Sur l'extension de solution partielle

UNIVERSITÉ DE MONTPELLIER

Étude bibliographique

Master *Sciences et Technologies*,

Mention *Informatique*,

Parcours Informatique Théorique

Auteur
Valentin Pollet

Superviseurs
Annie Chateau
Rodolphe Giroudeau

Lieu de stage
LIRMM UM5506 - CNRS, Université de Montpellier

21 février 2017

Résumé

On s'intéresse, dans cette étude bibliographique, à la classification au sens de la complexité et de l'approximation des problèmes lorsqu'une partie de la solution est imposée initialement. Dans l'état de l'art, on trouve que le problème du Plus Court Chemin Élémentaire dans les graphes orientés devient NP-difficile lorsqu'une arête du chemin à trouver est imposée. Le problème Coloration dans les graphes bipartis change également de complexité lorsque l'on cherche à étendre une précoloration des sommets. Enfin, il existe un seuil à partir duquel imposer des arêtes au problème de Voyageur de Commerce le rend facile à résoudre. On propose une approche qui pourra permettre d'établir des résultats de classification des problèmes en fonction de leur changement de statut et des seuils auxquels ces changements apparaissent.

Abstract

When one attempts to compute a solution to a combinatorial problem, the solution is initially empty. We investigate here the complexity and approximability of problems when part of the solution is given as input. We first study the state of the art. The Simple Shortest Path problem in directed graphs is shown to become NP-hard when an edge is given as a compulsory step. We found a similar result for the coloring of bipartite graphs. At last, the Travelling Salesman Problem becomes easy to solve when enough edges are forced to belong to the solution. We call such complexity changes *status changes.* Several approximability results are given for these problems. Following the background study, we present our approach which may help us to classify problems according to their complexity and approximability status change, and the threshold at which the changes appear.

Table des matières

Chapitre 1

Introduction

La théorie de la complexité permet de classifier les différents problèmes algorithmiques en fonction de leur difficulté de résolution. Les problèmes peuvent être dits de *décision* lorsqu'il s'agit de répondre OUI ou NON à une question, ou d'*optimisation* lorsque la résolution implique l'optimisation d'un critère et de donner une solution. Ce distinguo est primordial puisque les problèmes sont classifiés différemment selon leur nature.

Outre permettre de classifier et hiérarchiser les problèmes, la théorie de la complexité introduit les notions primordiales de *réductibilité* et de *complétude*. Assez intuitivement, un problème A est *réductible* à un problème B si dans tous les cas la résolution de A peut être ramenée à la résolution de B. La *complétude* d'un problème A appartenant à une classe $\mathscr{C}$ implique la réductibilité de tout autre problème de la classe à A ; on dit alors que A est *$\mathscr{C}$-complet*. Un problème complet d'une classe est donc au moins aussi difficile à résoudre que tous les autres problèmes de la classe.

Les classes de problèmes de décision P et NP forment la pierre angulaire de la théorie de la complexité, dans la mesure où de nombreux résultats reposent sur des hypothèses les concernant. On dit qu'un problème de décision appartient à la classe P s'il existe un algorithme résolvant toute instance du problème en temps raisonnable. On parle de temps raisonnable lorsqu'il est polynomial en la taille de la donnée. Un problème appartient à NP lorsque l'on peut certifier qu'une solution à une instance satisfait bien aux contraintes du problème en temps polynomial en la taille de l'instance.

Les problèmes NP-complets, les plus difficiles de la classe NP, sont parmi les plus étudiés puisqu'ils sont suspectés de ne pas être résolubles efficacement sur les machines déterministes au nombre d'états dénombrable que sont les ordinateurs actuels. Les méthodes de résolution ne sont, dans ce cas, pas exécutables en temps polynomial et on dit d'un tel problème qu'il n'est pas *tractable*. À l'opposé, les problèmes de la classe P, admettent tous des algorithmes de résolution polynomiaux, aussi ces problèmes sont-ils considérés comme faciles à résoudre.

Pour classifier un problème d'optimisation, on s'intéresse généralement à la variante décisionnelle qui lui est associée. Par exemple, si A est un problème de minimisation d'un coût, la variante décisionnelle consiste à décider s'il existe une solution de coût inférieur ou égal à un k donné. Naturellement, un problème d'optimisation dont la variante décisionnelle est NP-complète ne saurait être tractable, et on peut alors s'intéresser aux algorithmes d'approximation avec garantie de performance pour le résoudre. Ces algorithmes sont généralement polynomiaux et fournissent une solution qui n'est pas optimale mais dont la qualité est garantie par un ratio entre le coût de la solution approchée et le coût d'une solution optimale. On parle alors de ratio d'approximation.

Le ratio d'approximation permet de hiérarchiser les problèmes d'optimisation. Si un problème d'optimisation est approximable à ratio constant, le problème appartient à la classe APX. Si en revanche tous les algorithmes approchés exhibables pour un problème ont un ratio qui dépend de la taille de l'entrée, le problème n'est plus APX mais peut, par exemple, être Log-APX lorsque la dépendance est logarithmique. Il peut également exister un schéma d'approximation polynomial pour un problème, dans ce cas il appartient à la classe PTAS. Un schéma d'approximation est un algorithme approché dont le ratio peut être arbitrairement proche de 1, et donc la solution qu'il retourne arbitrairement proche de l'optimal, mais dont le temps d'exécution dépend de cette proximité. Naturellement, un problème PTAS est plus facilement approchable qu'un problème APX, qui est lui même plus facilement approchable qu'un problème Log-APX. En ce sens apparait la hiérarchie des approximations, et l'on peut par exemple montrer, sous l'hypothèse que $P \neq NP$, qu'un problème ne peut pas appartenir à APX ou à PTAS, et ainsi établir une hiérarchie des problèmes d'optimisation.

La donnée d'une instance d'un problème, qu'il soit de décision ou d'optimisation, n'impose habituellement pas de solution partielle devant être étendue. Si l'on considère un problème d'optimisation classique comme celui du voyageur de commerce, la donnée est un ensemble de villes et des distances entre tout couple de ces villes, et ce que l'on cherche est une tournée (un cycle Hamiltonien) passant une et une seule fois par toutes les villes telle que la distance totale parcourue soit minimale, ou maximale selon la variante considérée. La donnée n'impose alors pas que certaines villes soient parcourues consécutivement dans la tournée. Imposer une telle consécutivité reviendrait à fournir une solution partielle initiale, une contrainte supplémentaire au problème classique.

Dans le cadre de ce stage, on s'intéressera à la classification des problèmes lorsqu'une solution partielle doit être étendue. D'un point de vue de la théorie de la complexité, on cherchera des problèmes difficiles qui sont simplifiés par cette donnée supplémentaire, ou des problèmes faciles qui se complexifient, et l'on tentera d'identifier la densité de solution initiale à partir de laquelle ces changements de complexité apparaissent. La théorie des approximations avec garantie de performance sera l'autre angle d'approche ; on essayera d'étendre cette théorie à ce qu'on appellera le ratio résiduel, c'est à dire le ratio d'approximation sans prendre en compte le coût de la solution partielle imposée. On se posera la question de savoir s'il existe des problèmes de la classe APX qui deviennent non-approximables à facteur constant lorsqu'une solution partielle est imposée, ou au contraire si certains problèmes non-approximables deviennent APX.

Dans un premier chapitre, on définit formellement les notions évoquées au long de cette introduction, ainsi que les problématiques qui seront abordées. Ensuite, on dressera l'état de l'art des publications en relation avec lesdites problématiques. Enfin, l'approche envisagée et les perspectives pour la poursuite du stage seront présentées.

Chapitre 2

Complexité et approximations

La problématique principale de ce stage est la classification des problèmes d'optimisation selon leur comportement lorsqu'une solution partielle est fournie et doit être étendue. Ce chapitre présente, dans un premier temps, des définitions de complexité et d'approximation indispensables à la bonne compréhension des problématiques. Ensuite, le problème considéré dans le cadre de ce stage sera formalisé.

2.1 Généralités

La résolution de problèmes informatiques se fait au moyen d'algorithmes. On évalue généralement la performance d'un algorithme en mesurant sa complexité temporelle et spatiale dans le pire des cas. L'étude du cas moyen peut sembler être une mesure plus naturelle, mais s'avère souvent être complexe eu égards à l'ignorance sur la distribution des données. Dans le cadre de cette étude bibliographique, on étudiera essentiellement la complexité temporelle des algorithmes. De plus, on s'intéressera plutôt à l'ordre de grandeur de la complexité, ainsi la notation O sera employée.

Définition 1 (Notation O)**.** *Soient f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{R}$. On dit que f est un « grand o » de g, noté $f = O(g)$ si : $\exists c, \exists n_0$ tels que $\forall n > n_0$, $f(n) \leq c \times g(n)$. Autrement dit, g borne asymptotiquement f à un facteur constant près.*

Définition 2 (Complexité temporelle)**.** *Soit $\mathscr{A}$ un algorithme pour un problème $\mathscr{P}$. On appelle* complexité temporelle *de $\mathscr{A}$ la fonction donnant le nombre maximal d'opérations élémentaires que $\mathscr{A}$ devra effectuer pour résoudre une instance de $\mathscr{P}$.*

On parle alors de complexité, dans le pire des cas, en $O(f(n))$ où n est une variable représentant la taille d'une instance et f une fonction calculable. f détermine l'efficacité de l'algorithme. Si f est un polynôme l'algorithme est efficace, et son efficacité va dépendre du degré du polynôme. Si f est une exponentielle, il l'est bien moins.

Si ce n'est pas précisé, un temps « exponentiel » ou « polynomial » dépendra de la taille de l'instance. Ainsi lorsqu'on parlera de résoudre un problème en temps exponentiel, il sera sous entendu qu'il existe un algorithme permettant de le résoudre avec une complexité temporelle exponentielle en la taille de l'instance.

En théorie de la complexité, les problèmes sont caractérisés par les classes auxquelles ils appartiennent. Ces classes sont des ensembles de problèmes dont le temps ou l'espace nécessaire à leur résolution sont similaires. Deux notions sont essentielles à la classification par les temps de résolution. La première est la *réduction en temps polynomial.*

Définition 3 (Réduction)**.** *Une* réduction *d'un problème $\mathscr{A}$ vers un problème $\mathscr{B}$ est une fonction calculable permettant de transformer toute instance de $\mathscr{A}$ en une instance de $\mathscr{B}$ telle que $\mathscr{A}$ est positive si et seulement si $\mathscr{B}$ est positive. On dit alors que $\mathscr{A}$ se réduit à $\mathscr{B}$. Si la transformation peut être calculée en temps polynomial, la réduction est dite* polynomiale*, et la taille de l'instance résultante est polynomialement bornée par la taille de l'instance d'origine.*

Si une réduction polynomiale existe d'un problème $\mathscr{A}$ à un problème $\mathscr{B}$, alors $\mathscr{B}$ est au moins aussi difficile à résoudre que $\mathscr{A}$ puisque toute instance de $\mathscr{A}$ peut être transformée en temps polynomial à une instance de $\mathscr{B}$. On peut vérifier que si $\mathscr{B}$ est un problème résoluble en temps polynomial, alors $\mathscr{A}$ l'est aussi. La contraposée est donc valide : si $\mathscr{A}$ n'est pas résoluble en temps polynomial, alors $\mathscr{B}$ ne l'est pas non plus. En utilisant les réductions polynomiales, on peut définir la *$\mathcal{C}$-difficulté* d'un problème.

Définition 4 ($\mathcal{C}$-difficulté)**.** *Soient $\mathscr{P}$ un problème et $\mathcal{C}$ une classe de complexité. $\mathscr{P}$ est $\mathcal{C}$-difficile si et seulement si tout problème de $\mathcal{C}$ se réduit à $\mathscr{P}$ en temps polynomial.*

Au vu de la définition de réduction polynomiale, les problèmes $\mathcal{C}$-difficiles sont des problèmes au moins aussi difficiles à résoudre que les problèmes $\mathcal{C}$. La $\mathcal{C}$-difficulté dénote de la difficulté de résolution d'un problème par rapport à la classe $\mathcal{C}$. Si de plus le problème appartient à $\mathcal{C}$, on arrive à la notion de *complétude.*

Définition 5 ($\mathcal{C}$-complétude)**.** *Soient $\mathscr{P}$ un problème et $\mathcal{C}$ une classe de complexité. $\mathscr{P}$ est $\mathcal{C}$-complet si et seulement si $\mathscr{P}$ est $\mathcal{C}$-difficile et $\mathscr{P} \in \mathcal{C}$.*

2.2 Problèmes de décision

Les classes de problèmes de décision permettent d'établir une hiérarchie en terme de difficulté de résolution. Les classes habituellement considérées sont P et NP, qui peuvent être définies comme il suit.

Définition 6 (Classe P)**.** *Un problème de décision appartient à la classe P si et seulement si il existe un algorithme polynomial pour le résoudre.*

Définition 7 (Classe NP)**.** *Un problème de décision $\mathscr{P}$ appartient à la classe NP si et seulement s'il possède un* vérifieur *polynomial. On appelle* vérifieur *pour $\mathscr{P}$ un algorithme V à réponse booléenne, tel que pour toute instance x positive de $\mathscr{P}$, il existe un c tel que $V(x, c)$. c est appelé le certificat de x. De manière équivalente, il est possible de vérifier en temps polynomial qu'une instance est positive lorsqu'une solution est fournie.*

L'inclusion de la classe P dans la classe NP est triviale puisque s'il est possible de résoudre une instance en temps polynomial, vérifier qu'une solution donnée est bonne ne pourra pas être plus long que la résolution elle-même, ainsi un problème dans P est aussi dans NP. En revanche, on ne sait toujours pas si l'inclusion est stricte.

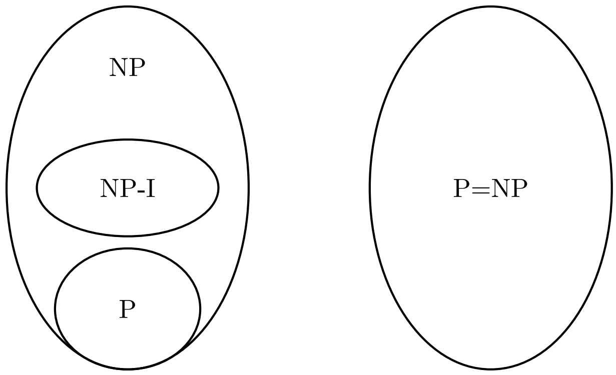

FIGURE 2.1 : $P \neq NP$ ou $P = NP$?

Bien que le problème soit toujours ouvert, puisqu'aucune preuve de l'inclusion stricte ou de l'égalité n'a encore été vérifiée, la communauté scientifique s'accorde à dire que l'inclusion est stricte. Dans ce cas, les problèmes NP-complets ne pourraient être résolus en temps polynomial. Cette hypothèse est cruciale puisqu'il existe des problèmes NP-complets, qui sont donc au moins aussi difficiles à résoudre que tous les autres problèmes de NP. Le premier problème à avoir été montré NP-complet est celui de la satisfiabilité de formule par Cook [12]. Par la suite, la complétude d'une multitude de problèmes a pu être montrée, en utilisant les *réductions en temps polynomial.* Les premiers catalogues de problèmes NP-complets ont été proposés par Karp [22] puis Garey et Johnson [18]. De plus, si $P \neq NP$ alors il existe des problèmes de NP qui ne sont ni dans P, ni NP-complets [26] ; ils forment la classe NP-intermédiaire (voir figure 2.1).

2.3 Problèmes d'optimisation

Les problèmes d'optimisation consistent à optimiser une fonction sous contraintes. Pour déterminer la difficulté d'un tel problème, on considère sa variante décisionnelle.

Définition 8 (Variante décisionnelle)**.** *Soit $\mathscr{P}$ un problème d'optimisation d'une fonction f, qui à toute solution x d'une instance I de $\mathscr{P}$ associe un réel $f(x)$. Si $\mathscr{P}$ est un problème de maximisation, sa variante décisionnelle est la suivante.*

> $\mathscr{P}$-DÉCISIONNEL
> **Donnée :** I instance de $\mathscr{P}$, $k \in \mathbb{R}$.
> **Question :** Existe-t-il une solution x de I telle que $f(x) \geq k$?

Si $\mathscr{P}$ est un problème de minimisation, on a $f(x) \leq k$ dans la question.

La variante décisionnelle étant un problème de décision, il peut appartenir aux classes P et NP. Si sa variante décisionnelle est un problème NP-complet, alors le problème d'optimisation est considéré comme difficile à résoudre. À l'inverse, si sa variante décisionnelle est un problème polynomial, alors le problème d'optimisation sera généralement facile à résoudre en procédant par dichotomie sur la valeur de la fonction à optimiser.

Lorsque le problème d'optimisation est difficile, plusieurs solutions sont envisageables pour le résoudre de manière approchée et efficacement. On peut citer

- Les heuristiques, qui consistent à modifier un algorithme exact en *orientant* la recherche. La complexité est en général améliorée, mais la solution que l'algorithme heuristique renvoie peut être arbitrairement mauvaise.
- Les méta-heuristiques, comme les algorithmes de recherche locale, qui sont des schémas de résolution génériques pouvant être adaptés à différents problèmes. Là encore, les temps d'exécution peuvent être meilleurs que des résolutions exactes, mais les solutions trouvées peuvent être très éloignées de l'optimal.
- Les algorithmes d'approximation en temps polynomial. Dans ce cas, les algorithmes sont spécifiques à chaque problème et retournent une solution dont le rapport à l'optimal est garanti par un ratio.

Dans le cadre de ce stage, ce sont les algorithmes d'approximation qui seront étudiés. Il convient donc de définir formellement ces derniers.

Définition 9 (ρ-approximation)**.** *Soit $\mathscr{P}$ un problème de maximisation (*resp. *minimisation). Pour toute instance I de $\mathscr{P}$, soit OPT_I la valeur d'une solution optimale. Soit $\rho \in]0,1]$ (*resp. $[1,+\infty[$*). Alors un algorithme $\mathscr{A}$ est une ρ-approximation pour le problème $\mathscr{P}$ si pour toute instance I de $\mathscr{P}$:*

- *$\mathscr{A}$ s'exécute en temps polynomial.*
- *$\mathscr{A}(I)$ retournée par l'algorithme est bien une solution.*
- *A_I, la valeur de $\mathscr{A}(I)$, vérifie $A_I \geq \rho \times OPT_I$ (*resp. $A_I \leq \rho \times OPT_I$*)*

ρ est appelé le ratio d'approximation *de l'algorithme.*

Ces algorithmes ont un intérêt double puisqu'ils sont d'une part polynomiaux et d'autre part ils fournissent une solution qui ne pourra être arbitrairement mauvaise, contrairement aux heuristiques. En pratique, les *méta*-heuristiques fournissent souvent de meilleures solutions, mais elles ne garantissent pas qu'elles soient proches de l'optimal. Utiliser un algorithme d'approximation permet d'y remédier puisque si l'on considère une instance d'un problème de maximisation pour laquelle un algorithme $\frac{1}{3}$-approché renvoie une solution de valeur 100, on est certain que l'optimal aura une valeur d'au plus 300. Si maintenant une *méta*-heuristique renvoie une solution de valeur 296 sur cette même instance, on aura la garantie que cette solution est bonne. À l'inverse, si l'heuristique renvoie une solution de valeur 30 on saura qu'il s'agit d'une mauvaise solution.

Il existe des algorithmes d'approximation fournissant encore plus de garantie que les ρ-approximations, il s'agit des schémas d'approximation en temps polynomial ou totalement polynomial (*Polynomial Time Approximation Scheme* en anglais, PTAS, ou *Fully Polynomial Time Approximation Scheme*, FPTAS). Avec ceux-ci, il est possible d'obtenir une solution aussi proche de l'optimal que l'on souhaite en ≪ payant ≫ cette proximité par un temps de calcul plus grand.

Définition 10 (PTAS, FPTAS)**.** *Soit $\mathscr{P}$ un problème de maximisation. Un algorithme $\mathscr{A}$ est un* PTAS *(*resp. FPTAS*) pour le problème $\mathscr{P}$ si pour tout $\epsilon > 0$:*

- *$\mathscr{A}$ est une $(1-\epsilon)$-approximation.*
- *Pour toute instance I de $\mathscr{P}$, $\mathscr{A}$ s'exécute avec une complexité $O(f(|I|, \frac{1}{\epsilon})\ p(|I|))$ (*resp. *$O(p(\frac{1}{\epsilon})\ p(|I|))$.*

où p est un polynôme quelconque, et f une fonction calculable quelconque.
Si $\mathscr{P}$ est un problème de minimisation, $\mathscr{A}$ doit être une $(1+\epsilon)$-approximation.

Un des premiers compendiums de classes d'approximation est introduit dans [3]. S'y trouvent, entre autres, les définitions des classes APX, PTAS et FPTAS.

Définition 11 (Classes APX, PTAS, FPTAS)**.** *Soit $\mathscr{P}$ un problème d'optimisation. Les classes APX, PTAS et FPTAS sont définies par les équivalences suivantes.*

- *$\mathscr{P}$ appartient à APX $\iff$ il existe une ρ-approximation pour $\mathscr{P}$, pour un ρ fixé.*
- *$\mathscr{P}$ appartient à PTAS $\iff$ il existe un PTAS pour $\mathscr{P}$.*
- *$\mathscr{P}$ appartient à FPTAS $\iff$ il existe un FPTAS pour $\mathscr{P}$.*

Naturellement, un $FPTAS$ est aussi un $PTAS$; et si l'on fixe ϵ un $PTAS$ devient un algorithme $(1-\epsilon)$-approché. Les inclusions $FPTAS \subseteq PTAS \subseteq APX$ sont donc trivialement vérifiées. La figure suivante reprend la hiérarchie des approximations.

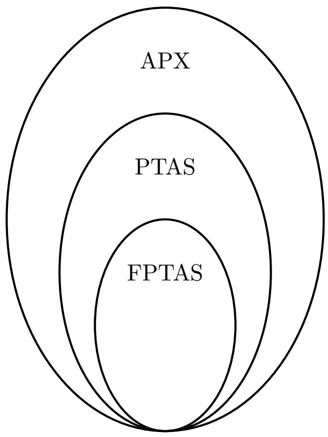

FIGURE 2.2 : Hiérarchie des approximations

En supposant que $P \neq NP$, on peut montrer que ces inclusions sont strictes, et qu'il ne peut exister de PTAS pour les problèmes APX-difficiles [33]. On donnera dans le dernier chapitre des problèmes APX et non-PTAS, ou de problème non-APX.

2.4 Extension de solution partielle

On s'intéressera, durant ce stage, à l'étude des changements de complexité et d'approximabilité des problèmes lorsqu'une solution partielle de l'instance doit être étendue ; et en particulier au seuil de densité à partir duquel ces changements peuvent être observés. Cette densité correspond à la différence de taille entre la solution partielle imposée et une solution générale. Si en effet un problème $\mathscr{P}$ donné change de complexité ou d'approximabilité lorsque la densité atteint un certain seuil, alors on parle de changement de statut de $\mathscr{P}$. Le problème consistant à étendre une solution partielle peut être qualifié de *méta*-problème, dans le sens où il désigne une problématique pouvant s'appliquer à tout problème. On peut définir formellement ce *méta*-problème d'extension.

Définition 12 (Extension de solution partielle)**.** *Soit $\mathscr{P}$ un problème d'optimisation. On définit $\mathscr{P}_{ext}$ le problème d'optimisation pour lequel les instances sont de forme (I, x), où I est une instance de $\mathscr{P}$ et x une solution partielle de I. La question pour $\mathscr{P}_{ext}$ est de trouver l'extension optimale de x à une solution de I.*

Dans la mesure où toute solution à une instance du problème d'extension contient la solution partielle imposée, optimiser la solution au problème d'extension revient à optimiser ce que l'on rajoute à la solution partielle. On appelle ce « rajout » le *résidu.*

Définition 13 (Résidu)**.** *Soit $\mathscr{P}$ un problème d'optimisation, I une instance de $\mathscr{P}$ et x une solution partielle de I. Soit S une solution de (I, x), l'instance de $\mathscr{P}_{ext}$. $R = S \backslash x$ est appelé le résidu de S, il correspond à ce que l'on doit ajouter à x pour qu'elle soit étendue à une solution. Si S est optimale, R est un résidu optimal.*

Puisque l'optimisation d'une solution au problème d'extension est équivalente à l'optimisation du résidu, on peut étendre les algorithmes d'approximation au résidu.

Définition 14 (ρ_r-approximation résiduelle)**.** *Soit $\mathscr{P}$ un problème de maximisation (*resp.* minimisation) d'un coût ω. Pour toute instance $I' = (I, x)$ de $\mathscr{P}_{ext}$, soit $R^*_{I'}$ un résidu optimal de I'. Soit $\rho_r \in]0, 1]$ (*resp.* $[1, +\infty[$). Alors un algorithme $\mathscr{A}$ est une ρ_r-approximation résiduelle pour $\mathscr{P}_{ext}$ si pour toute instance $I' = (I, x)$ de $\mathscr{P}_{ext}$:*

- *$\mathscr{A}$ s'exécute en temps polynomial.*
- *$\mathscr{A}(I')$ retournée par l'algorithme est bien une solution pour I et I'.*
- *$A_{I'}$, la valeur de $\mathscr{A}(I')$, vérifie $A_{I'} \geq \omega(x) + \rho_r \times \omega(R^*_{I'})$ (*resp.* $A_{I'} \leq \omega(x) + \rho_r \times \omega(R^*_{I'})$).*

ρ_r est appelé le ratio résiduel d'approximation *de l'algorithme.*

De manière analogue, les PTAS et FPTAS résiduelles pourront être introduites. Tout comme les classes APX, PTAS et FPTAS permettent de classifier les problèmes selon leur approximabilité, il sera possible de définir des classes de problèmes caractérisés par les approximations résiduelles pour le problème d'extension de solution leur correspondant.

L'objectif principal du stage sera d'aboutir à des *méta*-théorèmes de classification des problèmes permettant de « prédire » leur seuil de changement de statut, que ce soit sur le plan de la complexité ou sur celui l'approximabilité. On cherchera également des classes de problèmes pour lesquels le changement de statut a les mêmes implications.

Chapitre 3

État des connaissances

L'approche *méta* envisagée avec le problème d'extension de solution initiale partielle est assez inédite, aussi l'état de l'art est-il assez maigre si l'on se cantonne aux articles en lien direct avec la problématique. On propose trois exemples de changements de statut. Deux passages de P à NP-complet, et un passage de NP-complet à P.

3.1 Coloration

La problème de coloration est très largement étudié en informatique. On appelle *coloration* d'un graphe, une application de l'ensemble des sommets vers un ensemble de couleurs. Lorsque la coloration utilise k couleurs, on parle de k-coloration. Une coloration est dite *propre* si toute paire de sommets adjacents dans le graphe est colorée avec deux couleurs différentes. Le nombre minimum de couleurs nécessaires à la coloration propre d'un graphe est appelé le *nombre chromatique*, on le note $\chi(G)$ pour un graphe G donné.

k-Coloration
Donnée : $G = (V, E)$ un graphe.
Question : Existe-t-il une k-coloration propre de G ?

Le problème d'optimisation revient à trouver le nombre chromatique d'un graphe.

Nombre Chromatique
Donnée : $G = (V, E)$ un graphe.
Question : Quel est le nombre chromatique de G ?

Dans [22], Karp établit que la variante de décision de Nombre Chromatique est NP-complète, ainsi k-Coloration est NP-complet pour tout $k \geq 3$. Le problème d'optimisation est également difficile dans le cas général puisqu'il n'est pas approximable à facteur constant [35], il n'appartient donc pas à la classe APX. L'étude de la complexité et des approximations pour des restrictions à des classes de graphes particulières permet d'affiner les résultats de difficulté. Par exemple, les graphes planaires, qui peuvent être représentés dans le plan sans croisement d'arêtes, sont toujours 4-colorables [1] ; ou encore dans les graphes d'intersection de disques le nombre chromatique devient approximable à facteur constant [20]. Dans ces graphes tout sommet est un disque et deux sommets sont adjacents si et seulement si leurs disques s'intersectent. Dans les graphes bipartis, le problème devient trivial puisqu'un tel graphe est toujours 2-colorable en temps polynomial par un simple parcours.

Une variante du problème de coloration correspond à l'approche considérée dans le cadre de cette étude bibliographique. Il s'agit du problème d'extension de pré-coloration, que l'on peut voir comme une coloration avec une solution partielle imposée.

Extension de Pré-Coloration
Donnée : $G = (V, E)$ un graphe, $W \subset V$, f une k-coloration propre de $G[W]$.
Question : f peut-elle être étendue à une k-coloration de G ?

La notation $G[W]$ désigne le sous-graphe de G induit par l'ensemble de sommets W. Ce problème est NP-complet dans le cas général puisqu'il l'est déjà dans les graphes d'intervalles lorsque la k-précoloration est quelconque [6]. Chaque sommet d'un graphe d'intervalles est un intervalle de la droite réelle et deux sommets sont adjacents si et seulement si leurs intervalles se chevauchent. Nombre chromatique étant polynomial sur ces graphes [19], on a un premier exemple de changement de statut.

Le cas des graphes bipartis est aussi intéressant puisqu'il constitue un autre changement de statut. [21] prouve que l'extension de pré-coloration est NP-complète pour tout $k \geq 3$, même si l'on se restreint aux graphes bipartis et même lorsque chaque couleur n'est utilisée qu'une fois dans la pré-coloration.

On peut toujours étendre la k-précoloration à une $k + 2$ coloration. Il suffit pour cela d'utiliser 2 nouvelles couleurs pour colorer $G[V \backslash W]$, ce qui est toujours possible puisque $G[V \backslash W]$ est un graphe biparti. Une telle coloration peut être calculée en temps polynomial, on a donc un algorithme polynomial qui renvoie une solution utilisant $k+2$ couleurs, là où l'optimal est au mieux k. On a donc une $\frac{k+2}{k} = 1 + \frac{2}{k}$ approximation du problème de minimisation. Ce ratio est maximal lorsque $k = 3$, qui nous donne une $\frac{5}{3}$-approximation. Le changement de statut ne concerne donc que la complexité dans ce cas, le problème restant approximable à facteur constant.

3.2 Plus Court Chemin

Un deuxième exemple de changement de statut est celui du problème de plus court chemin élémentaire dans les graphes orientés. Un chemin est dit *élémentaire* s'il ne passe qu'une fois par chaque sommet qu'il contient. Ce problème est défini comme suit.

Plus Court Chemin Élémentaire
Donnée : $G = (V, A)$ un graphe orienté pondéré, s, t deux sommets de V.
Question : Quel est le plus court chemin élémentaire de s à t ?

Dijkstra a proposé un algorithme polynomial [14] pour le résoudre, le problème de décision associé appartient donc à la classe P. Nous proposons une variante dans laquelle des arêtes sont imposées en guise de solution partielle initiale [9].

Plus Court Chemin Élémentaire avec Étapes
Donnée : $G = (V, A)$ un graphe orienté pondéré, s, t deux sommets de V, $S \subset A$.
Question : Quel est le plus court chemin élémentaire de s à t, passant par S ?

Chemin Élémentaire avec Étapes
Donnée : $G = (V, A)$ un graphe orienté pondéré, s, t deux sommets de V, $S \subset A$.
Question : Existe-t-il un chemin élémentaire de s à t passant par S ?

Nous montrons dans [9] que le problème de décision Chemin Élémentaire avec Étapes, consistant à déterminer l'existence d'un chemin élémentaire entre deux sommets passant par des arêtes données, est NP-complet même lorsque $|S| = 1$. Puisque décider de l'existence du chemin est difficile, Plus Court Chemin Élémentaire avec Étapes l'est au moins autant et il n'est pas polynomialement approximable, quel que soit le ratio. En effet, si c'était le cas, on pourrait décider de l'existence en temps polynomial, ce qui est absurde. Le changement de statut s'opère donc à la fois au niveau de la complexité (P à NP-complet) et au niveau de l'approximabilité (les problèmes polynomiaux peuvent être vus comme 1-approximables).

3.3 Voyageur de Commerce

Enfin, le problème du Voyageur de Commerce est un exemple de changement de statut dans l'autre sens. Voyageur de Commerce est un classique de l'optimisation combinatoire et on propose une formulation en un problème de graphe, se basant sur la notion de *cycle Hamiltonien*. On appelle *cycle Hamiltonien* d'un graphe un cycle passant une et une seule fois par tous les sommets du graphe.

Voyageur de Commerce
Donnée : $G = (V, E)$ un graphe complet pondéré.
Question : Quel est le cycle Hamiltonien de poids minimum ?

Le problème de décision associé au Voyageur de Commerce fait partie des 21 problèmes NP-complets de Karp [22]. Le problème d'optimisation est donc également difficile. Il n'est, dans le cas général, pas approximable à facteur constant [30], donc n'appartient pas à APX. Dans le cas où la pondération respecte l'inégalité triangulaire, c'est à dire que pour tout triplet de sommets (x, y, z) on peut vérifier $d(x, y) + d(y, z) \geq d(x, z)$, le problème devient approximable à un facteur $\frac{3}{2}$ en utilisant l'algorithme de Christofides [11]. Une variante de Voyageur de Commerce avec un ensemble d'arêtes imposées comme solution partielle initiale est proposée dans [9].

P-Voyageur de Commerce
Donnée : $G = (V, E)$ un graphe complet pondéré, $P \subset E$.
Question : Quel est le cycle Hamiltonien de poids minimum passant par P ?

Il est montré, toujours dans [9], que lorsqu'il reste $O(log(|V|))$ arêtes à trouver pour compléter P en un cycle Hamiltonien de G, il est possible de réduire l'instance de P-Voyageur de Commerce à une instance équivalente de Voyageur de Commerce sur un graphe $G' = (V', E')$ tel que $|V'| = O(log(|V|))$. Puisqu'il existe une programmation dynamique permettant de résoudre toute instance $G = (V, E)$ de Voyageur de Commerce avec une complexité $O(2^{|V|}|V|^2)$ [5], on peut calculer la tournée optimale sur G' en temps $O(|V'|log(|V'|)^2)$ qui est une complexité polynomiale. On a donc un changement de statut pour le problème Voyageur de Commerce lorsque le nombre d'arêtes imposées est de la forme $|V| - k\ log(|V|)$ pour tout $k \geq 0$.

[9] propose également une adaptation de l'algorithme de Christofides pour montrer l'appartenance à APX du problème P-Voyageur de Commerce dans le cas où les distances respectent l'inégalité triangulaire. Le ratio de $\frac{3}{2}$ est conservé pour la solution totale, mais le ratio résiduel est dégradé à 3.

3.4 Conclusion

En ne considérant que trois problèmes et les problèmes d'extension de solutions partielles associés, on peut déjà observer des comportements différents.

Coloration dans les graphes bipartis est un problème polynomial, qui devient NP-complet quand des sommets sont précolorés avec au moins 3 couleurs. Il devient en revanche APX, dans la mesure où une $\frac{5}{3}$-approximation triviale existe. Le seuil est ici très bas puisque le statut change dès 3 sommets précolorés.

Plus Court Chemin Élémentaire dans les graphes orientés devient NP-complet dès qu'on impose une arête de passage obligatoire ; et il n'est pas approximable, quel que soit le ratio, ce qui en fait un problème d'optimisation non-APX, donc très difficile. Là encore le seuil de changement de statut est très bas.

Enfin, Voyageur de Commerce est NP-complet, et $\frac{3}{2}$-approximable dans le cas métrique. Sa variante avec un ensemble d'arêtes imposées reste NP-complète, et approximable dans le cas métrique. Le ratio d'approximation est, par contre, dégradé sur le résidu et on peut se demander si cette dégradation sera inévitable quel que soit le problème, et sa variante avec solution partielle imposée, considéré.

On a donc déjà trois changements de statut différents, et ils apparaissent dans les cas généraux des problèmes. Les résultats obtenus sur Coloration ou Voyageur de Commerce montrent que restreindre les classes d'instances peut altérer la complexité ou l'approximabilité d'un problème. On peut émettre l'hypothèse que ces restrictions d'instances auront des conséquences analogues quand on considèrera les problèmes d'extension de solution initiale.

Le chapitre suivant expose les problèmes qui seront étudiés durant le stage, ainsi que des exemples de preuves de changement de statut ou d'approximabilité résiduelle.

Chapitre 4

Approche et poursuite

L'objectif étant d'établir des théorèmes généraux de classification des problèmes en fonction de leur comportement lorsqu'une solution partielle doit être étendue, on va dans un premier temps, traiter des problèmes différents et montrer des résultats sur leur éventuel changement de statut. Une fois ces problèmes traités, on essayera de dégager les caractéristiques communes aux problèmes pour lesquels les changements de statut sont similaires. On propose dans ce chapitre un exemple de preuve de changement de statut et une preuve d'approximation résiduelle. Ces preuves illustrent la démarche qui pourra être adoptée lors de l'étude de l'extension de solution partielle pour un problème. Ensuite, les perspectives de poursuite du stage seront présentées, en exposant quelques problèmes NP-complets ayant des comportements différents vis-à-vis de l'approximation qui serviront comme support de base à l'établissement des *méta*-résultats recherchés.

4.1 Exemples d'approche

Plus Court Chemin Élémentaire

On présente ici la preuve de difficulté de Plus Court Chemin Élémentaire avec Étapes, lorsque $|S| = 1$. La preuve consiste en une réduction depuis le problème de cheminement sommets-disjoint entre paires de sommets, qui est bien NP-complet [28], vers le problème de décision Chemin Élémentaire avec Étapes.

> Chemins sommets-disjoints entre paires
> **Donnée :** $G = (V, A)$ un graphe orienté, (x, y) et (z, t) deux paires de sommets.
> **Question :** Existent-ils deux chemins sommets-disjoints, l'un allant de x à y, l'autre de z à t ?

On peut supposer dans ce problème que les chemins recherchés sont élémentaires, puisque, le cas échéant, il est possible d'éliminer rapidement les circuits d'un chemin non-élémentaire et obtenir ainsi un chemin ne passant pas par d'autres sommets.

Théorème 1. CHEMIN ÉLÉMENTAIRE AVEC ÉTAPES *est NP-complet.*

Démonstration. Soit $I = (G, x, y, z, t)$ une instance de CHEMINS SOMMETS-DISJOINTS ENTRE PAIRES sur un graphe $G = (V, A)$. On construit le graphe $G' = (V, A')$ où $A' = (A \backslash \{ya \in A\}) \cup \{yz\}$ - les arcs sortants de y sont tous retirés du graphe et on y rajoute l'arc yz. On définit $I' = (G', x, t, \{yz\})$ une instance de CHEMIN ÉLÉMENTAIRE AVEC ÉTAPES, dans laquelle on cherche à aller de x à t en passant par l'arc yz.

Cette construction est clairement polynomiale. Montrons à présent que I est une instance positive si et seulement si I' est une instance positive.

- Si I est positive, alors il existe P et Q deux chemins sommets-disjoints reliant respectivement x à y et z à t. Dans ces conditions, le chemin $T = P.yz.Q$ est bien un chemin élémentaire de x à t passant par yz dans G', I' est donc bien positive.
- Si I' est positive, alors il existe un chemin élémentaire T de x à t passant par yz. On peut l'écrire $T = P.yz.Q$ avec $P = (x, \ldots, y)$ et $Q = (z, \ldots, t)$. P et Q sont deux chemins sommets-disjoints dans G puisque T est élémentaire, P reliant x et y, Q reliant z et t. I est donc positive.

On a exhibé une réduction polynomiale de CHEMINS SOMMETS-DISJOINTS ENTRE PAIRES vers CHEMIN ÉLÉMENTAIRE AVEC ÉTAPES, ce dernier est donc NP-difficile. L'appartenance à NP étant triviale, on a aussi la NP-complétude, d'où le théorème.

□

Voyageur de Commerce Maximum

L'algorithme de Serdyukov [24] est une approximation en temps polynomial pour le problème MAX-VOYAGEUR DE COMMERCE dans le cas général.

> MAX-VOYAGEUR DE COMMERCE
> **Donnée :** $G = (V, E, \omega)$ un graphe complet pondéré.
> **Question :** Quel est le cycle Hamiltonien de poids maximum ?

On propose une adaptation de cet algorithme garantissant un rapport d'approximation de $\frac{1}{2}$ sur le résidu pour P-MAX-VOYAGEUR DE COMMERCE.

> P-MAX-VOYAGEUR DE COMMERCE
> **Donnée :** $G = (V, E, \omega)$ un graphe complet pondéré, $P \subset E$.
> **Question :** Quel est le cycle Hamiltonien de poids maximum passant par P ?

L'algorithme de Serdyukov nécessite que le graphe ait un nombre pair de sommets, mais une simple transformation permet de l'utiliser quand ce nombre est impair [24].

Remarque 1. *On appelle* couplage*, dans un graphe, un sous-ensemble d'arêtes telles que leurs extrémités soient deux à deux disjointes. On peut considérer, sans perte de généralité, que dans une instance* (G, ω, P) *de* P-MAX-VOYAGEUR DE COMMERCE*,* P *est un couplage. Si ce n'est pas le cas, alors il existe un chemin* $A \subset P$*. On peut alors supprimer de* G *tous les sommets qui ne sont pas les extrémités de* A *et créer une arête* e *de poids* $\omega(A)$ *entre les extrémités. Imposer* e *est alors équivalent à imposer* A*.*

Algorithme 1 : Serdyukov adapté

Données : $G = (V, E, \omega)$ un graphe complet pondéré, $P \subset E$
Résultat : Un cycle Hamiltonien couvrant P
début
  $\mathscr{C} \leftarrow \{C_1, \ldots, C_s\}$, une 2-factorisation de poids maximum de G;
  $\mathscr{M} \leftarrow$ P;
  pour chaque $i \in \{1, \ldots, s\}$ **faire**
    Choisir $e \in C_i \backslash P$ telle que $\mathscr{M} \cup \{e\}$ reste une union de chaines ;
    $\mathscr{M} \leftarrow \mathscr{M} \cup \{e\}$;
    $C_i \leftarrow C_i \backslash \{e\}$;
  fin
  Compléter $\mathscr{M}$ et $\mathscr{C}$ en T_1 et T_2 cycles Hamiltoniens ;
  retourner T_1 *si* $\omega(T_1) \geq \omega(T_2)$, T_2 *sinon*;
fin

On appelle *2-factorisation* d'un graphe une couverture des sommets par des cycles sommet-disjoints. [32] réduit la recherche d'une *2-factorisation* à la recherche d'un *couplage parfait.* Un couplage est *parfait* s'il contient tous les sommets du graphe. Un couplage parfait est en particulier *maximum* en nombre d'arêtes. Rechercher un couplage parfait de poids maximal revient donc à chercher un *couplage maximum* de poids maximal, et il existe des algorithmes polynomiaux pour trouver de tels couplages [27]. On peut donc trouver une *2-factorisation* de poids maximal d'un graphe en temps polynomial.

Remarque 2. *On peut changer le poids de tout $e \in P$ en un poids arbitrairement grand, de sorte que toute 2-factorisation de poids maximum de G contienne entièrement P. De plus, lorsque dans l'algorithme 1 on choisit une arête d'un cycle de la 2-factorisation, on peut toujours choisir une arête qui n'appartient pas déjà au couplage ; on peut ainsi garantir qu'en sortie de boucle $\mathscr{M}$ et $\mathscr{C}$ contiennent toujours P. La tournée renvoyée par l'algorithme est donc bien un cycle Hamiltonien utilisant toutes les arêtes de P.*

Théorème 2. *Avec la transformation proposée en Remarque 2, l'Algorithme 1 est une $\frac{1}{2}$-approximation résiduelle pour P-Max-Voyageur de Commerce.*

Démonstration. Notons OPT le poids d'une tournée optimale et $\mathscr{C}_0$ la 2-factorisation initialement calculée. Un cycle Hamiltonien étant une 2-factorisation particulière, on a $\omega(\mathscr{C}_0) \geq OPT$ puisque $\mathscr{C}_0$ est de poids maximum. Au cours de l'algorithme des arêtes de $\mathscr{C}$ sont transférées à $\mathscr{M}$, les inégalités suivantes sont donc vérifiées à toute étape :

$$\omega(\mathscr{C}) + \omega(\mathscr{M}) \geq \omega(\mathscr{C}_0) + \omega(P) \geq OPT + \omega(P) \tag{4.1}$$

On peut décomposer OPT comme étant la somme du poids de P et d'un résidu optimal : $OPT = \omega(P) + \omega(R^*)$, puis par l'équation (4.1) il vient :

$$\omega(\mathscr{C}) + \omega(\mathscr{M}) \geq 2\ \omega(P) + \omega(R^*) \tag{4.2}$$

Enfin, on a $\omega(T_1) + \omega(T_2) \geq \omega(\mathscr{C}) + \omega(\mathscr{M})$, et ainsi par l'équation (4.2) : $max\{\omega(T_1), \omega(T_2)\} \geq \omega(P) + \frac{1}{2}\ \omega(R^*)$ d'où l'approximation résiduelle de $\frac{1}{2}$.

□

4.2 Poursuite du stage

Problème	**Complexité**	**APX ?**	**PTAS ?**
Arbre de Steiner	NP-difficile [22]	ln(4) [7]	NON [10]
Euclidien	NP-difficile [22]	$\frac{2}{\sqrt{3}}$ [15]	OUI [2]
Arbre Couvrant Feuillu	NP-difficile [18]	$\frac{1}{2}$ [31]	NON [17]
Bin Packing	NP-difficile [18]	$\frac{3}{2}$ [34]	NON[18]
Max-Voyageur de Commerce	NP-difficile	$\frac{4}{5}$ [16]	?
Métrique	NP-difficile	$\frac{7}{8}$ [25]	?
Min-Voyageur de Commerce	NP-difficile [18]	NON [30]	$\times$
Métrique	NP-difficile [18]	$\frac{3}{2}$ [11]	NON [23]
Euclidien	NP-difficile [29]	$\frac{3}{2}$ [11]	OUI [2]
Nombre chromatique	NP-difficile [22]	NON [35]	$\times$
Dans les bipartis	P	1	$\times$
Plus Court Chemin	P	1	$\times$

Le tableau ci-dessus récapitule des problèmes que l'on a tenté ou que l'on tentera de classifier, avec leur complexité et leur approximabilité. Outre les problèmes VOYAGEUR DE COMMERCE, COLORATION et PLUS COURT CHEMIN définis plus haut, on considèrera les problèmes ARBRE DE STEINER, ARBRE COUVRANT FEUILLU et BIN PACKING. Ces problèmes sont tous NP-complets et les résultats d'approximabilité sont variés. Dans les variantes euclidiennes des problèmes, les sommets des graphes sont des points d'un espace euclidien et les arêtes sont pondérées par la distance euclidienne entre les points.

Le problème de l'arbre de Steiner consiste à trouver un arbre couvrant d'un sous-ensemble des sommets d'un graphe, appelé *ensemble terminal*. Le problème d'optimisation cherche à minimiser le poids total des arêtes de l'arbre.

ARBRE DE STEINER
Donnée : $G = (V, E)$ un graphe pondéré, $T \subset V$ un ensemble terminal.
Question : Quel est l'arbre de poids minimum couvrant T ?

ARBRE COUVRANT FEUILLU recherche un arbre couvrant, d'un graphe, comportant un maximum de feuilles (*Max Leaf Spanning Tree* en anglais). Le problème est intéressant puisque le problème de l'arbre couvrant de poids maximal est polynomial, pourtant ARBRE COUVRANT FEUILLU, qui est assez proche, est NP-complet.

ARBRE COUVRANT FEUILLU
Donnée : $G = (V, E)$ un graphe.
Question : Quel est l'arbre couvrant de G ayant un maximum de feuilles ?

Enfin BIN PACKING est le problème de l'optimisation du rangement d'un ensemble d'objets de tailles différentes dans des boites de capacité fixée. On dit que le rangement des objets est *valide* si les objets sont regroupés en sous-ensemble tels que la somme des tailles des objets ne dépasse pas la capacité.

BIN PACKING
Donnée : $\{1, \ldots, n\}$ n objets de tailles entières $\{c_1, \ldots, c_n\}$, C une capacité.
Question : Quel est le rangement valide minimisant le nombre de boites utilisées ?

Les résultats d'approximabilité étant divers sur les problèmes considérés, on s'attend à des comportements différents pour les problèmes d'extension de solution associés. Lorsque les seuils de changement de statut seront établis, on cherchera des propriétés communes aux problèmes ayant des comportements similaires, et finalement on essayera de dégager des classes de problèmes pour lesquels on saura, par exemple, qu'au delà d'un certain seuil le problème d'extension de solution partielle associé devient facile à résoudre. Ou encore des classes de problèmes pour lesquels toute approximation du problème de base pourra être utilisée pour le problème d'extension de solution. Tout comme il existe des réductions polynomiales préservant l'approximabilité des problèmes [13], on tentera d'exhiber des réductions préservant les changements de statut. Ces réductions pourront permettre de généraliser les résultats obtenus pour VOYAGEUR DE COMMERCE à d'autres problèmes d'optimisation difficile, ou ceux obtenus pour PLUS COURT CHEMIN ÉLÉMENTAIRE à d'autres problèmes faciles.

4.3 Conclusion

Le *méta*-problème qui sera étudié au cours du stage est relativement peu étudié. Il pourra fournir un nouveau point de vue sur la difficulté des problèmes. Outre cet intérêt théorique, le *méta*-problème correspond à une démarche largement utilisée en programmation par contraintes. On peut citer [8] qui définit une contrainte globale de chemin, et maintenir la consistance de cette contrainte revient à résoudre le problème CHEMIN ÉLÉMENTAIRE AVEC ÉTAPES que l'on a présenté plus haut.

Le fait d'ajouter des contraintes à un problème sous la forme d'une solution partielle à étendre est assez commun lorsque l'on s'attèle à la résolution pratique d'un problème. Par exemple, les problèmes bi-niveaux sont utilisés pour la résolution de problèmes industriels pratiques [4]. Ils sont constitués de deux problèmes imbriqués et la résolution du problème principal utilise les solutions trouvées pour un problème secondaire. En un sens, la résolution du problème secondaire impose une solution initiale partielle au problème principal sous la forme de contraintes supplémentaires.

On peut donc espérer que la nouvelle classification des problèmes qui sera proposée durant ce stage pourra permettre de mieux cerner la difficulté des problèmes théoriques mais aussi la difficulté pratique de résolution de problèmes *réels*.

Bibliographie

[1] K. Appel and W. Haken. Every planar map is four colorable. part i : Discharging. *Illinois J. Math.*, 21(3) :429–490, 09 1977.

[2] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5) :753–782, 1998.

[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation : Combinatorial optimization problems and their approximability properties.* Springer-Verlag, 1999.

[4] J. F. Bard. *Practical bilevel optimization : algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

[5] R. Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)*, 9(1) :61–63, 1962.

[6] M. Biró, M. Hujter, and Zs. Tuza. Precoloring extension I. Interval graphs. *Discrete Mathematics*, 100(1–3) :267 – 279, 1992.

[7] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP-based approximation for steiner tree. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 583–592, New York, NY, USA, 2010. ACM.

[8] H. Cambazard and É. Bourreau. Conception d'une contrainte globale de chemin. *10e Journées nationales sur la résolution pratique de problemes NP-complets (JNPC'04)*, pages 107–121, 2004.

[9] A. Chateau, R. Giroudeau, J.-C. König, V. Pollet, and M. Weller. On the complexity and approximation of problems with a given partial solution. In *Combinatorial Optimization - Fourth International Symposium, ISCO 2016, Salerno, Italy, May 16-18*, ISCO, 2016. Submitted.

[10] M. Chlebík and J. Chlebíková. The steiner tree problem on graphs : Inapproximability results. *Theor. Comput. Sci.*, 406(3) :207–214, 2008.

[11] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976.

[12] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[13] P. Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 262–273. IEEE Computer Society, 1997.

[14] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271, 1959.

[15] D.-Z. Du and F. K. Hwang. An approach for proving lower bounds : solution of gilbert-pollak's conjecture on steiner ratio. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 76–85. IEEE, 1990.

[16] S. Dudycz, J. Marcinkowski, K. E. Paluch, and B. Rybicki. A 4/5 - approximation algorithm for the maximum traveling salesman problem. *CoRR*, abs/1512.09236, 2015.

[17] G. Galbiati, F. Maffioli, and A. Morzenti. A short note on the approximability of the maximum leaves spanning tree problem. *Inf. Process. Lett.*, 52(1) :45–49, 1994.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[19] M. C. Golumbic. Interval graphs. In *Algorithmic Graph Theory and Perfect Graphs*, pages 171 – 202. Academic Press, 1980.

[20] A. Gräf, M. Stumpf, and G. Weißenfels. On coloring unit disk graphs. *Algorithmica*, 20(3) :277–293, 1998.

[21] M. Hujter and Zs. Tuza. Precoloring extension II. Graph classes related to bipartite graphs. *Acta Mathematica Universitatis Comenianae*, 62(1) :1–11, 1993.

[22] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[23] M. Karpinski, M. Lampis, and R. Schmied. New inapproximability bounds for tsp. In *Algorithms and Computation : 24th International Symposium, ISAAC 2013, Hong Kong, China, December 16-18, 2013, Proceedings*, pages 568–578, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[24] A.V. Kostochka and A.I. Serdyukov. Polynomial algorithms with the estimates 3/4 and 5/6 for the traveling salesman problem of the maximum. *Upravlyaemye Sistemy*, 26 :55–59, 1985.

[25] L. Kowalik and M. Mucha. Deterministic 7/8-approximation for the metric maximum TSP. *Theor. Comput. Sci.*, 410(47-49) :5000–5009, 2009.

[26] R. E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1) :155–171, January 1975.

[27] L. Lovász and M. D. Plummer. *Matching theory.* Elsevier, 1986.

[28] T. Ohtsuki. *The two disjoint path problem and wire routing design*, pages 207–216. Springer Berlin Heidelberg, Berlin, Heidelberg, 1981.

[29] C. H. Papadimitriou. The euclidean travelling salesman problem is np-complete. *Theoretical Computer Science*, 4(3) :237 – 244, 1977.

[30] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1) :1–11, 1993.

[31] R. Solis-Oba. *2-Approximation Algorithm for Finding a Spanning Tree with Maximum Number of Leaves*, pages 441–452. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[32] W. T. Tutte. A short proof of the factor theorem for finite graphs. *Canad. J. Math*, 6(1954) :347–352, 1954.

[33] V. V. Vazirani. *Approximation algorithms.* Springer-Verlag, 2001.

[34] A. N. Zehmakan. Bin packing problem : Two approximation algorithms. Unpublished, 2015.

[35] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1) :103–128, 2007.