches de 1, meilleure est la solution.

Par convention, lorsque $opt(I) = \lambda(I)$, on fixera $\delta(I) = 1$ pour toute solution $s(I)$ (puisque toute solution est optimale). La dissymétrie entre maximisation et minimisation s'efface dans ce cas puisque dans les deux cas le rapport est dans $[0,1]$ et plus il est proche de 1, meilleure est la solution.

Ce rapport a été utilisé pour des problèmes spécifiques [9, 93, 29], une discussion sur les mesures d'approximation est proposée dans [30]. Son emploi a été introduit pour étudier différents problèmes [68, 69].

Avec la définition des rapports d'approximation, nous avons une mesure de la qualité d'une solution. Nous introduisons maintenant la notion d'algorithmes approchés.

1.7 Algorithmes approchés

Soit (P) un problème d'optimisation et une fonction $\rho : I \mapsto R^+$.

Un algorithme approché A pour (P) est un algorithme qui donne, pour toute instance I de (P) , une solution réalisable $A(I)$. On dit que A est ρ -approché si, pour toute instance I , le rapport d'approximation de $A(I)$ est meilleur que $\rho(I)$. Cette définition est valable pour un problème de maximisation ou de minimisation, pour l'approximation standard ou pour l'approximation différentielle.

Nous nous intéresserons à des algorithmes polynomiaux et principalement à des garanties constantes ou dépendant uniquement de la taille de l'instance. Dans ce cas, étant donné une fonction $\phi(I)$ de \mathbb{N} dans \mathbb{R}^+ , on dit que l'algorithme A est $\phi(I)$ -approché si pour toute instance I le rapport d'approximation de $A(I)$ est meilleur que $\phi(|I|)$.

Par conséquent, l'approximation polynomiale consiste à déterminer des garanties de performance les meilleures possibles pour les problèmes NPO.

1.8 Classes d'approximation

Comme les comportements des problèmes NPO sont très différents en terme d'approximabilité, vu que pour certains l'obtention d'algorithmes à rapport constant semble facile, mais pour d'autres problèmes la tâche semble être difficile, les problèmes de NPO sont regroupés en classes d'approximation.

La première classe est constituée de problèmes approximables à rapport constant (indépendant de l'instance).

Un problème (P) de NPO est dans APX (resp., DAPX) s'il existe une constante ρ telle que (P) est approximable à rapport standard (resp., différentiel) ρ .

On distingue aussi d'autres problèmes pour lesquels, on ne peut trouver un algorithme approché à rapport constant. Dans ce cas, on peut approcher ces problèmes par une fonction ρ qui dépend de la taille de l'instance I . Cette fonction tendra vers l'infini dans le cas d'une minimisation en approximation standard et vers zéro dans les autres cas. Il s'agit de trouver une meilleure fonction ρ celle qui tend le moins vite vers l'infini ou

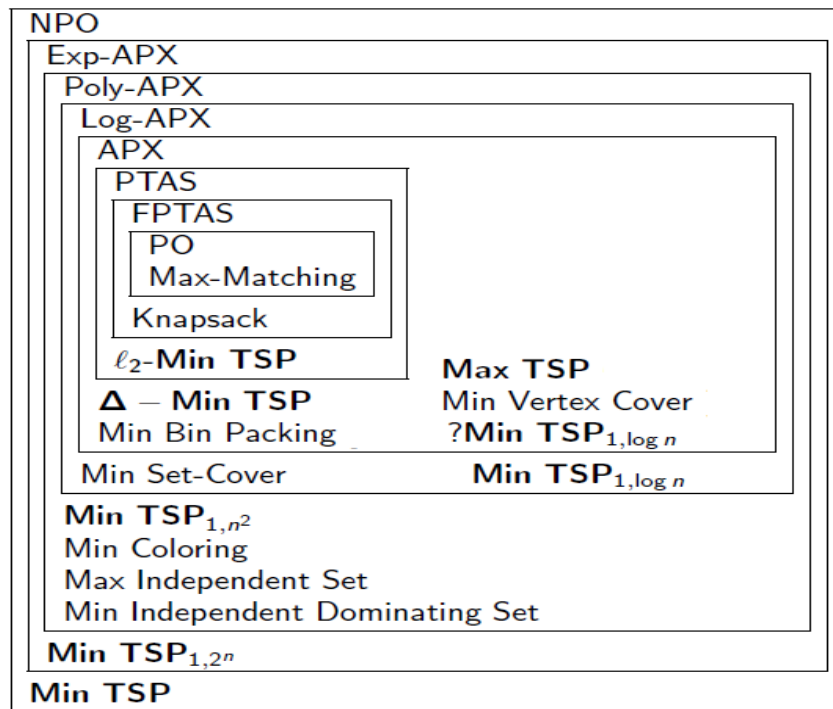
zéro. Les deux classes de fonctions les plus classiques sont les fonctions polynomiales et logarithmique.

Un problème (P) de NPO est dans Poly-APX (resp., Poly-DAPX) s'il existe un polynôme p (à valeur strictement positives) tel que (P) est approximable à rapport standard p si le but de (P) est à minimiser et $\frac{1}{p}$ si le but est à maximiser (resp., à rapport différentiel $\frac{1}{p}$).

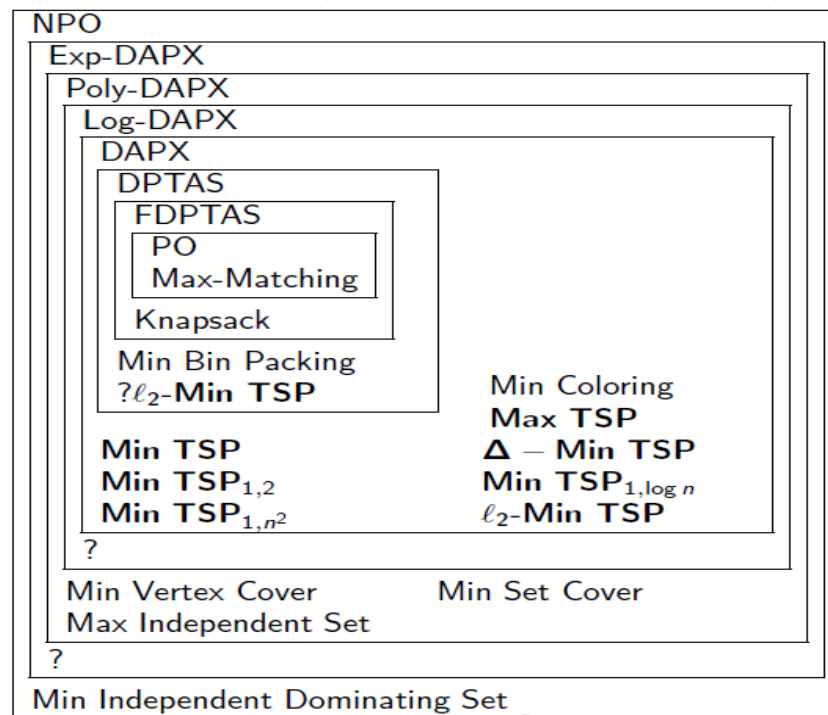
Un problème (P) de NPO est dans Log-APX (resp., Log-Dapx) s'il existe une constante c telle que (P) est approximable à rapport standard $c \text{Log}(\cdot)$ si le but de (P) est à minimiser et $\frac{1}{(c \text{Log}(\cdot))}$ si le but est à maximiser.

Il existe d'autres classes d'approximation que nous n'avons pas évoqué dans ce manuscrit, vu que notre travail n'a pas de rapport avec ces classes. Nous orientons le lecteur intéressé pour plus de détails vers les ouvrages suivants [5, 6, 31, 87, 89].

Le tableau ci dessous nous donne quelques classifications de quelques problèmes en approximation [28].



Structure de NPO (rapport d'approximation standard)



Structure de NPO (rapport d'approximation différentiel)

1.9 Inapproximabilité

L'analyse d'un problème combinatoire dans le cadre de l'approximation polynomiale comporte deux aspects : le développement d'algorithmes approchés afin d'obtenir des garanties de performance, on parle de résultats positifs d'approximation, un autre aspect d'analyse est de prouver si le problème n'est pas approximable, on parle de résultats négatifs.

1.10 Conclusion

Nous avons présenté dans ce chapitre les notions préliminaires essentielles sur l'approximation polynomiale qui consiste à trouver des algorithmes polynomiaux pour les problèmes de NPO et qui garantissent une certaine qualité.

Chapitre 2

État de l'art sur l'approximation du TSP et ses variantes

2.1 Introduction

Dans ce chapitre, nous présentons un état de l'art sur l'approximation du problème du voyageur de commerce et ses quelques variantes, cela permet toutefois de couvrir les tenants et aboutissants de l'essentiel de ce qui a été fait sur l'approximation de ces problèmes.

Le problème du voyageur de commerce (ou TSP pour Traveling Salesman Problem) a été évoqué pour la première fois en 1932 par le mathématicien Viennois Karl Menger [61]. Étant donné un ensemble de villes et une distance entre chaque paire de ces villes. Il s'agit de trouver une tournée qu'un voyageur doit parcourir en passant par toutes les villes une et une seule fois et ayant une distance totale minimale.

Le TSP est l'un des problèmes d'optimisation combinatoire le plus étudié [21]. En effet, il se pose dans différents contextes d'applications réelles : ramassage scolaire, ordonnancement de production, problèmes de logistique, de transport...etc (voir Cook [21] et Lawler et al [61]). Il existe aussi une variété d'autres contextes auxquels il peut s'appliquer par exemple, dans la construction des cartes mères des ordinateurs pour

minimiser la longueur des fils de cuivre entre les puces.

En tant que problème d'optimisation, le TSP appartient à la classe des problèmes NP-difficiles [52]. En effet, malgré des études approfondies, on ne connaît pas de méthodes de résolution permettant d'obtenir des solutions exactes en un temps raisonnable pour de grandes instances du problème. Le temps pour trouver une solution optimale augmente de façon exponentielle en fonction de la taille du problème. Le nombre de cycles hamiltoniens étant égal à $(n-1)!/2$.

Les chercheurs ont fourni de très grands efforts dans la résolution de ce problème par des méthodes approchées en particulier par des algorithmes d'approximation polynomiale afin de trouver des solutions ayant une certaine qualité en un temps polynomial.

2.2 Définition du problème général du voyageur de commerce

Une instance du problème général du voyageur de commerce se formule comme suit : Soit $G = (V, E)$ un graphe complet non orienté à n sommets et m arêtes. Pour chaque arête $(u, v) \in E$, on associe un coût $C(u, v) > 0$ (appelé aussi longueur ou valeur). Le problème est de trouver un plus court cycle passant par tous les sommets une seule et unique fois sauf pour le sommet de départ sur lequel on revient. On cherche donc à minimiser la somme des distances parcourues entre les sommets d'un cycle élémentaire. On note ce problème Min TSP.

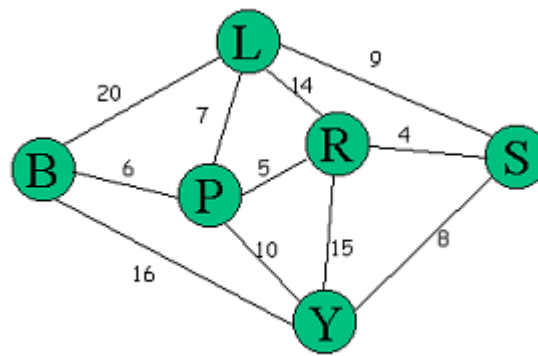


FIG. 2.1 – Exemple d'une tournée : (L,P,B,Y,S,R,L)

Il existe de nombreuses variantes pour le TSP, en fonction des propriétés de la fonction coût. Nous nous contentons de présenter uniquement celles qui ont en rapport avec le travail réalisé dans cette thèse.

2.3 Variantes du TSP

2.3.1 TSP asymétrique

Lorsque les coûts sur les arêtes sont définis comme suit : $C(u, v) \geq 0$ pour tout $u, v \in V$ et $C(u, v)$ peut être différent de $C(v, u)$, on parle de TSP asymétrique noté ATSP.

2.3.2 TSP symétrique

Le TSP est dit symétrique si les coûts vérifient l'égalité : $C(v, w) = C(w, v)$ pour tout $v, w \in V$. On le note : Min-TSP.

2.3.3 TSP métrique

C'est un cas spécial du TSP symétrique, noté Δ -Min TSP, où les coûts vérifient les inégalités triangulaires :

$$C(u, w) \leq C(u, v) + C(v, w), \quad \forall u, v, w \in V$$

2.3.4 TSP graphique

Soit $G = (V, E)$ un graphe où V est un ensemble de sommets et E un ensemble d'arêtes. Le TSP graphique est un cas particulier du TSP symétrique, lorsque les coûts des arêtes de E sont définis comme suit : $C(v, w) = 1$ si l'arête (v, w) appartient à E et $C(v, w) =$ la longueur (nombre d'arêtes) de la plus petite chaîne reliant v à w si l'arête (v, w) n'est pas dans E . Ce problème est noté Min-GTSP.

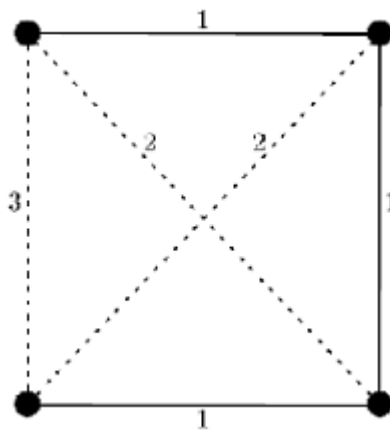


FIG. 2.2 – Exemple d'un TSP graphique

2.4 Variante chaîne du TSP

Le problème de la chaîne du voyageur de commerce (Δ -Min PTSP), en plus d'une métrique (V, C) , deux points sont donnés $s, t \in V$ et le but est de trouver une chaîne de s à t visitant chaque point exactement une fois, ce problème est NP-difficile.

Il existe des versions du TSP où l'objectif est de trouver une tournée du coût total maximum. Nous n'avons pas traité ces versions car nous nous sommes intéressés aux cas de minimisation.

2.5 État de l'art sur l'approximation du TSP et ses variantes

Le TSP dans sa forme générale est inapproximable (il n'existe pas d'algorithme d'approximation de rapport constant) à moins que $P=NP$ [42].

En 1976, Christofides [20] a conçu un algorithme d'approximation pour le TSP métrique de rapport de $\frac{3}{2}$. Son algorithme est basé sur :

- La construction d'un arbre de coût minimum.
- La recherche d'un couplage parfait de coût minimum en utilisant les sommets de degré impair de l'arbre.
- L'ajout des arêtes de l'arbre et du couplage pour avoir un graphe eulérien.
- Raccourcir les arêtes pour avoir un cycle hamiltonien.

Depuis, et jusqu'à présent, aucun algorithme n'a été trouvé pour le TSP métrique général avec une approximation meilleure que $\frac{3}{2}$. C'est pourquoi, plusieurs questions sont devenues le centre d'intérêt des spécialistes en approximation : Pouvons-nous faire mieux ? Pouvons-nous faire de même pour le TSP asymétrique ?

Toutefois, pour des cas particuliers du TSP métrique, des résultats significatifs ont été trouvés dont nous présentons l'essentiel.

En 1991, en se basant sur l'algorithme de Christofides, Hoogeveen [48] a généralisé ce résultat aux variantes chaîne du TSP et il a montré l'existence de $\frac{3}{2}$ -approximation pour le cas où au plus une extrémité de la chaîne est donnée et dans le même article, il a montré aussi l'existence d'un algorithme d'approximation polynomiale de rapport de $\frac{5}{3}$ lorsque les deux extrémités de la chaîne sont spécifiées.

En 1998, Guttmann beck *et al.*[45] ont proposé un algorithme similaire à celui de Hoogeveen pour la variante chaîne du TSP et ils ont trouvé un $\frac{5}{3}$ -approximation pour la version chaîne du TSP où le point de départ du voyageur est donné.

Récemment, l'exploration de Δ -Min TSP et Δ -Min PTSP a suscité un grand intérêt grâce à des résultats novateurs pour le cas graphique. Dans ce cas, en 2005, Gamarnik *et al.* [39] ont obtenu un rapport de $(\frac{3}{2} - \frac{5}{389})$ dans le cas où le graphe est cubique et 3-arête connexe. Puis, en 2010, leur résultat a été amélioré par Boyd *et al.* [14] qui ont trouvé un rapport de $\frac{4}{3}$ pour le TSP graphique dans le cas où le graphe est cubique.

Des progrès significatifs ont été accomplis en approximation du Min G -TSP et Min G -PTSP. En 2011, Oveis Gharan *et al.* [71] ont donné un algorithme avec un rapport d'approximation de $\frac{3}{2} - \epsilon$ pour le TSP. Par la suite, Mömke et Svensson [66] ont donné deux algorithmes d'approximation polynomiale, un qui obtient une solution de rapport nettement mieux de $\frac{14(\sqrt{2}-1)}{12\sqrt{2}-13} \approx 1.461$ pour le problème de Min- G TSP et un autre de rapport de $3 - \sqrt{2} + \epsilon \approx 1.587$, pour tout $\epsilon > 0$ pour le problème de Min- G PTSP dans le cas où les extrémités de la chaîne sont données.

En 2012, Mucha [70] a proposé un algorithme pour le Min- G TSP et a montré qu'il garantissait un rapport de $\frac{13}{9}$.

Puis, An *et al.* [3] ont amélioré le rapport de $\frac{5}{3}$ de la version chaîne du problème de Δ -Min TSP où les deux extrémités sont données, pour obtenir un rapport de $\frac{1+\sqrt{5}}{2} \approx .625$.

En 2013, Sebó [83] a amélioré le résultat pour Min G -TSP et il a obtenu un rapport de $\frac{8}{5}$.

Enfin, en 2014, Sebó et Vygen [84] ont trouvé un nouvel algorithme d'approximation pour Min- G TSP (resp., Min- G PTSP) de rapport $\frac{7}{5}$ (resp., $\frac{3}{2}$) dans le cas où le graphe est 2-arête connexe. Leur algorithme est basé sur l'utilisation des techniques de décomposition du graphe en oreilles.

2.6 Généralisation au problème du TSP multiple

Le Min- k TSP est une généralisation du problème de Min-TSP où k voyageurs de commerce doivent visiter un ensemble de sommets qui forment k cycles de telle manière

que tous les sommets du graphe soient visités.

De même, il existe aussi des versions chaîne du TSP multiple notées par : Min- k PTSP pour le cas symétrique et Δ -Min k PTSP pour le cas métrique.

Comme mentionné ci-dessus, le Min-TSP a reçu d'importantes études dans la littérature alors que le Min- k TSP et certaines de ses variantes restent inexplorés voir par exemple [11, 92, 51].

Frieze [37] a été parmi les premiers à étudier une variante de Min- k TSP qui est la version métrique (notée par Δ -Min k TSP). En 1983, il a présenté un algorithme d'approximation pour le problème de k voyageurs de commerce, qui est une extension de l'algorithme de Christofides et il a prouvé l'existence de $\frac{3}{2}$ -approximation dans le cas où k voyageurs doivent parcourir tous les sommets du graphe une seule fois en partant d'un même sommet v fixé en formant k cycles ayant au moins trois arêtes, de telle sorte que le coût total des cycles soit minimum.

En 2006, Rathinam *et al.*[78] ont proposé un algorithme d'approximation de rapport de 2 pour la version chaîne du TSP multiple Δ -Min k PTSP où k voyageurs doivent parcourir tous les sommets du graphe une seule fois partant chacun d'un sommet initial fixé, puis aller vers une destination finale non visitée.

En 2007, Rathinam et Sengupta[79] ont proposé un autre algorithme pour la version chaîne du TSP multiple de rapport de $\frac{5}{3}$ dans le cas métrique et les deux extrémités des chaînes sont fixées.

En 2010, Xu et Rodrigues [92] ont prouvé l'existence de $\frac{3}{2}$ -approximation pour Δ -Min k TSP où k voyageurs doivent parcourir k cycles en passant par tous les sommets du graphe une seule fois à partir de k extrémités initiales fixées et revenir chacun au sommet de départ.

En 2012, Bae et Rathinam [11] ont montré l'existence d'un algorithme d'approximation de rapport de $\frac{5}{3}$ pour la version chaîne du TSP multiple Δ -Min k PTSP où le point de départ des k voyageurs est le même et k destinations sont fixées.

Puis dans la même année, Rathinam et Sengupta [80] ont proposé un autre algorithme de rapport de $\frac{3}{2}$ pour le Δ -Min k PTSP en utilisant deux voyageurs ($k=2$) qui partent de deux extrémités initiales fixées (dépôts) et visitent tous les sommets du graphe une seule fois puis aller vers une destination finale non visitée. On note ce problème par 2DHPP (2 Depot Hamiltonian Path Problem).

2.7 Inapproximabilité

Comme mentionné dans la section 2.5, le TSP général est inapproximable à moins que $P=NP$, mais pour des cas particuliers de ce problème (variantes), nous avons vu qu'il existe des algorithmes d'approximation. Récemment, les variantes du TSP sont été étudiées du point de vue d'inapproximabilité où les chercheurs ont prouvé l'existence de bornes pour lesquelles elles sont inapproximables.

En 2006, Papadimitriou et Vempala [72] ont montré l'existence d'une borne d'inapproximabilité pour le TSP asymétrique qui est de $\frac{117}{116}$.

En 2012, Karpinski et Schmieid [53] ont étudié l'inapproximabilité du TSP graphique et ils ont montré qu'une borne de $\frac{535}{534}$ existe. Puis en 2013, ils ont étudié le cas où le graphe est cubique [54] et ils ont prouvé qu'il est $\frac{1153}{1152}$ -inapproximable.

En 2014, Lampis [59] a montré l'existence d'une borne d'inapproximabilité pour le TSP symétrique qui est de $\frac{185}{184}$.

Puis en 2015, Karpinski *et al.* [55] ont amélioré les bornes du TSP métrique et asymétrique et ils ont trouvé deux nouveaux résultats qui sont : $\frac{123}{122}$ pour le premier cas et $\frac{75}{74}$ pour le deuxième.

2.8 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur l'approximation du problème du voyageur de commerce et ses variantes. L'étude bibliographique que nous avons menée nous a permis d'une part, de voir les résultats obtenus en approximation de ces problèmes ainsi que les différentes méthodes utilisées pour concevoir des méthodes d'approximation, puis, de nous inspirer de quelques unes pour réaliser notre travail présenté dans cette thèse.

D'autre part, cette étude bibliographique nous a permis aussi de conclure que le problème du voyageur de commerce reste toujours d'actualité, étant donné la diversité de ses variantes et le nombre important de problèmes réels auxquels elles correspondent.

Chapitre 3

Un algorithme d'approximation pour le problème de la chaîne hamiltonienne à trois dépôts

3.1 Introduction

Nous proposons dans ce chapitre un algorithme d'approximation pour résoudre le problème de la chaîne hamiltonienne à trois dépôts. Le problème étudié peut être considéré comme une variante du problème de la chaîne hamiltonienne à plusieurs dépôts (cf. Demange [28] et Malik *et al.* [64]). Nous montrons l'existence d'un algorithme d'approximation de rapport de $\frac{3}{2}$. L'algorithme proposé est une extension de l'algorithme de Rathinam *et al.* [80] utilisé pour deux dépôts. Le résultat obtenu est établi lorsque les coûts sont symétriques et satisfont l'inégalité triangulaire.

3.2 Définition du problème

Soit $D = \{d_1, d_2, d_3\}$ un ensemble de trois dépôts distincts. $U = \{1, 2, 3, \dots, n\}$ un ensemble de n sommets avec ($n \geq 3$) et $V = D \cup U$. On affecte à chaque arête (i, j) joignant les sommets i et j , un coût $C_{ij} \in Q^+$ avec $C_{ij} = C_{ji}$ pour tout $i, j \in V$ et $C_{ik} \leq C_{ij} + C_{jk}$ pour tout $i, j, k \in V$ (inégalité triangulaire). Une chaîne parcourue par

un voyageur ℓ est une séquence de sommets $P_\ell = \{d_\ell, v_1^\ell, v_2^\ell, \dots, v_{k_\ell}^\ell\}$ où $\ell = 1, 2, 3$ et k_ℓ représente le nombre de sommets visités par le $\ell^{\text{ème}}$ voyageur et $\forall j \in \{1, 2, \dots, k_\ell\}$, $v_{\ell j} \in U$.

Le coût de chaque chaîne P_ℓ parcourue par le $\ell^{\text{ème}}$ voyageur est défini par :

$$C(P_\ell) = \begin{cases} C(d_\ell, v_1^\ell) + \sum_{j=1}^{k_\ell-1} C(v_j^\ell, v_{j+1}^\ell) & \text{si } k_\ell > 0, \text{ avec } \ell = 1, 2, 3. \\ 0 & \text{sinon,} \end{cases}$$

L'objectif du problème est de trouver les chaînes P_1 , P_2 et P_3 tel que :

1. Chaque destination de U soit visitée une seule fois par un seul voyageur.
2. Chaque voyageur visite au moins une destination.
3. La somme totale des coûts $\sum_{\ell=1}^3 C(P_\ell)$ soit minimum.

On note ce problème par 3DHPP (Three Depots Hamiltonian Path Problem).

3.3 Algorithme d'approximation pour le 3DHPP

Nous allons montrer que le problème du 3DHPP admet un algorithme d'approximation comprenant cinq étapes qui sont décrites de la manière suivante :

Algorithm 1 Trois chaînes hamiltoniennes

- 1- Relier les trois dépôts à un sommet w par trois arêtes de coût zéro et supprimer toutes les arêtes reliant ces dépôts.
Chercher un arbre de coût minimum puis supprimer les arêtes du coût zéro, ainsi nous obtenons une forêt F contenant trois arbres $T(d_1)$, $T(d_2)$ et $T(d_3)$ tel que :
 - (a) Chaque arbre contient un seul dépôt.
 - (b) Chaque dépôt est de degré 1.
 - (c) Chaque arbre contient un nombre impair de destinations de degré impair.

On note l'ensemble d'arêtes de la forêt par $E(F)$ et les destinations de degré impair dans chaque arbre de F par O_i pour $i = 1, 2, 3$.

- 2- Considérer le graphe $(V_0, E(V_0))$ où V_0 est l'ensemble des sommets de degré impair de la forêt F et $E(V_0)$ ensemble d'arêtes joignant deux sommets de V_0 . Comme $|O_1|$, $|O_2|$ et $|O_3|$ sont impairs alors :

$$|V_0| = |O_1| + |O_2| + |O_3| \text{ est impair.}$$

Trouver un couplage partiel M contenant un sous-ensemble d'arêtes $E(M) \subseteq E(V_0)$ de cardinal $\frac{|V_0|-3}{2}$ tel que le coût du couplage partiel soit minimum. On note les trois destinations non reliées par m_1 , m_2 et m_3 .

- 3- Ajouter les arêtes du couplage trouvées dans l'étape (2) à la forêt et relier les dépôts d_1 et d_2 par une arête du coût 0 ainsi que les destinations m_2 et m_3 .

On considère le nouveau graphe

$$G_n = ((V, E(F)) \cup E(M) \cup \{(d_1, d_2), (m_2, m_3)\}).$$

Le degré de chaque sommet de G_n est pair sauf les sommets m_1 et d_3 .

- 4- Trouver une chaîne eulérienne E_r dans le graphe G_n qui commence par m_1 et se termine par d_3 .
- 5- Appliquer le shortcutting à la chaîne eulérienne E_r pour avoir une chaîne hamiltonienne où chaque sommet est visité une seule fois, de la manière suivante : Pour tout sommet v de degré > 2 , considérer deux arêtes (u, v) et (v, w) consécutives dans E_r et les remplacer par l'arête (u, w) . On note cette chaîne par :

$$E_{sc} = (m_1, \dots, d_1, d_2, \dots, m_2, m_3, \dots, d_3)$$

Supprimer les arêtes du coût zéro dans E_{sc} pour obtenir les trois chaînes hamiltoniennes parcourues par les trois voyageurs de commerce.

Théorème 3.3.1. *L'algorithme présente une approximation de rapport de $\frac{3}{2}$ pour le problème de la chaîne hamiltonienne à 3 dépôts (3DHPP) avec une complexité de $O(n^3)$.*

Preuve . (a) *Montrons que la complexité de cet algorithme est de $O(n^3)$:*

Dans l'algorithme, on cherche une forêt, un couplage, une chaîne eulérienne et une chaîne hamiltonienne.

L'algorithme est dominé par deux étapes : trouver une forêt de coût minimum et un couplage partiel de coût minimum.

Trouver une forêt peut être vu comme étant un problème d'intersection de deux matroïdes, pour voir cela :

Soit V l'ensemble de tous les sommets et les trois dépôts ($|V| = n + 3$) et E un ensemble d'arêtes joignant deux sommets quelconques de V . Soit \mathcal{F}_1 une famille de sous ensembles de E tel que pour chaque $\mathcal{F} \in \mathcal{F}_1$, le graphe $G = (V, \mathcal{F}_1)$ est acyclique et il n'existe aucune chaîne reliant les trois dépôts dans G . $M_g = (E, \mathcal{F}_1)$ est un matroïde graphique.

Considérons \mathcal{F}_2 une famille de sous-ensembles de E tel que pour chaque $\mathcal{F} \in \mathcal{F}_2$, le degré de chaque dépôt dans $G = (V, \mathcal{F})$ est au plus égal à 1 et le nombre d'arêtes joignant chaque deux destinations dans G est au plus égal à $n - 3$. $M_p = (E, \mathcal{F}_2)$ est un matroïde de partition.

Pour chaque élément $B \in \mathcal{F}_1 \cap \mathcal{F}_2$, nous avons les propriétés suivantes :

- *B est acyclique.*
- *B ne contient pas de chaîne joignant les dépôts.*
- *le degré de chaque dépôt est au plus égal à 1 et le nombre d'arêtes joignant deux destinations est au plus $n - 3$.*

d'où B contient n arêtes, cette propriété implique que chaque dépôt est de degré 1, et le problème de la forêt de coût minimum peut être vu comme un problème d'intersection de deux matroïdes M_g et M_p , celui-ci peut être trouvé en $O(n^3)$ en utilisant l'algorithme de Brezovec [15].

Dans l'étape 2, on cherche un couplage partiel de coût minimum, cette étape peut être résolue en $O(n^3)$ en appliquant l'algorithme de Edmonds [33].

La complexité de l'étape(4)et(5) est de $O(n)$, d'où la complexité de l'algorithme est de $O(n^3)$.

(b) Montrons maintenant que l'approximation de cet algorithme est de $\frac{3}{2}$.

Cela revient à montrer que : $C(s) \leq \frac{3}{2}C(\text{opt})$ avec $C(s)$ est le coût de la solution produite par l'algorithme et $C(\text{opt})$ coût de la solution optimale.

D'un côté, on a la solution s majorée par le coût de la forêt F noté par $C(F)$ et le coût du couplage M noté par $C(M)$ car l'algorithme détermine une forêt et un couplage de coût minimum, puis raccourcir les chaînes si elles passent plusieurs fois par un même sommet, donc : $C(s) \leq C(F) + C(M)$.

D'un autre côté, on a $C(F) \leq C(\text{opt})$. Il suffit de montrer que le coût du couplage M est majoré par le coût de la solution optimale.

On considère une solution optimale (opt) de 3DHPP, cette solution contient trois chaînes disjointes P_1^* , P_2^* et P_3^* parcourues par les trois voyageurs de commerce 1,2,3 respectivement.

On définit $\mathcal{H}_\ell = \{v, v \in V_0, v \in P_\ell^*\}$ pour $\ell = 1, 2, 3$. Chaque sommet de \mathcal{H}_ℓ est un sommet visité par le $\ell^{\text{ème}}$ voyageur de commerce dans la solution optimale et aussi un sommet de degré impair dans la forêt de coût minimum obtenue par l'algorithme.

On a $\mathcal{H}_1 \cap \mathcal{H}_2 \cap \mathcal{H}_3 = \emptyset$ avec $\mathcal{H}_1 \cup \mathcal{H}_2 \cup \mathcal{H}_3 = V_0$ et $|\mathcal{H}_1| + |\mathcal{H}_2| + |\mathcal{H}_3| = |V_0|$ est impair. Raccourcir les arêtes de la solution optimale telle que les chaînes P_1 , P_2 et P_3 soient incidentes aux sommets de \mathcal{H}_ℓ pour $\ell = 1, 2, 3$. Nous montrons maintenant qu'on peut décomposer l'ensemble des sommets présents dans les chaînes en deux couplages partiels disjoints. On distingue les cas suivants :

Cas 1 : $|\mathcal{H}_1|$ impair, $|\mathcal{H}_2| = |\mathcal{H}_3| = 0$

La chaîne qui relie les sommets de \mathcal{H}_1 dans l'ordre de leur occurrence à d_1 est de coût inférieur ou égal au coût de la solution optimale et elle peut être décomposée en deux couplages partiels comme suit :

Soit $P_1 = \{u_1, u_2, \dots, u_{2p+1}\}$, $|P_1| = 2p + 1 \geq 5$, dans ce cas, on considère la décomposition

suivante : (cf. Figure 3.1)

$\mathbb{E}_1 = \{(u_2, u_3), (u_4, u_5), \dots, (u_{2p-4}, u_{2p-3}), (u_{2p-2}, u_{2p-1})\}$, \mathbb{E}_1 ne couvre pas les sommets u_1 , u_{2p} et u_{2p+1} .

$\mathbb{E}_2 = \{(u_3, u_4), (u_5, u_6), \dots, (u_{2p-1}, u_{2p})\}$, \mathbb{E}_2 ne couvre pas les sommets u_1 , u_2 et u_{2p+1} .

\mathbb{E}_1 et \mathbb{E}_2 sont deux couplages partiels disjoints contenant les sommets de V_0 de cardinal $\frac{|V_0|-3}{2}$, d'où : $2C(M) \leq C(\mathbb{E}_1) + C(\mathbb{E}_2)$ et comme les arêtes de \mathbb{E}_1 et \mathbb{E}_2 sont disjointes et sont obtenues après avoir éliminé quelques sommets de la solution optimale d'où le résultat suivant :

$$2C(M) \leq C(\mathbb{E}_1) + C(\mathbb{E}_2) \leq C(opt)$$

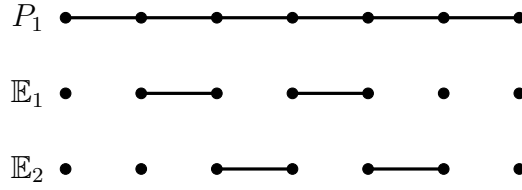


FIG. 3.1 – Décomposition des sommets de V_0 en deux couplages disjoints, Cas 1 : $|\mathcal{H}_1|$ impair, $|\mathcal{H}_2| = |\mathcal{H}_3| = 0$

Cas 2 : $|\mathcal{H}_1|$ impair, $|\mathcal{H}_2|$ pair et $|\mathcal{H}_3| = 0$

Même décomposition en deux couplages (cf. Figure 3.2).

Soient $P_1 = \{u_1, u_2, \dots, u_{2p+1}\}$, $P_2 = \{v_1, v_2, \dots, v_{2k}\}$. Dans ce cas, on considère deux ensembles disjoints $\mathbb{E}_1, \mathbb{E}_2$ avec :

$\mathbb{E}_1 = \{(u_1, u_2), (u_3, u_4), \dots, (u_{2p-1}, u_{2p})\} \cup \{(v_1, v_2), (v_3, v_4), \dots, (v_{2k-1}, v_{2k})\}$, \mathbb{E}_1 ne couvre pas les sommets u_{2p+1} , v_1 et v_{2k} .

$\mathbb{E}_2 = \{(u_2, u_3), (u_4, u_5), \dots, (u_{2p}, u_{2p+1})\} \cup \{(v_2, v_3), (v_4, v_5), \dots, (v_{2k-2}, v_{2k-1})\}$, \mathbb{E}_2 ne couvre pas les sommets u_1 , v_1 et v_{2k} .

\mathbb{E}_1 et \mathbb{E}_2 sont deux couplages disjoints de cardinal $\frac{|V_0|-3}{2}$, chacun de coût inférieur au coût du couplage partiel M .

Cas 3 : $|\mathcal{H}_1|$ impair, $|\mathcal{H}_2|$ et $|\mathcal{H}_3|$ sont pairs

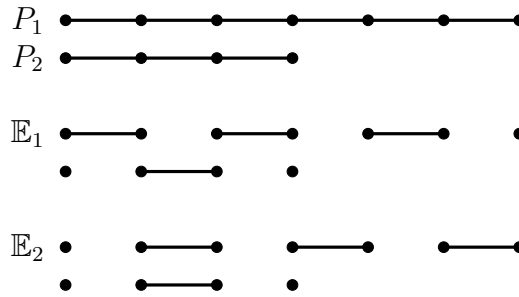


FIG. 3.2 – Décomposition des sommets de V_0 en deux couplages disjoints, Cas 2 : $|\mathcal{H}_1|$ impair, $|\mathcal{H}_2|$ pair et $|\mathcal{H}_3| = 0$

Même décomposition, en deux couplages : l'un qui laisse un sommet de P_1 non couplé, deux sommets de P_3 non couplés et l'autre qui laisse un de P_1 et deux de P_2 non couplés, (cf. Figure 3.3).

Soient $P_1 = \{u_1, u_2, \dots, u_{2p+1}\}$, $P_2 = \{v_1, v_2, \dots, v_{2k}\}$ et $P_3 = \{w_1, w_2, \dots, w_{2r}\}$, dans ce cas, on considère deux ensembles disjoints \mathbb{E}_1 et \mathbb{E}_2 avec :

$$\mathbb{E}_1 = \{(u_1, u_2), (u_3, u_4), \dots, (u_{2p-1}, u_{2p})\} \cup \{(v_1, v_2), (v_3, v_4), \dots, (v_{2k-1}, v_{2k})\} \cup$$

$\{(w_2, w_3), (w_4, w_5), \dots, (w_{2r-2}, w_{2r-1})\}$. \mathbb{E}_1 ne couvre pas les sommets u_{2p+1} , w_1 et w_{2r} .

$$\mathbb{E}_2 = \{(u_2, u_3), (u_4, u_5), \dots, (u_{2p}, u_{2p+1})\} \cup \{(v_2, v_3), (v_4, v_5), \dots, (v_{2k-2}, v_{2k-1})\} \cup$$

$\{(w_1, w_2), (w_3, w_4), \dots, (w_{2r-1}, w_{2r})\}$. \mathbb{E}_2 ne couvre pas les sommets u_1 , v_1 et v_{2k} .

\mathbb{E}_1 et \mathbb{E}_2 sont deux couplages disjoints de cardinal $\frac{|V_0|-3}{2}$, chacun de coût inférieur au coût du couplage partiel M .

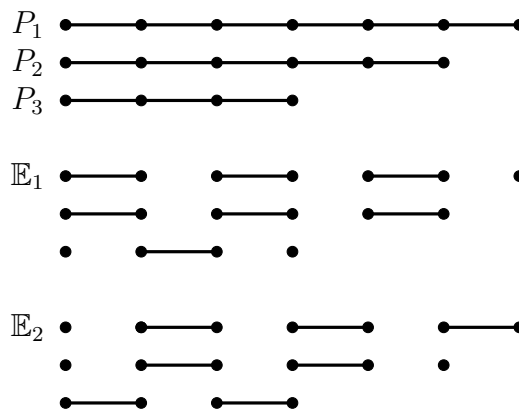


FIG. 3.3 – Décomposition des sommets de V_0 en deux couplages disjoints, Cas 3 : $|\mathcal{H}_1|$ impair, $|\mathcal{H}_2|$ et $|\mathcal{H}_3|$ sont pairs

Cas 4 : $|\mathcal{H}_1|$, $|\mathcal{H}_2|$ et $|\mathcal{H}_3|$ sont impairs

Même décomposition, en deux couplages : Chacun des P_i , $i = 1, 2, 3$ laisse un sommet non couplé. (cf. Figure 3.4).

Soient $P_1 = \{u_1, u_2, \dots, u_{2p+1}\}$, $P_2 = \{v_1, v_2, \dots, v_{2k+1}\}$ et $P_3 = \{w_1, w_2, \dots, w_{2r+1}\}$, dans ce cas on considère deux ensembles disjoints $\mathbb{E}_1, \mathbb{E}_2$ avec :

$$\mathbb{E}_1 = \{(u_1, u_2), (u_3, u_4), \dots, (u_{2p-1}, u_{2p})\} \cup \{(v_1, v_2), (v_3, v_4), \dots, (v_{2k-1}, v_{2k})\} \cup \{(w_1, w_2), (w_3, w_4), \dots, (w_{2r-1}, w_{2r})\}$$

\mathbb{E}_1 ne couvre pas les sommets u_{2p+1} , v_{2k+1} et w_{2r+1} .

$$\mathbb{E}_2 = \{(u_2, u_3), (u_4, u_5), \dots, (u_{2p}, u_{2p+1})\} \cup \{(v_2, v_3), (v_4, v_5), \dots, (v_{2k}, v_{2k+1})\} \cup \{(w_2, w_3), (w_4, w_5), \dots, (w_{2r}, w_{2r+1})\}$$

\mathbb{E}_2 ne couvre pas les sommets u_1 , v_1 et w_1 .

\mathbb{E}_1 et \mathbb{E}_2 sont deux couplages disjoints de cardinal $\frac{|V_0|-3}{2}$, chacun de coût inférieur au coût du couplage partiel, d'où

$$2C(M) \leq C(\mathbb{E}_1) + C(\mathbb{E}_2) \leq C(opt)$$

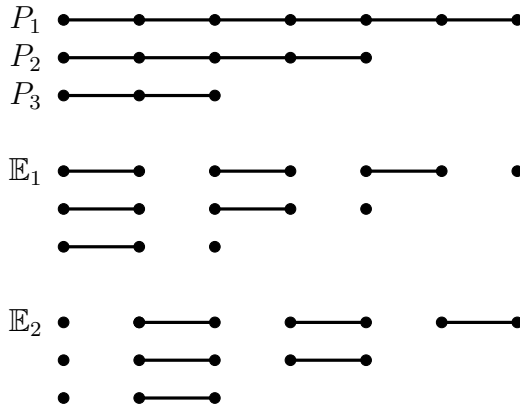


FIG. 3.4 – Décomposition des sommets de V_0 en deux couplages disjoints, Cas 4 : $|\mathcal{H}_1|$, $|\mathcal{H}_2|$ et $|\mathcal{H}_3|$ sont impairs

D'après ce qui précède, on a :

$$2C(M) \leq C(E_1) + C(E_2) \leq C(opt) \iff C(M) \leq \frac{1}{2}C(opt) \quad (1)$$

$$C(F) \leq C(opt) \quad (2)$$

de (1) et (2) on obtient : $C(S) \leq C(M) + C(F) \leq (\frac{1}{2} + 1)C(opt) = \frac{3}{2}C(opt)$.

3.4 Exemple illustratif

Dans cette section, nous montrons comment les différentes étapes de l'algorithme peuvent être appliquées pour construire la solution finale de l'algorithme d'approximation.

Dans la première étape, la figure 3.5 montre les trois arbres qui forment la forêt obtenue et la façon dont chacun de ces arbres est relié à l'un des dépôts (d_1, d_2, d_3).

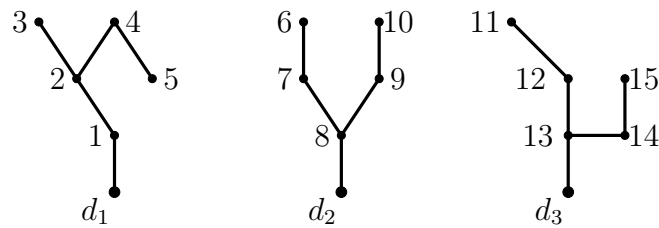


FIG. 3.5 – Étape 1 : (a) La forêt de coût minimum

L'étape 2 sert à trouver les sommets de degré impair pour la recherche du couplage partiel de coût minimum, (cf. Figure 3.6).

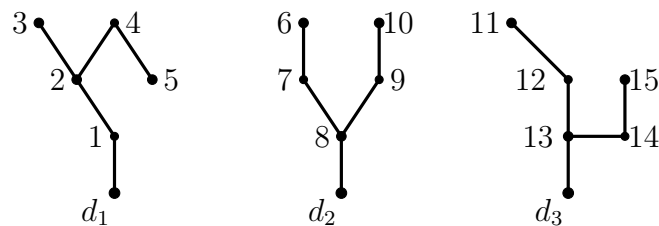


FIG. 3.6 – Étape 1 : (b) En rouge, les destinations de degré impair, $|V_0| = |O_1 \cup O_2 \cup O_3| = 9$

La figure 3.7 illustre le couplage partiel trouvé en utilisant les sommets de degré impair.

Après les deux étapes 3 et 4, nous pouvons observer (cf. Figure 3.8) la façon dont nous utilisons l'ajout des arêtes pour obtenir la chaîne eulérienne E_r .

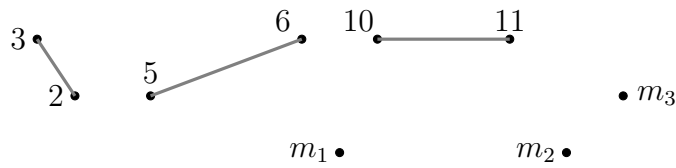


FIG. 3.7 – Étape 2 : Le couplage partiel de coût minimum de cardinal $\frac{|V_0|-3}{2} = 3$

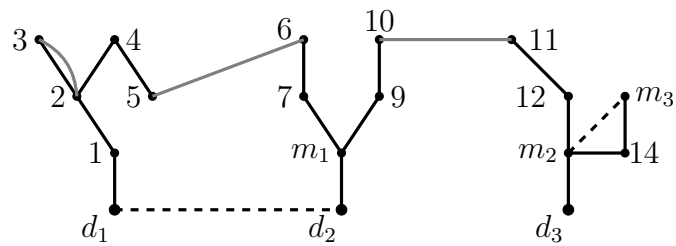


FIG. 3.8 – Étape 3 et 4 : Ajout des arêtes à la forêt pour obtenir une chaîne eulérienne $E_r = \{m_1, 7, 6, 5, 4, 2, 3, 2, 1, d_1, d_2, m_1, 9, 10, 11, 12, m_2, m_3, 14, m_2, d_3\}$

En appliquant la méthode du shortcutting de l'étape 5, nous pouvons observer (cf. Figure 3.9) la nouvelle chaîne hamiltonienne E_{sc} obtenue à partir de la chaîne eulérienne E_r .

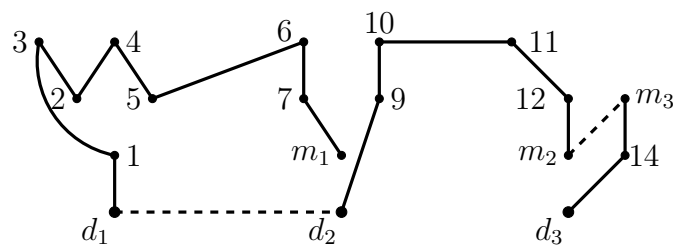


FIG. 3.9 – Étape 5 : Suppression des arêtes pour obtenir une chaîne hamiltonienne $E_{sc} = \{m_1, 7, 6, 5, 4, 2, 3, 1, d_1, d_2, 9, 10, 11, 12, m_2, m_3, 14, d_3\}$

Enfin, selon les trois derniers points de l'étape 5, la figure 3.10 illustre la solution finale fournie par l'algorithme.

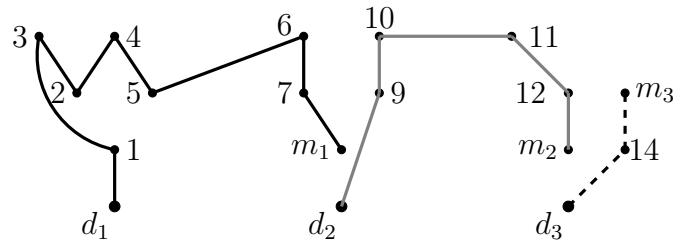


FIG. 3.10 – Étape 5 : Suppression des arêtes de coût 0 pour obtenir les trois chaînes hamiltoniennes $\{d_1, 1, 3, 2, 4, 5, 6, 7, m_1\}$, $\{d_2, 9, 10, 11, 12, m_2\}$, $\{d_3, 14, m_3\}$

3.5 Conclusion

Nous considérons le problème de la chaîne hamiltonienne à trois dépôts (3DHP) pour lequel, il n'existe pas d'algorithme d'approximation meilleur que 2 et nous présentons un algorithme polynomial de complexité de $O(n^3)$ avec un rapport d'approximation de $\frac{3}{2}$ dans le cas où les coûts sont symétriques et vérifient l'inégalité triangulaire.

Chapitre 4

Algorithmes d'approximation pour le k DHPP et le TSP graphique

4.1 Introduction

Dans ce chapitre, nous considérons à la fois le problème de la recherche de k chaînes hamiltoniennes dans un graphe à k dépôts fixes (noté par k DHPP) et le problème de la recherche du plus court cycle hamiltonien (noté par Min-GTSP). Premièrement, nous étudions le problème de k DHPP dans un graphe cubique et 2-sommet connexe, pour lequel nous montrons l'existence d'un algorithme d'approximation de rapport de $\frac{5}{3}$ qui donne une solution avec au plus $(\frac{5}{3}n - \frac{4k-2}{3})$ arêtes. La méthode proposée est basée sur la recherche d'un couplage parfait utilisant un processus de suppression des arêtes. Deuxièmement, nous proposons un algorithme d'approximation pour le Min-GTSP où le graphe est facteur critique et 2-sommet connexe. L'algorithme proposé réalise un rapport d'approximation de $\frac{7}{6}$ meilleur que celui de *Sebo* et *Vygen* [84] qui est de $\frac{7}{5}$ où le principe de la méthode est basé sur la décomposition du graphe en une série d'oreilles.

4.2 Notations et définitions

Nous décrivons ici quelques notations et définitions utilisées dans ce chapitre.

Définition 4.2.1. Soit $G = (V, E)$ un graphe connexe sans boucles, tel que V est un

ensemble de n sommets et E un ensemble de m arêtes. Les coûts sur les arêtes sont définis comme suit : Pour tout $(i, j) \in V^2$, $C(i, j)$ est la longueur (nombre d'arêtes) de la plus courte chaîne reliant i à j dans G et $C(i, i) = 0, \forall i \in V$ et $C(i, j) = C(j, i), \forall (i, j) \in V^2$

Définition 4.2.2. Dans un graphe connexe et non orienté G , une oreille est une chaîne dont les deux extrémités de la chaîne peuvent coïncider et dont les sommets internes sont de degré deux dans l'oreille.

Une décomposition en oreilles d'un graphe G est une partition de l'ensemble de ses arêtes en séquence d'oreilles \mathcal{P}_i telle que : $G = \mathcal{P}_0 + \mathcal{P}_1 + \dots + \mathcal{P}_k$ où \mathcal{P}_0 est une chaîne triviale contenant un seul sommet et chaque $\mathcal{P}_i (i = 1, 2, \dots, k)$ est une chaîne ayant ses extrémités dans $V_{i-1} = V(\mathcal{P}_0) \cup V(\mathcal{P}_1) \cup \dots \cup V(\mathcal{P}_{i-1})$ et les sommets internes de $\mathcal{P}_i \notin V_{i-1}$

Par convention \mathcal{P}_0 est pair.

- Une oreille est dite fermée si ses extrémités coïncident, sinon elle est dite ouverte.
- Une oreille est dite paire si elle contient un nombre pair d'arêtes et elle est impaire si le nombre d'arêtes est impair.
- Si dans une décomposition en oreilles, toutes les oreilles sont paires, on dit que c'est une décomposition en oreilles paire, de manière similaire, on définit la décomposition impaire.
- Une oreille est dite courte si le nombre d'arêtes qu'elle contient est soit 2 ou 3.
- Une oreille contenant une seule arête est dite triviale.
- Une oreille pendante est une oreille non triviale et aucune autre oreille non triviale n'est adjacente à ses sommets internes.

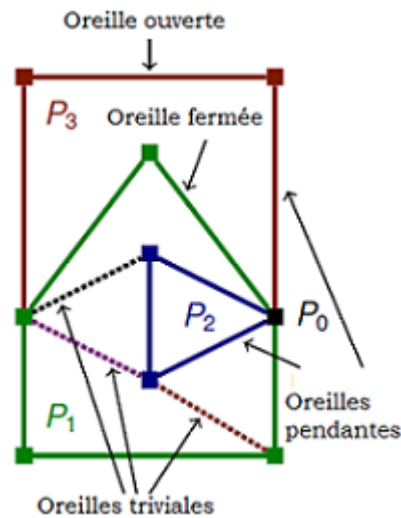


FIG. 4.1 – Une décomposition en oreilles

Définition 4.2.3. Un graphe G à n sommets (n impair) est dit facteur critique si chaque sous graphe de $n - 1$ sommets de G admet un couplage parfait.

Proposition 1. [84] Un graphe G est dit facteur critique si et seulement s'il admet une décomposition en oreilles impaire.

Définition 4.2.4. Soit G un graphe facteur critique . Une décomposition en oreilles de G est dite bonne (nice) si :

- (i) le nombre d'oreilles paires est égal à $\varphi(G)$, avec $\varphi(G)$ le nombre minimum d'oreilles paires dans une décomposition en oreilles de G .
- (ii) toutes les oreilles courtes de G sont pendantes,
- (iii) tous les sommets internes des différentes oreilles courtes sont non-adjacentes dans G .

Définition 4.2.5. 1. Soit F la forêt obtenue par l'application de DFS_F (Depth first search) d'une forêt sur le graphe G . Chaque arête e appartenant à F est appelée "arête-arbre", et toutes les autres arêtes appartenant à $G \setminus F$ sont appelées "arête-retour".

2. Une arête e du graphe k -connexe $G = (V, E)$ est dite arête amovible si le graphe

$G(V, E \setminus \{e\})$ noté par $G - e$ reste k -connexe.

3. Soit v un sommet de F qui n'est pas un sommet racine. Une paire d'arêtes (arête-retour, arête-arbre) incidente à v est appelée amovible. (Une seule arête sera amovible).

4.3 Problème de la chaîne hamiltonienne à k dépôts

Soit $G = (V, E)$ un graphe cubique et 2-sommet connexe. $D \subset V$ un ensemble de k dépôts d_i , $i = 1, \dots, k$ et $U = V \setminus D$, $U = \{1, 2, \dots, n\}$ un ensemble de destinations (sommets), ($2 \leq k \leq n$). On définit sur G la métrique m_G . Une solution P_ℓ , $1 \leq \ell \leq k$, est représentée par une séquence ordonnée de sommets :

$$P_\ell = (d_\ell, v_1^\ell, v_2^\ell, \dots, v_{k_\ell}^\ell),$$

Où $\ell = 1, 2, \dots, k$, k_ℓ est le nombre de sommets visités par le ℓ -ème voyageur et $\forall j \in \{1, \dots, k_\ell\}$, $v_{\ell j} \in U$. Le coût des chaînes P_ℓ parcourues par le ℓ -ème voyageur est défini comme suit :

$$C(P_\ell) = \begin{cases} C(d_\ell, v_1^\ell) + \sum_{j=1}^{k_\ell-1} C(v_j^\ell, v_{j+1}^\ell) & \text{if } k_\ell > 0 \\ 0 & \text{sinon,} \end{cases}$$

L'objectif du problème est de trouver toutes les chaînes P_1, P_2, \dots, P_k telles que :

1. Chaque destination de U soit visitée une seule fois par un voyageur,
2. Chaque voyageur visite au moins une destination,
3. La somme totale des coûts $\sum_{\ell=1}^k C(P_\ell)$ soit minimum.

On note ce problème par k DHPP (k -Depots Hamiltonian Path Problem).

4.3.1 Quelques résultats

Avant de décrire l'algorithme proposé, nous établissons d'abord quelques résultats qui seront utilisés par la suite.

Proposition 2. *Soit G un graphe cubique et 2-sommet connexe. Alors, le graphe obtenu par suppression d'arêtes amovibles du couplage M est eulérien.*

Preuve . *Soit $G = (V, E)$ un graphe 2-sommet connexe et cubique et nous supposons que $e \in G$ est une arête amovible de G .*

D'après Monma et al. [86], nous avons :

$$\forall e \in E, e \in M \text{ avec une probabilité de } \frac{1}{3} .$$

c'est à dire qu'il y a au plus une arête de chaque paire amovible dans le couplage parfait M . D'où, $\forall v \in V$, le sommet v est de degré pair dans le multi graphe $(V, E \cup M)$.

□

Proposition 3. *Le graphe obtenu en éliminant les arêtes amovibles d'un graphe cubique et 2-arête connexe dont au plus une arête de chaque paire est retirée, est connexe.*

Preuve . *Soit $M \subseteq E$ un sous-ensemble d'arêtes qui forment un couplage parfait, soient M_1 et M_2 deux sous-ensembles de M tels que :*

M_1 contient les arêtes amovibles et M_2 les arêtes restantes de M .

Considérons le graphe $H = (V, E \setminus M_1 \cup M_2)$. Observons que l'ajout et la suppression d'une arête permute la parité du degré d'un sommet incident. Nous avons d'après la proposition 2, le degré de chaque sommet dans H est pair. En outre, si (R, R') est une paire amovible, au plus une arête dans chaque paire est dans M , d'où H est connexe. □

4.3.2 Algorithme d'approximation pour le k DHPP

L'algorithme proposé 4.3.2 noté par A_{kD} contient quatre étapes :

Théorème 4.3.1. *Soit G un graphe cubique et 2-sommet connexe. Alors, l'algorithme A_{kD} donne une solution avec au plus $\frac{5}{3}n - \frac{4k-2}{3}$ arêtes pour le k DHPP avec une complexité de $O(n^{2.5})$.*

Preuve . *Nous montrons d'abord que l'algorithme A_{kD} réalise un rapport d'approximation de $\frac{5}{3}$ et donne une solution avec au plus $\frac{5}{3}n - \frac{4k-2}{3}$ arêtes. Puis, nous montrons que*

Algorithm 2 Déterminer k chaînes hamiltoniennes (A_{kD}).

Entrée : Un graphe cubique et 2-sommet connexe G .

Sortie : k chaînes hamiltoniennes.

Soit $D = \{d_1, d_2, d_3, \dots, d_k\}$ les k dépôts.

1. **Construction de la forêt.**

Soit F la forêt contenant k arbres obtenue par l'application de la méthode DFS_F à G .

2. **Obtention du couplage.**

Chercher un couplage parfait aléatoire M dans G suivant la proposition 2.

3. **Obtention d'un graphe eulérien.**

Appliquer la proposition 3 comme suit :

(a) $\forall e \in M$ où e appartient à la paire amovible, supprimer e de G .

(b) $\forall e \in M$ tel que e n'appartient pas à la paire amovible, ajouter e à G où une double arête est créée. Nous obtenons ainsi un graphe eulérien.

4. **Obtention de k chaînes hamiltoniennes.**

Appliquer le shortcutting au graphe obtenu à l'étape 3 pour avoir un graphe hamiltonien puis couper $k - 1$ arêtes pour avoir les k chaînes hamiltoniennes.

sa complexité est d'ordre de $O(n^3)$.

1) Soit H un ensemble contenant toutes les arêtes-arbre et arêtes-retour. Puisque $G = (V, E)$ est cubique, donc $|E| = \frac{3}{2}n$. Considérons b le nombre d'arêtes-retour de G . Alors,

$$b = |E| - (n - k) \quad \iff \quad b = \frac{3}{2}n - n + k = \frac{1}{2}n + k,$$

Où k désigne le nombre d'arbres qui forment la forêt (nombre de dépôts). Comme toutes les arêtes amovibles sont des paires, donc pour chaque arête- retour, on a exactement une autre arête amovible sauf celles reliant les dépôts d'où :

$$|H| = 2b - (k - 1) = 2\left(\frac{1}{2}n + k\right) - k + 1 = n + k + 1.$$

Donc, nous avons $\frac{n+k+1}{2}$ arêtes amovibles et les autres sont inamovibles, d'où le nombre d'arêtes inamovibles est :

$$\frac{3}{2}n - \frac{n + k + 1}{2} = \frac{2n - k - 1}{2}.$$

D'autre part, chaque arête peut être dans le couplage parfait choisi avec une probabilité

de $\frac{1}{3}$. Si elle est dans le couplage, soit c'est une arête amovible ou inamovible. D'où : le nombre d'arêtes de la chaîne eulérienne obtenue est égal à :

$$\text{arêtes de } G + \text{arêtes inamovible} * \text{prob} \in M - \text{arêtes amovibles} * \text{prob} \in M.$$

$$\frac{3}{2}n + \frac{1}{3}\left(\frac{2n - k - 1}{2}\right) - \frac{1}{3}\left(\frac{n + k + 1}{2}\right) = \frac{5}{3}n - \frac{k + 1}{3}.$$

Soit \mathcal{S} le nombre d'arêtes des chaînes hamiltoniennes obtenues. Alors,

$$\mathcal{S} \leq \frac{5}{3}n - \frac{k + 1}{3} - (k - 1) = \frac{5}{3}n - \frac{4k - 2}{3}.$$

2) Nous avons aussi mentionné que l'algorithme A_{kD} est basé sur quatre étapes : génération de F en utilisant la recherche DFS_F dans le graphe G , recherche d'un couplage, construction d'un graphe eulérien et enfin obtention de k chaînes hamiltoniennes.

D'un côté, la complexité de DFS appliquée au graphe $G = (V, E)$ est d'ordre de $O(m + n)$, comme G est cubique, cette complexité devient $O(kn)$. D'un autre côté, le couplage parfait dans le graphe G peut être obtenu en $O(n^{2.5})$ (cf., Micali and Vazirani [65]).

Enfin, la complexité de la recherche d'un graphe eulérien est de $O(n)$ et comme l'algorithme est dominé par la recherche d'un couplage parfait, la complexité de l'algorithme est égale à $O(n^{2.5})$ □

4.4 Exemple pour $k = 3$

Dans cette section, nous montrons comment appliquer les différentes étapes de l'algorithme A_{kD} pour construire k chaînes hamiltoniennes où $k = 3$ dans un graphe G décrit dans la figure 4.2 (une instance composée de trois dépôts).

D'après la première étape de l'algorithme A_{kD} , la figure 4.3.(a) montre la forêt (qui est représentée par les traits gras) qui contient trois arbres obtenus par l'application de DFS_F au graphe G de la figure 4.2. Chacun de ses trois arbres est relié à un sommet

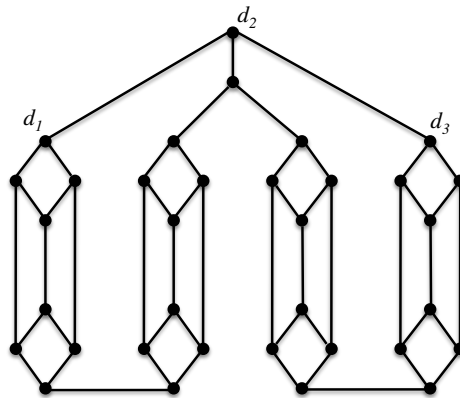


FIG. 4.2 – Une instance du graphe G .

racine, i.e., d_1 , d_2 et d_3 , respectivement. La figure 4.3.(b) montre les trois couplages parfaits obtenus en appliquant l'étape 2 de A_{kD} .

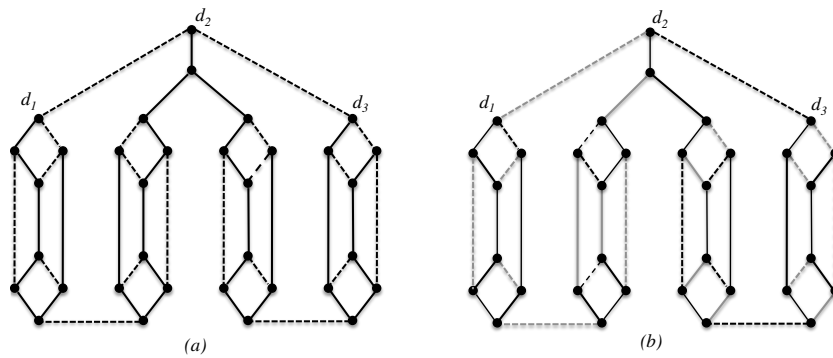


FIG. 4.3 – Illustration de l'étape 1 (à gauche, notée(a)) et de l'étape 2 (à droite, notée (b)) de l'algorithme A_{kD} .

(a). Dans ce cas, on s'intéresse à un couplage parfait qui est obtenu par l'application de la proposition 3 sur la forêt de la figure 4.3. (a). Selon l'étape 3 de l'algorithme A_{kD} , la figure 4.4.(a) montre les arêtes amovibles du couplage parfait supprimées afin de parvenir à un graphe eulérien. Ensuite, par l'application de la stratégie dite shortcutting (étape 4 de A_{krmD}), on obtient le graphe hamiltonien (4.4.(b)).

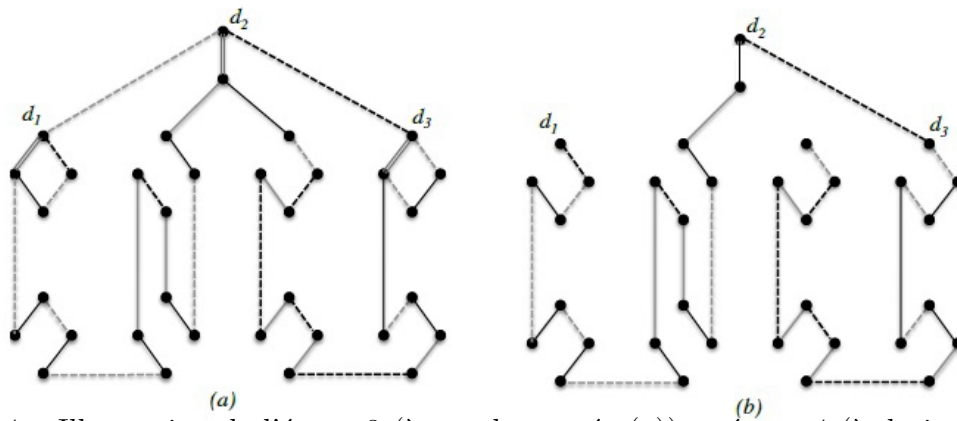


FIG. 4.4 – Illustration de l'étape 3 (à gauche, notée (a)) et étape 4 (à droite, notée (b)) de l'algorithme A_{kD} sur le dernier graphe résultant de la Fig. 4.3

Finalement, en coupant $k - 1$ arêtes (étape 4 de A_{kD}), on obtient les trois chaînes hamiltoniennes de la solution finale (comme illustré dans la figure 4.5).

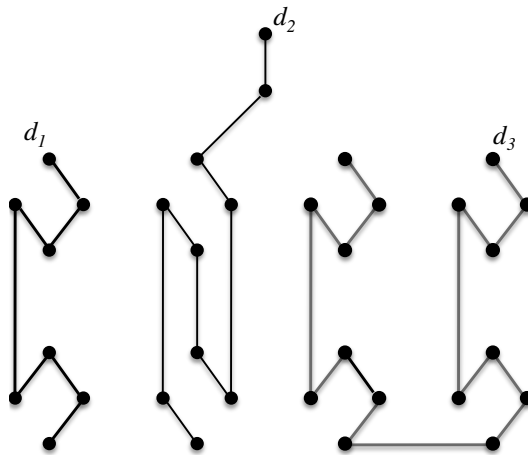


FIG. 4.5 – La solution finale obtenue par l'algorithme A_{kD} .

4.5 Le TSP graphique dans un graphe facteur critique et 2-sommet connexe

Soit $G(V, E)$ un graphe facteur critique et 2-sommet connexe. L'objectif du problème est de trouver le plus court cycle hamiltonien dans le graphe G . On note ce problème par Min-GTSP

4.5.1 Un algorithme d'approximation

L'idée de la méthode proposée consiste à décomposer l'instance du problème en oreilles à partir d'un sommet arbitraire. On note cet algorithme par A_{HT} .

Algorithm 3 : Algorithme d'approximation A_{HT} pour le graphe facteur critique

Entrée : G un graphe 2-sommet connexe et facteur critique.

Sortie : Le plus court cycle hamiltonien.

1. Déterminer une bonne décomposition en oreilles impaires.
 2. Supprimer toutes les oreilles triviales : oreilles contenant une seule arête.
 3. Corriger la parité des sommets en ajoutant une arête entre chaque deux sommets de degré impair pour obtenir un graphe dont tous les sommets sont de degré pair.
 4. Chercher un graphe eulérien sur le graphe obtenu.
 5. Appliquer le shortcutting pour le graphe eulérien pour avoir un cycle hamiltonien comme suit :
 - (a) Soit (u, z) une chaîne représentée comme suit : (u, v, w, u, z) obtenu dans le graphe eulérien où le sommet u est répété deux fois.
 - (b) Comme G est un graphe métrique, les coûts des arêtes vérifient les inégalités triangulaires i.e., $C(w, u) + C(u, z) \geq C(w, z)$. Puis, supprimer toutes les arêtes (w, u) et (u, z) du graphe eulérien et les remplacer par (w, z) .
 - (c) Sur tous les sommets, répéter le processus de la suppression et remplacement de l'étape 5b jusqu'à obtention d'un cycle hamiltonien.
-

Théorème 4.5.1. *Soit G un graphe non orienté, sans boucles, 2-sommet connexe et facteur critique. L'algorithme A_{HT} donne une solution de rapport d'approximation de $\frac{7}{6}$ pour le Min-GTSP de complexité de $O(nm)$.*

Avant de démontrer le théorème ci-dessus, nous définissons la relaxation linéaire suivante :

Soit $G = (V, E)$ un graphe non orienté sans boucles. La relaxation linéaire du problème du sous-graphe 2-arête connexe(2ECSS) est définie comme suit :

$$L_\varphi(G) = \min \{x(E) : x \in \mathbb{R}_{\geq 0}^E, x(\delta(W)) \geq 2 \text{ pour chaque } \emptyset \neq W \subset V\}$$

Où $x \subseteq V$, $\delta(x)$ est un ensemble d'arêtes ayant une seule extrémité dans x . x_e sont des variables de décision pour tout $e \in E$ de valeurs dans $[0, 1]$. Pour $S \in E$,

$$x(S) = \sum_{e \in S} x_e.$$

Preuve . *Nous montrons d'abord que l'algorithme A_{HT} donne une solution de rapport d'approximation de $\frac{7}{6}$. Puis, nous montrons que la complexité de l'algorithme est de $O(nm)$.*

1) De l'étape 1 de l'algorithme A_{HT} , soit \mathcal{K} le nombre d'oreilles dans une décomposition. D'après les résultats de Whitney 1932 [90], $\mathcal{K} = |E(G)| - n + 1$ [90].

D'après les résultats de Sebö et al. [84], une bonne décomposition en oreilles dans un graphe 2-sommet connexe G donne un cycle de cardinal au plus égal à $\frac{4}{3}(|n - 1) + \frac{2}{3}\pi$; obtenu par la suppression des oreilles triviales et de complexité de $O(n)^3$; Où π désigne le nombre d'oreilles pendantes dans la décomposition.

On sait aussi que le graphe G est 2-arête connexe, d'où d'après les résultats de Cheriyan et al. [19], la valeur de la relaxation du Min 2ECSS $L_\varphi(G)$ est :

$$L_\varphi(G) = n + \varphi(G) - 1 \quad \Longleftrightarrow \quad L_\varphi(G) \leq LP(G).$$

$LP(G)$ est la valeur de la solution optimale du programme linéaire du TSP dans G . Comme G est décomposé en oreilles formant une bonne décomposition, alors $\varphi(G) = 0$ [63]. D'un côté, le graphe G est 2-arête connexe; il s'ensuit qu'on peut trouver en temps polynomial un cycle de cardinal au plus égal à $L_\varphi(G)$ [19] d'où :

$$(n - 1) \leq LP(G) \tag{4.1}$$

D'un autre côté, le graphe G est 2-sommet connexe; il s'ensuit qu'il existe un cycle de cardinal au plus égal à

$$\frac{4}{3}(n - 1) + \frac{2}{3}\pi \leq \frac{4}{3}LP(G) \tag{4.2}$$

En combinant les équations (4.1) et (4.2), nous déduisons que l'algorithme A_{HT} trouve un cycle hamiltonien de cardinal au plus égal à :

$$\frac{7}{6}(n - 1) + \frac{1}{3}\pi \leq \frac{7}{6}LP(G) \tag{4.3}$$

Pour tout graphe connexe G , Sebö et al. [84] ont établi les inégalités suivantes :

$$Opt(G) \geq LP(G). \tag{4.4}$$

D'après les équations (4.3) et (4.4), nous déduisons que l'algorithme A_{HT} donne une solution de cardinal au plus égal à :

$$\frac{7}{6}(n-1) + \frac{1}{3}\pi \leq \frac{7}{6}LP(G) \leq \frac{7}{6}Opt(G).$$

2) L'algorithme A_{HT} est basé sur trois étapes : une bonne décomposition en oreilles du graphe G , la recherche d'un graphe eulérien puis la recherche du cycle hamiltonien. Cependant, d'une part, pour ces types de graphes (facteur critique) Sebö et al. [84] ont montré l'existence d'un algorithme polynomial pour la recherche d'une bonne décomposition impaire en oreilles de complexité de $O(nm)$. D'un autre côté, la complexité de recherche d'un cycle eulérien ou d'un cycle hamiltonien est $O(n)$. D'où la complexité de A_{HT} est de $O(nm)$. \square

4.5.2 Application de l'algorithme A_{HT} à un exemple

Dans cette section, nous montrons comment les différentes étapes de l'algorithme A_{HT} sont appliquées pour avoir un cycle hamiltonien.

La figure 4.6 représente une instance composée de treize sommets.

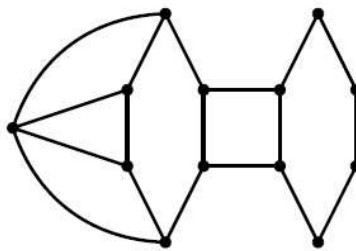


FIG. 4.6 – Une instance du problème.

D'après la première étape de l'algorithme A_{HT} , la figure 4.7.(a) montre comment le graphe G est décomposé en oreilles $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_6$ formant une bonne décomposition. Nous observons que dans la figure 4.7(a), les trois oreilles suivantes $\mathcal{P}_3, \mathcal{P}_4$ et \mathcal{P}_6 (en pointillés) sont triviales. Puis, dans l'étape (2) de A_{HT} , on supprime ces trois oreilles pour obtenir le graphe de la figure 4.7.(b).

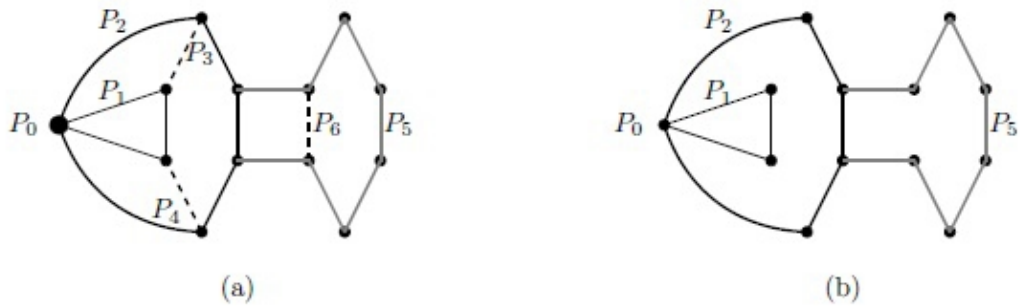


FIG. 4.7 – Illustration de l'étape 1 (à gauche, notée (a)) et de l'étape 2 (à droite, notée (b)) de A_{HT} .

En outre, le graphe obtenu après avoir retiré les oreilles triviales contient deux sommets de degré impair. Ensuite, l'étape 3 de A_{HT} corrige la parité de tous les sommets par injection d'une arête pour obtenir un graphe où les degrés de tous ses sommets sont pairs. (figure 4.8(a).); Un graphe contenant un cycle eulérien (déterminé par l'étape 3 de A_{HT}).

Enfin, en appliquant la procédure dite shortcutting (étape 4 de A_{HT}), nous pouvons observer que le résultat (comme le montre la figure 4.8.(b) représente un cycle hamiltonien obtenu à partir du graphe eulérien de la figure 4.8.(a).

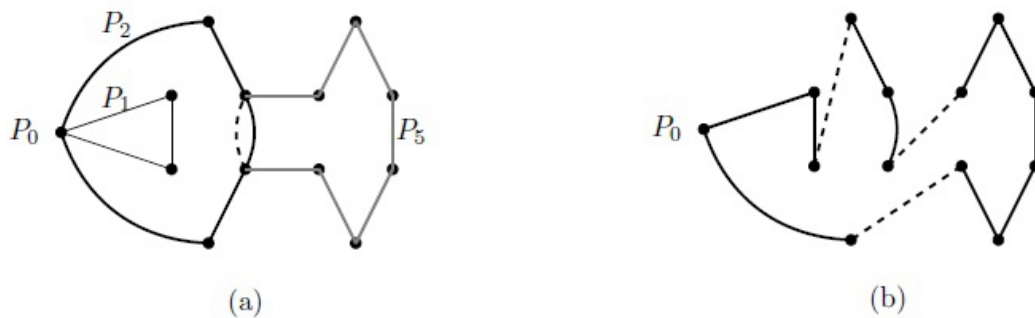


FIG. 4.8 – Illustration de l'étape 3 (à gauche, notée (a)) et de l'étape 4 (à droite, notée (b)) de A_{HT} .

4.6 Conclusion

Dans ce chapitre, nous avons étudié à la fois le problème de recherche de k chaînes hamiltoniennes, dans un graphe cubique et 2-sommet connexe avec k dépôts fixes (noté k DHPP) et le problème du plus court cycle hamiltonien (noté Min-GTSP) dans un graphe facteur critique et 2-sommet connexe. Tout d'abord, nous avons prouvé l'existence d'un algorithme polynomial pour le k DHPP de rapport d'approximation de $\frac{5}{3}$ qui donne une solution avec au plus $(\frac{5}{3}n - \frac{4k-2}{3})$ arêtes. Nous avons proposé un algorithme d'approximation pour le Min-GTSP pour lequel il n'existe pas de rapport meilleur que $\frac{7}{5}$ [84]. Son principe est basé sur la décomposition du graphe en une série d'oreilles et il réalise une approximation de rapport de $\frac{7}{6}$.

Chapitre 5

Approximation du problème de k voyageurs de commerce

5.1 Introduction

Dans ce chapitre, nous considérons le problème de k voyageurs de commerce dans un graphe métrique et 2-arête connexe où les k voyageurs doivent parcourir k chaînes hamiltoniennes en passant par tous les sommets du graphe et en utilisant un nombre minimum d'arêtes sans spécifier les extrémités des chaînes. Nous montrons pour ce problème qu'il existe un algorithme d'approximation de rapport de $\frac{3}{2}$.

5.2 Définitions et préliminaires

5.2.1 Un multigraphe

Soit $G = (V, E)$ un graphe simple non orienté, où V est un ensemble de n sommets et E un ensemble de m arêtes. Le multigraphe est obtenu en doublant les arêtes d'un graphe simple. On le note $2G$.

5.2.2 T -join

Soit $G = (V, E)$ un graphe et T un sous-ensemble de V ($T \subseteq V$ est de cardinalité paire). Un T -join est un sous-ensemble $J \subseteq E$ tel que chaque sommet de T est l'extrémité d'un nombre impair des arêtes de J et chaque sommet de $V \setminus T$ est l'extrémité d'un nombre pair des arêtes de J . La taille d'un T -join est $|J|$.

Exemple

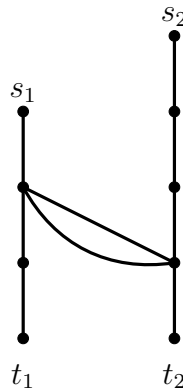


FIG. 5.1 – Un 4-join connexe

Dans [84], *Sebő* et al. ont prouvé l'existence d'un algorithme polynomial pour trouver les T -join connexes dans $2G$, où G est 2-arête connexe de rapport d'approximation de $\frac{3}{2}$ et pour le cas particulier où $T = \emptyset$, un rapport de $\frac{7}{5}$. Leur algorithme utilise des techniques basées sur la décomposition du graphe en oreilles.

En nous basant sur leur approche, nous montrons qu'il existe un algorithme d'approximation pour le problème de k voyageurs de commerce dans le cas graphique.

5.3 Définition du problème

Le problème de k voyageurs de commerce version chaîne (Min- k GPTSP), sans spécifier les extrémités des chaînes, dans un graphe métrique et 2-arête connexe est défini comme suit :

Soit $G = (V, E)$ un graphe connexe. Trouver k chaînes qui passent par tous les sommets de G en utilisant un nombre minimum d'arêtes.

Lemme 5.3.1. Soit $opt(G)$ une solution optimale du problème de k voyageurs de commerce graphique, du graphe connexe G . Alors il existe un T -join connexe dans $2G$ de taille $\leq |opt(G)| + 2(k - 1)$.

Preuve . Soit $opt(G) = P_1, P_2, \dots, P_k$, et $\{s_i, t_i\}$ les extrémités de P_i pour $i = 1 \dots k$. Considérons le multi graphe $P = P_1 + P_2 + \dots + P_k$ et C_1, C_2, \dots, C_ℓ ($\ell \leq k$) ses composantes connexes.

Puisque G est connexe il existe alors $\ell - 1$ arêtes $e_1, e_2, \dots, e_{\ell-1}$ qui rendent $P_1 + P_2 + \dots + P_k + e_1 + e_2, \dots + e_{\ell-1}$ connexe.

Considérons le multigraphe $G' = P_1 + P_2 + \dots + P_k + 2e_1 + 2e_2 + \dots + 2e_\ell$ où $2e_i$ est notée une double copie de e_i . Il est clair que dans ce graphe, tous les sommets sont de degré pair sauf les sommets $\{s_i, t_i\}$ pour $i = 1 \dots k$.

Soit G'' un sous-graphe partiel de G' obtenu de la manière suivante :

Pour toute multi-arête de multiplicité impaire, laissons une seule copie et pour toute multi-arête de multiplicité paire, laissons deux copies. Alors G'' est un sous-graphe partiel de $2G$ connexe et tous les sommets sauf les $\{s_i, t_i\}$ sont de degré pair dans G'' .

Par conséquent, G'' est une $\bigcup_{i=1}^k \{s_i, t_i\}$ -join connexe dans $2G$ et sa taille est $\leq |opt(G)| + 2\ell \leq |opt(G)| + 2(k - 1)$.

5.4 Algorithme

L'algorithme comprend les étapes suivantes :

Algorithm 4 Recherche de k chaînes hamiltoniennes

Soit G un graphe métrique et 2-arête connexe.

1. Chercher une décomposition en oreilles de G en utilisant la méthode citée dans [84] puis supprimer les oreilles triviales et déterminer tous les T -join pour $T = 0, \dots, 2k$ et $T \subseteq V$.
 2. Déterminer un graphe eulérien pour chaque T -join en ajoutant un sommet dit hub relié à tous les sommets de l'ensemble T .
 3. Supprimer les occurrences du hub si nécessaire pour avoir une solution de k chaînes correspondantes à chaque T -join.
 4. Retourner la meilleure solution.
-

Théorème 5.4.1. [84] Soit $G = (V, E)$ un graphe 2-arête connexe et $T \subseteq V$ tel que $|T| = 2k$, un T -join connexe peut être trouvé par un algorithme d'approximation de rapport de $\frac{3}{2}$ dans $2G$ en $O(n^3)$.

Théorème 5.4.2. Il existe un algorithme polynomial de rapport d'approximation de $\frac{3}{2} + \epsilon$ de complexité de $O(n^{2k+3})$ pour le Min- k GPTSP dans le cas où le graphe est 2-arête connexe.

Preuve . Soit $\tau(T, G)$ la taille du T -join obtenu par l'algorithme de [84]. Par le lemme 5.3.1, il existe T^* , $|T^*| \leq 2k$ tel que : il existe un T^* -join connexe de taille $\sigma(T^*, G) \leq \text{opt}(G) + 2(k - 1)$. Par construction, l'algorithme ci-dessus donne toujours une solution s de coût total inférieur ou égal à $\sigma(T^*, G)$ qui est inférieur ou égal à $\frac{3}{2}\sigma(T^*, G)$. Et du théorème 5.4.1 on a :

$$C(s) \leq \frac{3}{2}\sigma(T^*, G) \leq \frac{3}{2}(\text{opt}(G) + 2(k - 1)) \Rightarrow C(s) \leq \frac{3}{2}\text{opt}(G) + O(1).$$

Montrons maintenant que la complexité de l'algorithme est de $O(n^{2k+3})$.

Par le théorème 5.4.1, il existe un algorithme polynomial pour la recherche d'un T -join de coût minimum de rapport d'approximation de $\frac{3}{2}$ de complexité de $O(n^3)$, tandis que trouver k -chaînes eulériennes pour chaque join peut se faire en temps linéaire. L'algorithme calcule au total $O(n^{2k})$ T -join, d'où la complexité totale de l'algorithme est de $O(n^{2k+3})$.

5.5 Application de l’algorithme pour $k = 2$

Nous allons maintenant illustrer les différentes étapes de l’algorithme par un exemple pour $k = 2$. En utilisant le graphe 2-arête connexe de la figure 5.2 et en appliquant la méthode de décomposition en oreilles, nous obtenons la figure 5.3.

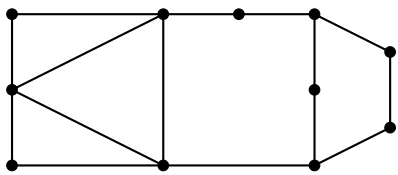


FIG. 5.2 – Graphe 2-arête connexe

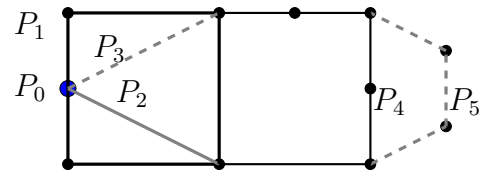


FIG. 5.3 – Décomposition en oreilles du graphe G

Nous supprimons les oreilles triviales : P_2 et P_3 de la figure 5.3 pour obtenir la figure 5.4.

Les figures 5.5, 5.6 et 5.7 représentent les \emptyset -join, 2-join et 4-join de coût minimum respectivement obtenus en appliquant l’étape 1 de l’algorithme.

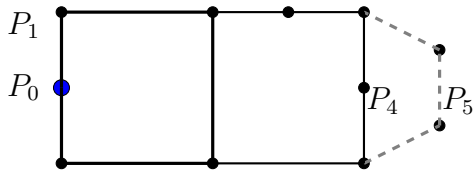


FIG. 5.4 – Suppression des oreilles triviales

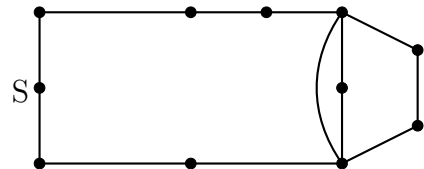


FIG. 5.5 – \emptyset -join

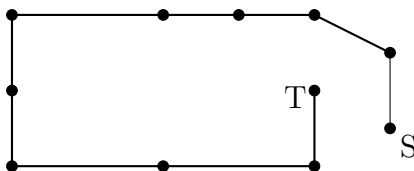


FIG. 5.6 – 2-join

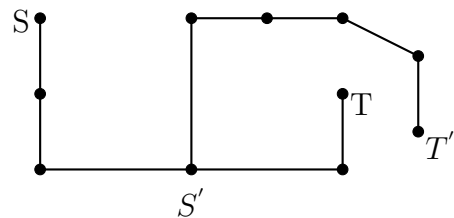


FIG. 5.7 – 4-join

Pour chaque T -join de coût minimum et en appliquant l’étape 2 de l’algorithme, on cherche un graphe eulérien. La figure 5.8 montre le graphe eulérien obtenu par le graphe

de 4-join en ajoutant un sommet H dit hub relié à toutes les extrémités du graphe 4-join. En utilisant l'étape 3 de l'algorithme, on obtient la meilleure solution représentée dans la figure 5.9.

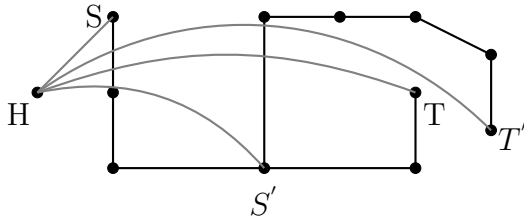


FIG. 5.8 – Le graphe eulérien obtenu

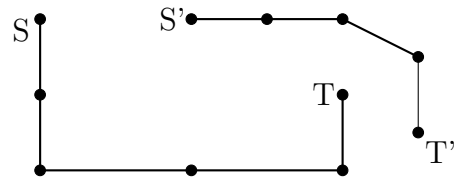


FIG. 5.9 – La solution finale

5.6 Conclusion

Dans ce chapitre, nous avons étudié le problème de k -voyageurs de commerce où aucune extrémité de la chaîne n'est spécifiée. Un tel problème peut être considéré comme une variante du problème de la chaîne hamiltonienne à plusieurs dépôts. Nous avons établi un algorithme d'approximation polynomial de rapport de $\frac{3}{2} + \epsilon$ en nous basant sur la recherche d'un T -join connexe dans le graphe. Cette stratégie est déjà utilisée par *Sebő* et *Vygen* [84]. Le résultat est établi lorsque le graphe est métrique et 2-sommet connexe.

Conclusion et perspectives

Dans ce manuscrit, nous avons étudié l'approximation polynomiale de quelques variantes du problème du voyageur de commerce où il s'agit de trouver des algorithmes qui fournissent, en un temps polynomial, une solution réalisable garantissant une certaine qualité qu'on mesure avec un rapport d'approximation.

Les instances générales de ce problème sont inapproximables mais dans la famille d'instances vérifiant l'inégalité triangulaire, le meilleur facteur d'approximation connu est $\frac{3}{2}$ prouvé par Christofides en 1976. Il est conjecturé qu'un rapport de $\frac{4}{3}$ existe [41].

Dans notre travail, nous avons d'abord étudié le problème de la chaîne hamiltonienne à trois dépôts pour lequel nous avons proposé un algorithme d'approximation qui est une extension de l'algorithme utilisé par Rathinam et al[79]. Pour ce problème, nous avons trouvé une approximation de $\frac{3}{2}$ dans le cas où les coûts sur les arêtes sont symétriques et vérifient l'inégalité triangulaire.

Nous avons aussi étudié la version généralisée de ce problème (k dépôts) dans un graphe métrique, cubique et 2-sommet connexe. Dans ce cas, nous avons utilisé la technique de décomposition en couplage parfait. L'algorithme proposé donne une solution de rapport $\frac{5}{3}$ avec au plus $(\frac{5}{3}n - \frac{4k-2}{3})$ arêtes où n est le nombre de sommets du graphe et k le nombre de dépôts.

La troisième variante étudiée est la recherche d'un cycle hamiltonien dans un graphe 2-arête connexe. Dans ce cas, nous avons présenté un algorithme de rapport

d'approximation de $\frac{7}{6}$. La méthode proposée s'appuie sur la décomposition du graphe en oreilles, cette technique est utilisée par Sebö et al.[83].

A la fin, nous avons étudié une variante chaîne du problème de k voyageurs de commerce sur un graphe 2-arête connexe où k voyageurs vont parcourir k chaînes en passant par tous les sommets du graphe une seule fois sans spécifier les extrémités des chaînes. Nous avons montré qu'il existe un algorithme d'approximation de rapport de $\frac{3}{2} + O(1)$.

Cependant, tous ces résultats ont été orientés uniquement vers l'approximation standard (ou du moins ont permis d'obtenir des résultats uniquement en approximation standard). En particulier, nous avons utilisé des classes restreintes de graphes.

En perspective, il nous paraît intéressant d'étendre ces résultats à une classe de graphes plus large, étudier l'approximation de ces variantes dans le cas des graphes orientés et tenter aussi d'améliorer les rapports d'approximation.

Annexe

Graphe

Un graphe non-orienté $G(V, E)$ est un schéma constitué d'un ensemble fini de points appelés sommets et d'un ensemble fini de lignes appelées arêtes. On note $n = |V|$ et $m = |E|$ le nombre de sommets et le nombre d'arêtes de G , respectivement. Le nombre de sommets du graphe G est appelé ordre de G . Une arête de la forme (x, x) est appelée boucle.

Arêtes adjacentes, sommets adjacents, voisin d'un sommet

Deux sommets u et v sont adjacents dans G si u et v sont reliés par au moins une arête.

Deux arêtes sont adjacentes si et seulement si elles ont au moins une extrémité en commun.

Un voisin d'un sommet u est un sommet qui est adjacent à u .

Degré d'un sommet

Le degré d'un sommet x est égal au nombre d'arêtes dont il est une extrémité. On le note $d_G(x)$.

Sommet hub

Un sommet hub est un sommet du plus grand degré dans un graphe.

Sommet racine

Un sommet est dit racine s'il n'a pas de prédécesseurs.

Graphe simple

Un graphe simple est un graphe sans boucle dans lequel deux sommets sont reliés par au plus une arête.

Sous- graphe induit

Soit $V' \subseteq V$ ($E' \subseteq E$, respectivement). Le sous-graphe induit de G sur V' (sur E' , respectivement), noté $(G[V'] \ G[E']$, respectivement) est $(V', v, u \in E | v, u \in V')$ ($v \in V | \exists v, u \in E', E'$), respectivement).

Graphe cubique

Un graphe $G = (V, E)$ est dit cubique si $\forall x \in V, d_G(x) = 3$.

Sous- graphe

Un sous- graphe $H = (V', E')$ de G est un graphe tel que $V' \subseteq V$ et $E' \subset E$.

Chaîne

Soit $G = (V, E)$ un graphe. Une chaîne $\mu = \{u_1, u_2, \dots, u_q\}$ d'arêtes de G telle que chaque arête de la séquence ait une extrémité en commun avec l'arête précédente et l'autre extrémité en commun avec l'arête suivante. Le nombre d'arêtes dans la séquence est appelée la taille de la chaîne μ . Une chaîne qui ne rencontre pas deux fois le même sommet est dite élémentaire. Une chaîne qui n'utilise pas deux fois la même arête est dite simple.

Cycle

Un cycle est une chaîne $\mu = (u_1, u_2, \dots, u_1)$ telle que :

- la même arête ne figure pas deux fois dans la séquence,
- les deux sommets aux extrémités de la chaîne coïncident.

Un cycle élémentaire est une chaîne élémentaire dont les extrémités coïncident.

Un cycle simple est une chaîne simple dont les extrémités coïncident.

Graphe acyclique

Un graphe est dit acyclique s'il ne contient pas de cycles.

Chaîne eulérienne, cycle eulérien

Une chaîne eulérienne est une chaîne composée de toutes les arêtes du graphe, prises une seule fois.

Un cycle eulérien est une chaîne eulérienne dont les extrémités coïncident.

Chaîne hamiltonienne, cycle hamiltonien

Un cycle (respectivement une chaîne) hamiltonien est un cycle (resp. une chaîne) qui passe une et une seule fois par tous les sommets de G .

Graphe pondéré

Un graphe (orienté ou non) est dit pondéré lorsque ses arêtes sont affectées de nombres positifs.

Le poids ou le coût d'une arête est le nombre positif qui lui est affecté. Le poids ou le coût d'une chaîne est la somme des poids des arêtes qui la composent.

Une plus courte chaîne entre deux sommets u et v est une chaîne de poids minimal parmi toutes les chaînes reliant les sommets u et v .

Graphe connexe

Un graphe $G = (V, E)$ est connexe si $\forall x, y \in V$, il existe une chaîne reliant x à y .

Graphe k-arête connexe

Soit G un graphe connexe. Un graphe G est k-arête connexe si la suppression de k-1 arêtes, quelles qu'elles soient, le graphe reste connexe.

Graphe k-sommet connexe

Soit G un graphe connexe. Un graphe G est k-sommet connexe (ou graphe k-connexe) si la suppression de k-1 sommets, quels qu'ils soient, le graphe reste connexe.

Graphe complet

Un graphe complet est un graphe simple dont tous les sommets sont adjacents deux à deux. On le note K_n .

Arbre

Le graphe G est un arbre si et seulement si G est connexe et ne contient pas de cycle. De manière équivalente, G est un arbre si et seulement si G est connexe et comporte $n - 1$ arêtes, n est le nombre de sommets de G .

Forêt

Une forêt est un graphe sans cycle (chacune de ses composantes connexes est un arbre).

Matroïde

Soit E un ensemble fini et soit \mathcal{F} une famille de sous-ensembles de E . Le couple (E, \mathcal{F}) est un matroïde si :

- (a) $\emptyset \in \mathcal{F}$.
- (b) si $\mathcal{F}_1 \in \mathcal{F}$ et $\mathcal{F}_2 \subseteq \mathcal{F}_1$ alors $\mathcal{F}_2 \in \mathcal{F}$.
- (c) si $\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{F}$ et $|\mathcal{F}_2| = |\mathcal{F}_1| + 1$, alors il existe $e \in \mathcal{F}_2 - \mathcal{F}_1$ tel que $\mathcal{F}_1 \cup \{e\} \in \mathcal{F}$.

Les éléments de \mathcal{F} sont appelés ensembles indépendants de E .

Matroïde graphique

Soit $G = (V, E)$ un graphe. Soit \mathbf{F} un ensemble de sous-graphes partiels sans cycle. Alors $M_g = (E, \mathbf{F})$ est un matroïde, appelé matroïde graphique.

Matroïde de partition

Soit E un ensemble fini, soit $\{E_1, E_2, \dots, E_k\}$ une partition de E en k sous-ensembles, et soient n_1, \dots, n_k , k entiers positifs. Soit $\mathcal{F} = \{F \subseteq E \text{ tel que } |F \cap E_i| \leq n_i, i = 1, \dots, k\}$, (E, \mathcal{F}) est un matroïde appelé matroïde de partition.

Couplage

Soit G un graphe simple. Un couplage M de G est un ensemble d'arêtes deux à deux non-adjacentes.

Sommet saturé

Un sommet v est saturé par un couplage M si v est l'extrémité d'une arête de M .

Couplage maximum et couplage parfait

Un couplage maximum est un couplage contenant le plus grand nombre possible d'arêtes. Un graphe peut posséder plusieurs couplages maximum.

Un couplage parfait est un couplage où chaque sommet du graphe est saturé.

Bibliographie

- [1] Aggarwal N., Garg N., and Gupta S., *A $\frac{4}{3}$ -approximation for TSP on cubic 3-edge-connected graphs*, arXiv :1101.5586v1, 28 janvier 2011.
- [2] Alexander A., Boyd S. and Elliott-Magwood P., *On the integrality gap of the 2-edge connected subgraph problem*, Technical Report TR-2006-04, SITE, University of Ottawa 2006.
- [3] An H. C., Kleinberg R. and Shmoys D. B., *Improving Christofides' algorithm for the s-t path TSP*, In Proc. of the 44th ACM STOC, 2012.
- [4] Applegate D. L., Bixby R., Chvátal V. and Cook W. J., *The Traveling Salesman Problem : A Computational Study*, Princeton University Press 2006.
- [5] Arora S., *Approximation Schemes for NP-hard geometric optimization problems : a survey*, Mathematical Programming, Ser. B. 97 : 43-69, 2003.
- [6] Arora S. and Lund C., *Hardness of approximations*, In D. Hochbaum, editor, Approximation Algorithms for NP-hard Problems. PWS Publishing, 1996.
- [7] Asadpour A., Goemans M.X., Madry A., Oveis Gharan S. and Saberi A., *An $O(\frac{\log n}{\log(\log n)})$ -approximation algorithm for the asymmetric traveling salesman problem*, Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms

- (SODA 2010), 379–389, 2010.
- [8] Ausiello G., D’Atri A. and Protasi M. *On the structure of combinatorial problems and structure preserving reductions. In Proc. ICALP’77, volume 52 of Lecture Notes in Computer Science, pages 45–60. Springer-Verlag, 1977.*
- [9] Ausiello G., D’Atri A. and Protasi M. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Science* 21.136-153, 1980
- [10] Ausiello G., Crescenzi P., Gambosi G., Kann V., Spaccamela A. M., and Protasi M., *Complexity and approximation, Combinatorial optimization problems and their approximability properties*, Springer-Verlag, Berlin, 1999.
- [11] Bae J. and Rathinam S., *Approximation algorithms for multiple terminal, Hamiltonian path problems*, *Optimization Letters* vol 6(p) : 69-85, 2012.
- [12] Barahona F. and Conforti M., *A construction for binary matroids*, *Discrete Mathematics* 66 : 213-21, 1987.
- [13] Berman P. and Karpinski M., $\frac{8}{7}$ -approximation algorithm for (1,2)-TSP , *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*.
- [14] Boyd S., Sitters R., Van der Ster S. and Stougie L., *TSP on cubic and subcubic graphs*, *In Proc. of the 15th IPCO, LNCS 6655, Springer, 65-77, 2010.*
- [15] Bresovec C., Cornuèjols G., Glover F., *A matroid algorithm and its application to the efficient solution of two optimization problems on graphs*, *Mathematical Programming* vol. 42 : 471-487, 1998.

- [16] Carr R. and Ravi R., *A new bound for the 2-edge connected subgraph problem*, In : Integer Programming and Combinatorial Optimization ; Proceedings of the 6th International IPCO Conference ; LNCS 1412 (R.E. Bixby, E.A. Boyd, R.Z. Los-Mercado, eds.), Springer, Berlin 1998.
- [17] Cerdeira J. O., *Matroids and a forest cover problem*, Mathematical Programming 66 : 403-405, 1994.
- [18] Cheriyan J., Friggstad Z. and Gao Z., *Approximating minimum-cost connected T-joins*, Approximation, Randomization, and Combinatorial Optimization Proceedings of the 16th APPROX Workshop ; LNCS 7408 (A. Gupta, K. Jansen, J. Rolim, R. Servedio, eds.), Springer, Berlin 2012, 110–121.
- [19] Cheriyan J., Sebő A. and Szigeti Z., *Improving on the 1.5-approximation of a smallest 2-edge connected spanning subgraph*, SIAM, Journal on discrete mathematics 14 : 170-180, 2001.
- [20] Christofides N., *Worst-case analysis of a new heuristic for the travelling salesman problem*, Report 388. Graduate School of industrial Administration. Carnegie-Mellon University, Pittsburgh 1976.
- [21] Cook W. J., *In Pursuit of the Traveling Salesman*, Mathematics at the Limits of Computation. Princeton University Press 2012.
- [22] Cook W. J. and Rohe A., *Computing Minimum Weight Perfect Matchings*, Inform's Journal on Computing 11, 138-148, 1999.
- [23] Cook W. J., Cunningham W. J., Pulleyblank W. R. and Shrijver A., *Combinatorial Optimization*, Wiley Interscience, 1998.

- [24] Cook.S.A., *The complexity of theorem-proving procedures. In Proceedings of the 3rd Symposium on Theory of Computing, pages 151–158, New-York, 1971. Association for Computing Machinery.*
- [25] Cordone R. and Maoli F., On the complexity of graph tree partition problems, *Discrete Applied Mathematics 134 : 51-65, 2004.*
- [26] Cornu Lejols G. and Nemhauser G. L., Tight bounds for Christofides' traveling salesman heuristic, *Math. Programming 14 : 116-121, 1978.*
- [27] Cornu Lejols G., Fonlupt J., and Naddef D., The traveling salesman problem on a graph and some related integer polyhedra, *Mathematical Programming 33, 1.27, 1985.*
- [28] Demange M., Algorithme d'approximation : un petit tour en compagnie d'un voyageur de commerce, *Mathématiques et Informatique, Gazette 102, 2004.*
- [29] Demange M., Grisoni P. and Paschos V. Th., Differential approximation algorithms for some combinatorial optimization problems, *Theoretical Computer Science 209 : 107-122, 1998.*
- [30] Demange M and Paschos.V.Th., On an approximation measure founded on the links between optimization and polynomial approximation theory. *Theoretical Computer Science 158, 117-141, 1996.*
- [31] Demange M. and Paschos V.Th., *Quelques étapes vers la conciliation de la théorie d'approximation polynomiale et résultat préliminaires*, C.R. Acad. Sci. Paris Sér.I Math.317 : 409-414, 1993.

- [32] Edmonds J. and Johnson E. L., *Matching, Euler tours and the Chinese postman*, Mathematical Programming 5 : 88-124, 1973.
- [33] Edmonds J., *Maximum matching and a polyhedron with 0,1-vertices*, Journal of Research of the National Bureau of Standards vol. 69B : 125-130, 1965.
- [34] Escoffier B. and Paschos V. Th., *Structures des classes d'approximation : un état de l'art*, CAHIER DU LAMSADE 247, université Paris DAUPHINE, Février 2007.
- [35] Fernandes C. G., *A better approximation ratio for the minimum size k -edge-connected spanning subgraph problem*, Journal of Algorithms 28, 105–124, 1998.
- [36] Fisher M. L., Nemhauser G.L. and Wolsey L.A., *An analysis of approximations for finding a maximum weight hamiltonian circuit*, Operation Research 27 (4) : 799-809, 1979.
- [37] Frieze A. M., *An extension of Christofides heuristic to the k -person travelling salesman problem*, Discrete Appl. Math. 6 : 79–83, 1983.
- [38] Gabow H. N., *An ear decomposition approach to approximating the smallest 3-edge connected spanning subgraph of a multigraph*, SIAM J. Discrete Math. 18 No.1, 41-70, 2004.
- [39] Gamarnik D., Lewenstein M. and Sviridenko M., *An improved upper bound for the TSP in cubic 3-edge-connected graphs*, Operations Research Letters, 33(5) : 467-474, 2005.
- [40] Garey M. R. and Johnson D. S., *Computers and intractability : a guide to the theory of NP-completeness*. Freeman, 1979.

- [41] Goemans M., “Worst-case comparison of valid inequalities for the TSP,” *Mathematical Programming*, vol. 69, pp. 335–349, 1995.
- [42] Gonzalez T. and Sahni S., *P-complete approximation problems*, *Journal of the ACM (JACM)*, 23(3) :555-565, July 1976.
- [43] GuoXing Y., *Transformation of multidepot multisalesmen problem to the standard traveling salesman problem*, *European Journal of Operational Research* 81 : 557-600, 1995.
- [44] Gutin G. and Punnen A. P., *The traveling salesman problem and its variations*, Kluwer Academic Publishers, 2002.
- [45] Guttmann B., Hassin R., Khuller S., Raghavachari B., *Approximation algorithm with bounded performance guarantees for the clustered travelling salesman problem*.1998 FST & TCS conference (Madras,India) *Algorithmica* 28, 422-437, 2000.
- [46] Held M. and Karp R. M., *The traveling salesman problem and minimum spanning trees*, *Operations Research* 18 : 1138-1162, 1970.
- [47] Hochbaum D.S., *Approximation algorithms for NP-hard problems*, PWS, Boston, 1997. chapter 10 : C. Arora and C. Lund, *Hardness of approximation*, 399-446, 1997.
- [48] Hoogeveen J. A., *Analysis of Christifides’ heuristic : Some paths are more difficult than cycles*, *Operations Research Letters*, 10 : 291-295, July 1991.
- [49] Johnson D. S. and Papadimitriou C. H., *”Performance guarantees for heuristics”*, In : E. L. Lawler, J.K. Lenstra, A.H.G. RINNOOY KAN and D. B. SHMOYS, (eds.) *The Traveling Salesman Problem : A Guided Tour of Combinatorial Optimization*,

- Wiley, Chichester, 145-180, 1985.
- [50] Johnson. D.S, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., 9 : 256–278, 1974.
- [51] Jonker R. and Volgenant T., *An improved transformation of the symmetric multiple traveling salesman problem*, Operations Research 36(1) : 163-167, 1988.
- [52] Karp M. R, *Reducibility among combinatorial problems*, Complexity of computer computations, 40(4) :85-103, 1972.
- [53] Karpinski M. and R. Schmied R., *On approximation lower bounds with bounded metrics. CoRR arXiv :abs/1201.5821,2012.*
- [54] *Karpinski M. and R. Schmied R., Approximation Hardness of Graphic TSP on Cubic Graphs, Working paper, 2013.*
- [55] *Karpinski M., Lampis M., and Schmied R., New inapproximability bounds for TSP. Journal of Computer and System Sciences 81 : pp.1665-1677, 2015.*
- [56] *Kolaitis P. G. and Thakur M. N., Approximation properties of NP minimization classes, Journal of Computer and System Sciences, 50(3) : 391–411, 1995.*
- [57] *Korte B., and Vygen J., Combinatorial Optimization - Theory and Algorithms, Springer Verlag (2d édition), 2002.*
- [58] *Krentel M. W., The complexity of optimization problems, J. Comput. System Sci., 36 : 490–509, 1988.*

- [59] Lampis M., Improved inapproximability for TSP, *Theory Comput* 10(9) : 217-236, 2014
- [60] Lawler E.L., Combinatorial Optimisation : Networks and Matroids, *Dover Publication*, 2001.
- [61] Lawler E.L., Lenstra J.K., Rinnooy Kan and Shmoys D.B., The traveling salesman problem, *Wiley Interscience*, New York, 1986.
- [62] Lawler E.L., Lenstra J.K., Rinnooy Kan A. H. G. and Shmoys D.B., The Traveling Salesman Problem : A Guided Tour of Combinatorial Optimization, *Wiley Interscience Series in Discrete Mathematics*, 1985.
- [63] Lovász L., A note on factor critical graphs, *Studia Sci. Math. Hungar* 7 : 279-280, 1972.
- [64] Malik W., Rathinam S. and Darbha S., An approximation algorithm for a symmetric generalized multiple depot, multiple traveling salesman problem, *Operations Research Letters*, vol. 35 : 747-753, 2007.
- [65] Micali S., and Vazirani V.V., An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *Proc. of the 21st Annual IEEE Symposium on Foundations of Computer Science*, 17-27, 1980.
- [66] Mömke T. and Svensson O., Approximating graphic TSP by matchings, In *Proc. IEEE FOCS*, 560-569, 2011.
- [67] Monnot J., Paschos V. Th. and Toulouse S., Approximation algorithms for the traveling salesman problem, *Mathematical Methods of Operations Research*, 57(1) :

- 387-405, 2003.
- [68] Monnot J., Paschos V. Th. and Toulouse S., Approximation polynomiale des problèmes NP- difficiles : optima locaux et rapport différentiel, *Hermès Science*, 2003.
- [69] Monnot J., Differential approximations results for the traveling salesman and related problems, *Information Processing Letters* 82 : 229–235, 2002.
- [70] Mucha M., $\frac{13}{9}$ -approximation for graphic TSP, *In Proc. STACS*, 30-41, 2012.
- [71] Oveis Gharan S., Saberi A., and Singh M., Worst case analysis of a new heuristic for the traveling salesman problem, *In FOCS 2011*.
- [72] Papadimitriou C. H. and Vempala S., On the approximability of the traveling salesman problem, *Combinatorica* 26(1) : 101-120, 2006.
- [73] Papadimitriou C.H., and Yannakakis M., The traveling salesman problem with distances one and two, *Mathematics of Operations Research* 18, 1–12, 1993.
- [74] Papadimitriou C. H. and Yannakakis M., Optimization, approximation and complexity classes, *J. Comput. System Sci.* 43 : 425-440, 1991.
- [75] Papadimitriou C. H. and Steiglitz K., Combinatorial Optimization : Algorithms and Complexity, *Prentice Hall, New Jersey*, 1981.
- [76] Paschos V. Th., Complexité et approximation polynomiale, *Hermès Science*, 2004.
- [77] Rao M.R., A note on Multiple Travelling Salesmen Problem, *Oper. Res.* 28 (3) : 628–632, 1980.

- [78] Rathinam S. and Sengupta R., Lower and upper bounds for a multiple depot UAV routing problem, *In Proc IEEE Control and Decision Conference, San Diego, California 2006*.
- [79] Rathinam S. and Sengupta R., $\frac{5}{3}$ -Approximation Algorithm for a Multiple Depot, Terminal Hamiltonian Path Problem, *RESEARCH REPORT UCB-ITS-RR-2007-1*.
- [80] Rathinam S. and Sengupta R., $\frac{3}{2}$ -approximation algorithm for two variants of 2-depot Hamiltonian path problem, *Operations Research Letters, vol. 38 : 63-68, 2012*.
- [81] Sakarovitch. M, *Optimisation Combinatoire, Méthodes Mathématiques et Algorithmiques. 1984*.
- [82] Sahi S. and Gonzalez T., P-Complete Approximation Problems, *Journal of the ACM, New York, NY, USA, (JACM), Volume 23 Issue 3 : 555-565, July 1976*.
- [83] Sebö A., Eight-Fifth Approximation for the Path TSP, *In Proc. IPCO 2013. LNCS 7801 : 362-374, 1209.3523; 2013*.
- [84] Sebö A. and Vygen V., Shorter tours by nicer ears. $\frac{7}{5}$ -approximation for graphic TSP, $\frac{3}{2}$ for the path version, and $\frac{4}{3}$ for two-edge-connected subgraphs, *Combinatorica 34(5), 1-34; 2014*.
- [85] Schrijver A., *Combinatorial Optimization : Polyhedra and Efficiency, Springer, Berlin 2003*.
- [86] Spellman Munson Clyde B., Monma L. and Pulleyblank R.W., Minimum-weight two-connected spanning networks, *Mathematical Programming, 153-171, 1990*.

- [87] Toulouse S., Approximation polynomiale. optima locaux et rapport différentiel, *Thèse de doctorat, LAMSADE, Université Paris-Dauphine, 2001.*
- [88] Tutte W.T., On hamiltonian circuits, *J. London Math. Soc.* 7, 169–176, 1946.
- [89] Vazirani V., Approximation algorithms, *Springer, Berlin, 2001.*
- [90] Whitney H., Non-separable and planar graphs, *Transactions of the American Mathematical Society* 34 : 339–362, 1932.
- [91] Williamson D. P. and Shmoys D. B., The Design of Approximation Algorithms, *Cambridge University Press, New York, NY, 2011.*
- [92] Xu Z. and Rodrigues B., A $\frac{3}{2}$ -Approximation Algorithm for Multiple Depot Multiple Traveling Salesman Problem, *In Proc Algorithm Theory—SWAT 2010. Lecture Notes in Computer Science, vol. 6139 : 127–138. Springer, Berlin 2010.*
- [93] Zemel. E., Measuring the quality of approximate solutions to zero-one programming problems. *Mathematics of Operations Research* 6, 319–332, 1981.