

Table des matières

Introduction générale	6
I Problème de transport sélectif	8
Introduction	9
1 Problème de tournée de véhicules sélectif	11
1.1 Définition du problème	11
1.2 Quelques variantes du TOP	13
1.2.1 Le problème de tournées sélectif multipériodiques- m TOP	13
1.2.2 Le problème de tournées sélectif avec fenêtres de temps-TOPTW	13
1.2.3 Le problème de tournées sélectif avec avec profits et contraintes de capacité-CTOP	14
1.3 Terminologie et propriétés	14
1.3.1 Terminologie	15
1.3.2 Propriétés	16
1.4 Formulation linéaire	18
1.4.1 Modélisation du TOP à deux dépôts sous forme d'un problème sur un graphe	18
1.4.2 Modélisation sous forme d'un programme linéaire en nombres entiers	19
1.5 Conclusion	20
2 La résolution du problème de tournées de véhicules sélectif	21
2.1 Une approche basée sur un algorithme de cutting-plane	21

2.1.1	Élimination des sous-tours	22
2.1.2	Propriétés de dominance	23
2.1.3	Les inégalités valides basées sur les incompatibilités	28
2.1.4	Algorithme de cutting-plane :	31
2.2	La résolution de TOP grâce à l’algorithme génétique	35
2.2.1	Codage utilisé :	35
2.2.2	Création de la population initiale :	36
2.2.3	Sélection :	37
2.2.4	Croisement :	37
2.2.5	Mutation :	39
2.2.6	Résultats Numériques :	40
2.3	Conclusion :	42
 II Dimension des ordres bipartis		44
Introduction		45
 3 Ensembles ordonnés		47
3.1	Relation d’ordre	47
3.1.1	Minorants, majorants, minimaux et maximaux	49
3.1.2	Diagramme de Hasse d’un ordre	49
3.1.3	Isomorphisme et dualité	50
3.1.4	Chaînes, Antichaînes et paramètres fondamentaux	51
3.2	Conclusion	52
 4 Dimension des ordres bipartis		53
4.1	Le théorème de Dilworth	53
4.2	Extensions et générateurs	56
4.3	Dimension d’un ordre	57
4.3.1	Ordres de dimension deux	59
4.4	Ensemble ordonné biparti :	62
4.5	Les algorithmes :	64
4.5.1	Problèmes, algorithmes, complexité	64
4.5.2	Les problèmes P, NP, et NP-complets :	65
4.6	Conclusion :	68

Conclusion générale	69
Bibliographie	70

Introduction générale

De plus en plus souvent, les informaticiens, les ingénieurs de production et les gestionnaires sont confrontés à des problèmes combinatoires. Ces problèmes ont un nombre de solutions fini mais très grand, de sorte qu'une résolution par énumération des solutions est inconcevable en pratique. Avec l'augmentation de la concurrence, la pression se fait plus forte non seulement pour trouver une solution qui satisfait les contraintes et soit raisonnablement performante, mais encore pour trouver la meilleure solution possible. C'est notamment le cas en télécommunications où le développement des transports et des communications pose des problèmes dont la taille croît très rapidement et où les enjeux financiers sont considérables. D'une manière plus générale, la phase de l'informatisation de base des administrations publiques et des entreprises est largement entamée et des préoccupations d'amélioration et d'optimisation des processus se font jour.

Face à l'apparition des problèmes d'optimisation de plus en plus complexes, les méthodes de résolution classiques se sont trouvées impuissantes à les résoudre. Dans ce contexte, l'élaboration des heuristiques ou méthodes dites approchées, c'est-à-dire déterminant une solution satisfaisante non obligatoirement optimale, constitue un passionnant axe de recherche. Ces méthodes sont apparues face à l'impuissance des méthodes dites exactes, c'est à dire déterminant la meilleure solution admissible existante, à résoudre des problèmes NP-difficiles de taille importante. Ainsi, dans les années 1960 et 1970, ont été conçues des heuristiques consacrées à différents problèmes particuliers d'optimisation combinatoire (voyageur de commerce, tournées de distribution, ordonnancement de production ...). Ce mémoire est composé de quatre chapitres scindés en deux parties :

- La première partie traite le problème de tournée de véhicules sélectif (TOP), et comprend deux chapitres. Nous commençons par la définition du problème, ensuite,

nous décrivons un modèle de notre problème sous forme d'un programme linéaire en nombres entiers ainsi qu'un autre sous forme d'un problème sur un graphe, puis nous présentons deux méthodes pour la résolution du TOP ainsi que le principe général et ses caractéristiques. A la fin, nous présentons une comparaison analytique de notre méthode approchée avec celle proposée par Hao et Miller-Hooks.

- La deuxième partie est consacrée au dimension des ordres bipartites, et comprend aussi deux chapitres. D'abord en commençant par donner les notions fondamentales utilisées pour décrire, étudier, raisonner sur les ensembles ordonnés, puis on définit la dimension des ensembles ordonnés, et après donner quelques propriétés et caractéristiques de dimension des ordres bipartis et on s'intéresse plus particulièrement aux ordres bipartis de dimension 2. En fin nous prouverons qu'un ordre est de dimension k , pour $k \geq 3$, est NP-complet.

Première partie

Problème de transport sélectif

Rapport-gratuit.com 
LE NUMERO 1 MONDIAL DU MÉMOIRES

Introduction

Les problèmes de tournées de véhicules (VRP) ont été le sujet d'une recherche intensive durant plus de cinquante ans, est l'un des problèmes les plus étudiés d'optimisation combinatoire [3]. Cet engouement peut en partie s'expliquer par les applications directes dans le domaine de la logistique, et à leur grande difficulté. Le VRP implique la planification des routes de livraison à moindre coût, afin de servir un ensemble de clients dispersés géographiquement, tout en respectant les contraintes. Il existe différentes extensions des problèmes de tournées prenant en compte des objectifs et contraintes opérationnelles des contextes d'applications visée.

Notre étude consiste à traiter un problème de transport appelé Problème de tournées de Véhicules Sélectif PTVS ou Problème de mTournées Sélectives PmTS, a été d'abord mentionné pour la première fois, dans un article de Butt et Cavalier (1994) [24] sous la dénomination "The Multiple Tour Maximum Collection Problème" (MTMCP). Il s'agit d'une variante du problème de tournées de véhicules (VRP) connue encore sous le nom "Team Orienteering Problem" TOP, qui a été officiellement introduit plus tard dans [9]. Plus précisément, le problème de tournée de Véhicule Sélectif présente une variante d'un ensemble de problèmes, de nature bi-objectifs, d'une classe appelée les Problèmes de Tournées avec Gains PTG. Le TOP consiste à organiser les visites pour la flotte afin de maximiser la somme des profits collectés tout en respectant un temps de trajet limite L imposé pour chaque véhicule. La nature bi-objectifs induit plusieurs variantes qui peuvent figurer dans cette classe en jouant sur la considération des deux objectifs déjà mentionnés. Parmi ces variantes : Le Problème de Tournées Sélectives TOP consiste à maximiser le profit récolté par tous les véhicules et à limiter, au niveau des contraintes, le temps de trajet L imposé pour chaque véhicule.

Le problème de tournée de Véhicule Sélectif est un problème NP-difficile [9]. Face à ce

constat, notre objectif de recherche consiste à résoudre le problème de tournée de Véhicule Sélectif par une méthode approchée. Ce mémoire présente les résultats de notre travail de recherche et comprend deux chapitres.

Dans le premier chapitre, nous présentons une description complète du problème de tournée de véhicules et ses variantes. Ensuite, nous présentons la terminologie et quelques propriétés qui caractérisent ce problème. A la fin, nous décrivons deux formes de modélisation du TOP à deux dépôts, la première est une modélisation sous forme d'un graphe et la deuxième consiste à une modélisation en programme linéaire en nombres entiers binaires.

Dans le deuxième chapitre, nous nous intéressons à la résolution de problème de tournée de Véhicule Sélectif. Nous proposons deux méthodes, la première est une méthode exacte basée sur l'algorithme de cutting-plane, et la deuxième est une méthode approchée basée sur l'algorithme génétique. Les résultats numériques obtenus par la méthode approchée afin de les comparer aux résultats obtenus par les résultats obtenus par Hao et Miller-Hooks obtenus par l'heuristique recherche TABOU.

Problème de tournée de véhicules sélectif

Dans ce chapitre, nous présentons la définition du problème de tournée de Véhicule Sélectif avec ses variantes. Ensuite, nous présentons les caractéristiques du TOP ainsi que la terminologie utilisée lors de sa description. Et en fin, nous décrivons un modèle pour le problème de tournée de Véhicule Sélectif sous forme d'un programme linéaire en nombres entiers ainsi qu'un autre sous forme d'un problème sur un graphe.

1.1 Définition du problème

Le problème de tournée de Véhicule Sélectif est une variante de VRP où à priori il n'est pas possible de servir tous les clients. En TOP un nombre limité de véhicule est disponible pour visiter un ensemble potentiel des clients, où chaque véhicule doit partir d'un dépôt et revenir en un autre après avoir visité un ensemble de clients.

Sa solution consiste à construire un ensemble de m tournées associées à m véhicules contenant chacune les points \mathbf{d} et \mathbf{a} comme point de départ et point d'arrivée. Les m tournées sont identifiées de telle sorte à respecter le temps de trajet limite L et le profit total récolté par l'ensemble des m tournées soit maximisé. Dans notre cas la capacité des véhicules n'est pas prise en considération vu que l'on considère que l'on fournit un service aux clients. Chaque client ne peut être desservi qu'une seule fois et par au plus un seul véhicule. Pour cette raison, ces problèmes sont les plus utilisés dans les domaines industriels où les ressources humaines et matérielles ont souvent certaines limites par rapport à la demande des clients à un moment donné, à savoir le nombre d'employés, le temps de travail de chaque employé, etc. tout en respectant ces limites.

L'une des caractéristiques du TOP est qu'une solution peut avoir des clients non desservis

1.1. Définition du problème

qui le seront éventuellement ultérieurement (voir figure (1.1)). Cette caractéristique est due d'une part à la limitation du nombre de véhicules et d'autre part à la limitation de chaque tournée par une longueur maximale.

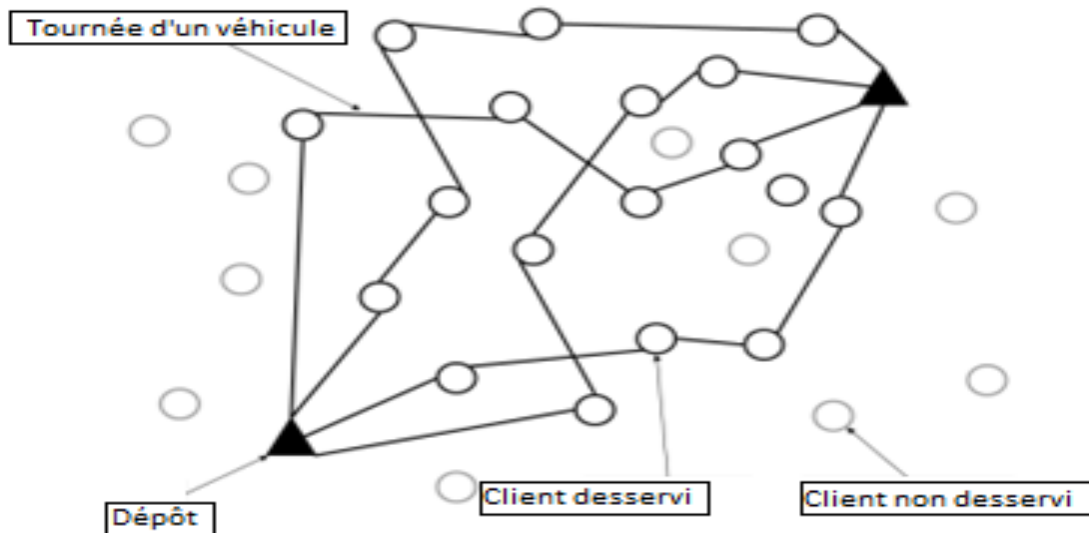


Figure 1.1 – Exemple d'une solution du TOP à deux dépôts avec 4 véhicules

Le TOP peut être défini avec un dépôt unique, c'est à dire que chaque véhicule part du dépôt et y revient après la visite d'un ensemble de clients. La variante du TOP à un seul dépôt est un cas particulier de la variante à deux dépôts. En effet, il suffit de mettre une distance nulle entre les deux dépôts (voir figure (1.2)).

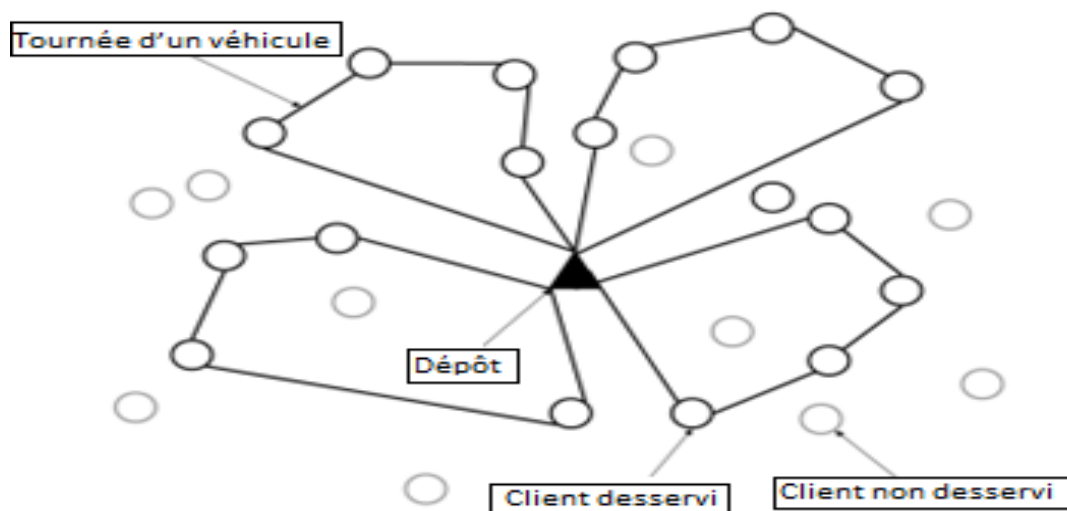


Figure 1.2 – Exemple d'une solution du TOP à un seul dépôt avec 4 véhicules

Un autre cas particulier du problème correspond au cas où nous considérons qu'un seul véhicule et un seul dépôt, cette variante revient à résoudre un problème connu sous

le nom du Problème du Voyageur de Commerce Sélectif PVCS ([7], [17] et [25]).

1.2 Quelques variantes du TOP

Nous définissons dans cette section quelques variantes du problème de tournée de véhicule sélectif les plus connues dans la littérature.

1.2.1 Le problème de tournées sélectif multipériodiques- m TOP

Le problème de tournées de véhicules sélectif multipériodique (m TOP) a été récemment présenté par Zhang et al. [16] comme une nouvelle variante du problème de tournées sélectif (TOP). Dans le cas de m TOP, toutes les caractéristiques de TOP mentionnées dans la section précédente sont prises en compte, avec quelques contraintes supplémentaires à respecter. De même, tous les clients ne doivent être visités. Un nombre limité de véhicules identiques est disponible pour visiter ces clients. Chaque véhicule est associé à un temps de trajet limite prédéfinie et deux dépôts particuliers au cours de laquelle la visite doit commencer et terminer. Chaque client est associé à un profit qui est recueillie au plus une fois par la flotte de véhicules. De plus, chaque circuit est divisé en un certain nombre de périodes ayant chacune son propre délai de trajet maximal. Chaque période doit se terminer à un certain client, à partir de laquelle la période suivante commence. Par exception, la première période de chaque tournée doit commencer à partir du dépôt et le dernier doit se terminer à l'autre dépôt. Ainsi, le but de m TOP est de maximiser les profits recueillis auprès des clients servis dans les périodes sans dépasser la limite de temps de trajet de chaque période.

1.2.2 Le problème de tournées sélectif avec fenêtres de temps-TOPTW

Le problème de tournées sélectives avec fenêtres de temps (Team Orienteering Problem with Time Windows-TOPTW), est une variante bien connue de TOP. Afin de respecter la disponibilité de certains clients, Kantor et Rosenwein [9] ont présenté cette variante de TOP pour résoudre ce problème. Dans cette variante, la flotte de véhicules

est disponible pour visiter un ensemble de clients, tout en respectant le temps de trajet limite L imposé pour chaque véhicule. De même que le TOP, le profit de chaque client peut être recueillie au plus une fois par la flotte, où La disponibilité de chaque client i est représentée par une fenêtre de temps définie par la date de service au plus tôt e_i et celle au plus tard l_i . Un véhicule arrivant au client i avant la date e_i doit attendre jusqu'à l'ouverture de la fenêtre de temps, alors qu'une arrivée tardive après l_i n'est pas permise et rend la solution irréalisable. La résolution du TOPTW consiste à organiser les visites pour tous les véhicules afin de maximiser la somme des profits collectés tout en respectant les fenêtres de temps des clients servis et le temps de trajet limite L imposé pour chaque véhicule.

1.2.3 Le problème de tournées sélectif avec avec profits et contraintes de capacité-CTOP

Le problème de tournées de véhicules avec profits et contraintes de capacité (Capacitated Team Orienteering Problem-CTOP) est une variante du Problème de Tournées de Véhicules où à priori il n'est pas possible de servir tous les clients à cause de certaines contraintes de limitation de ressources. On dispose d'une flotte de véhicules identiques ayant chacun une certaine limite de capacité et de longueur à parcourir. Un ensemble de clients, ayant chacun son propre profit et une demande de service, est réparti géographiquement autour d'un dépôt.

La résolution du CTOP consiste à construire des tournées de clients choisis de telle sorte à maximiser la somme des profits collectés tout en respectant les capacités des véhicules et les longueurs de trajet limites. Le CTOP apparaît principalement dans les problèmes de livraison à domicile, où les transporteurs disposent d'un nombre limité de véhicules ayant chacun une limite de capacité et une distance maximale à respecter.

1.3 Terminologie et propriétés

Afin de modéliser et de décrire les modèles du TOP, nous présentons dans cette section quelques propriétés de TOP et la terminologie que nous avons utilisée dans ce travail.

1.3.1 Terminologie

Les données d'un TOP est un réseau d'arêtes pouvant être traversées par l'ensemble de véhicules fixé, que nous pouvons représenter par un graphe. Il s'agit en fait, d'une abstraction de la réalité de sorte à permettre sa modélisation. Nous définissons dans cette section les éléments de base utiles à l'encodage d'un graphe et ainsi la définition d'une solution du TOP.

Un graphe G orienté est un ensemble de sommets (ou de nœuds) V et d'arêtes E . Il s'agit d'une abstraction d'un lieu tel une ville, une division administrative, une intersection routière ou une infrastructure de transfert (stations, terminus, ports et aéroports). L'arête $\{i, j\}$ est caractérisée par deux sommets i et j . Une arête est une représentation abstraite d'infrastructures de support des déplacements entre deux nœuds. La désignation du terme orienté consiste dans le fait que les liens entre les sommets à une direction bien déterminée.

Dans ce qui suit, nous présentons des termes propres au TOP. Soit un graphe $G = (V = \{1, \dots, n\} \cup \{d, a\}, E)$ orienté.

- $V^- = \{1, 2, \dots, n\}$ désigne l'ensemble de clients potentiels,
- d et a représentent les dépôts, nous utilisons V^d , V^a et V pour désigner respectivement les ensembles $V^- \cup \{d\}$, $V^- \cup \{a\}$ et $V^- \cup \{d, a\}$.
- $E \subseteq V \times V$ désigne l'ensemble des arcs, représentant les différentes routes reliant les clients.
- Une matrice des temps de trajets $C = \{c_{ij}\}$, de taille $(n \times n)$ permet de définir pour tout $(i, j) \in E$ le temps de trajet entre le client i et le client j , $c_{ij} = \infty$ si $(i, j) \notin E$.
- Un vecteur P , de taille n permet de définir pour tout $i \in V^-$ le gain (ou profit) récolté lors de la visite d'un client i .

Dans le cas du TOP à deux dépôts, une tournée est représentée par une chaîne où ces extrémités sont constituées par les deux sommets (ou dépôts) d et a . Pour le TOP à un seul dépôt, une tournée est représentée par un cycle partant du dépôt et revenant au même dépôt après la visite d'un ensemble de sommets.

Une solution du TOP notée Π est un ensemble de m tournées assurées par les m véhicules définis, $\Pi = \{\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_m\}$ où π_k est la chaîne (ou tournée) composée par les sommets $i \in V$ desservi par le véhicule k .

$$\pi_k = (V(\pi_k), E(\pi_k)), \quad V(\pi_k) \subseteq V, \quad E(\pi_k) \subseteq E$$

Mise à part les dépôts, un sommet (ou un client) n'est visité qu'une seule fois et par une seule tournée (ou un véhicule). Le temps de trajet limite $L(\pi_k)$ d'une tournée π_k est définie par le cumul de temps de trajet des arêtes qui la composent. Chaque tournée ne doit pas dépasser le temps de trajet limite L imposé pour chaque véhicule. Un gain (ou profit) $p(\pi_k)$ d'une tournée est la somme des profits p_i des sommets i qui la composent. Le profit total d'une solution est le profit récolté par les m tournées. Il est à noter que les dépôts d et a ont des profits nuls ($p_d = p_a = 0$).

1.3.2 Propriétés

Nous supposons qu'une solution Π du TOP vérifie les hypothèses suivantes :

- Π est définie sur un graphe orienté.
- Π peut être fournie avec un seul ou deux dépôts selon l'instance du TOP traitée.
- Chaque véhicule part du dépôt d et arrive à a .
- Le nombre des véhicules m est fixé par le décideur.
- Un véhicule n'a pas de limite de capacité.
- Chaque véhicule k est limité par un temps de trajet limite de parcours L , $\ell(\pi_k) \leq L$.
- Chaque client i peut être desservi par n'importe quel véhicule k de la flotte.
- La matrice des temps de trajet est symétrique (graphe orienté) et vérifie l'inégalité triangulaire : $\forall i, j, k \in V$, $c_{ij} = c_{ji}$ et $c_{ik} + c_{kj} \leq c_{ij}$.

La définition du TOP et les hypothèses précédentes permettent de déduire les propriétés suivantes :

Propriété 1.1. *Une solution optimale peut être composée uniquement par un sous ensemble de sommets (ou clients) sélectionnés par la méthode de résolution. Cette caractéristique de la solution du TOP a motivé l'apparition du terme "sélectif" dans le nom du problème.*

Propriété 1.2. *Pour un graphe $G(V, E)$ de taille n et une distance maximale L , la résolution du TOP peut être effectuée sur un graphe réduit $G'(V', E')$ de taille n' tel que $n' \leq n$, $V' \subseteq V$ et $\forall i \in V - V'$, $c_{di} + c_{ia} > L$.*

Propriété 1.3. *Dans une solution optimale Π^* , nous pouvons trouver un sous ensemble de k tournées vides $\Pi^v \subset \Pi^*$ tel que $\Pi^v = \{\pi_1, \dots, \pi_k\}$ et $0 \leq |\Pi^v| \leq m$. $\forall \pi_k \in \Pi^v$, $\pi^k = \{d, a\}$ et $\ell(\pi^k) = c_{1n}$.*

Preuve 1.1. En considérant la deuxième propriété et l'hypothèse de l'inégalité triangulaire, et supposant que $\forall i \in V - \{d, a\}, c_{di} + c_{ia} > L$ tous les sommets i ne peuvent pas figurer dans la solution et nous aurons toutes les tournées vides ($|\Pi^v| = m$). Pour $i \in V$ tel que $c_{di} + c_{ia} > L$, la solution peut avoir un certain nombre de tournées vides ($0 \leq |\Pi^v| < m$).

Propriété 1.4. Résoudre un TOP à deux dépôts sur un graphe non orienté de taille n revient à résoudre un TOP à un seul dépôt sur un graphe orienté dont le nombre des sommets est $(n - 1)$.

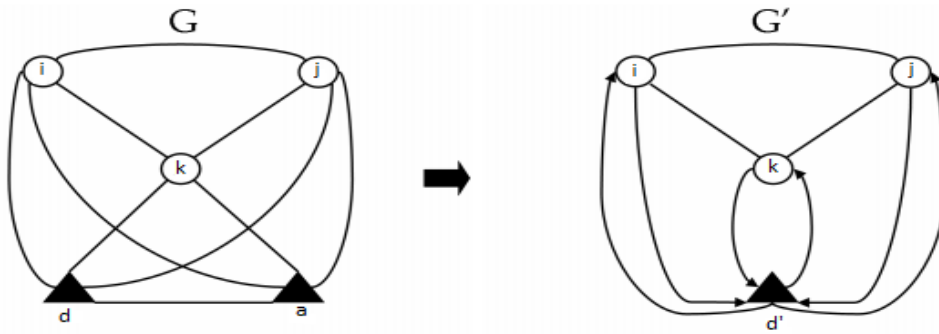


Figure 1.3 – Illustration de la propriété (1.4)

Preuve 1.2. Supposons que nous avons un graphe $G(V, A)$ non orienté ($\forall i, j \in V, c_{ij} = c_{ji}$) de taille n , d et a représentant les dépôts (voir le graphe G de la figure (1.3)). Pour se ramener à la résolution du TOP à un seul dépôt, il suffit d'annuler la distance entre les deux dépôts d et a ($c_{da} = 0$). Dans ce cas, d et a sont considérés comme un seul sommet, soit d' et nous obtenons un graphe $G'(V', E')$ avec $V' = \{d'\} \cup V \setminus \{d, a\}$ et $E' = \{(d', i), (i, d')\} \cup E \setminus \{(d, i), (i, a)\}, i \in V\}$ (voir le graphe G' de la figure (1.3)). Dans le graphe G' , les distances $c'_{d'i}$ et $c'_{id'}$ de d' par rapport aux autres sommets i (clients) et inversement sont asymétriques. Nous avons $c'_{d'i} = c_{di}$, $c'_{id'} = c_{ia}$ donc $c'_{d'i} \neq c'_{id'} \forall i \in V' \setminus \{d'\}$. De ce fait, le graphe G' est un graphe asymétrique de taille $(n - 1)$.

Propriété 1.5. Résoudre un TOP à deux dépôts revient à résoudre un TOP à un seul dépôt sur un graphe non orienté ayant $|A| + 1$ arêtes.

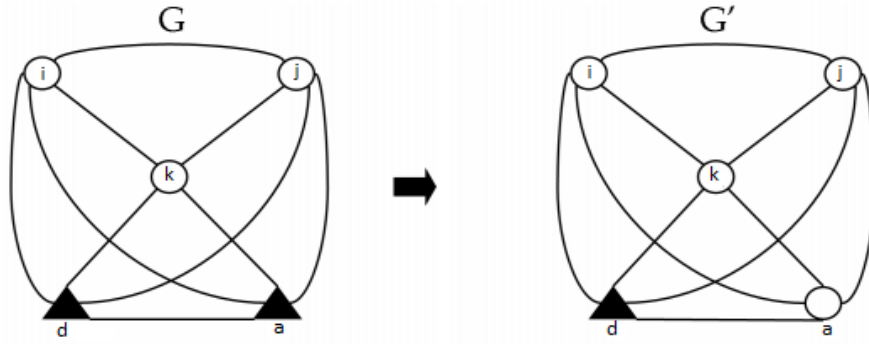


Figure 1.4 – Illustration de la propriété (1.5)

Preuve 1.3. Pour la résolution du TOP à deux dépôt d et a sur un graphe G nous avons $p_d = p_a = 0$ et $c_{da} \neq 0$. Si nous considérons que les deux dépôts d et a sont confondus c'est à dire $c_{da} = 0$ et que $p_a = \infty$ nous obtenons un nouveau graphe G' (voir figure (1.4)). Avec les conditions du graphe G' , le processus de résolution considère obligatoirement le sommet a car il possède un profit infini et il est situé à une distance nulle du dépôt d . La distance nulle entre d et a garantit l'adjacence entre ces deux sommets dans chaque tournée de la solution. Par conséquent nous aurons une solution du TOP à deux dépôts.

1.4 Formulation linéaire

Quelques définitions et notation utiles dans la modélisation de TOP ont été définis au-dessus, section(1.3).

1.4.1 Modélisation du TOP à deux dépôts sous forme d'un problème sur un graphe

Afin de pouvoir modéliser le TOP à deux dépôts sur un graphe, nous considérons les notations supplémentaires suivantes :

- $\pi_k^h = (V(\pi_k^h), E(\pi_k^h))$ est une chaîne extraite d'une tournée π_k avec $V(\pi_k^h) \subseteq V(\pi_k)$ et $E(\pi_k^h) \subseteq E(\pi_k)$, avec h l'indice sur la chaîne extraite.
- $\pi_k^h \leftrightarrow \pi_k^q$ signifie que la chaîne extraite π_k^h est non adjacente à la chaîne extraite π_k^q .
- $V(\pi_k) \cap V(\pi_h)$ est l'ensemble des sommets en intersection entre les deux tournées π_k et π_h .

Notre problème TOP est modélisé sous forme :

$$Max \sum_{k=1}^m p(\pi_k) \quad (1.1)$$

sous les contraintes :

$$V(\pi_k) \cap V(\pi_h) = \{d, a\} \quad \forall k, h \in \{1, \dots, m\}, k \neq h \quad (1.2)$$

$$V(\pi_k^h) \cap V(\pi_k^q) = \emptyset \quad \forall k \in \{1, \dots, m\}, \forall \pi_k^h, \pi_k^q \subset \pi_k, \pi_k^h \leftrightarrow \pi_k^q \quad (1.3)$$

$$\ell(\pi_k) \leq L \quad \forall k \in \{1, \dots, m\} \quad (1.4)$$

Les significations de chaque équation du modèle sont les suivantes :

L'équation (1.1) présente la fonction objectif qui consiste à maximiser la somme des profits récoltés par les m tournées. La contrainte (1.2) assure qu'un sommet n'est visité que par une seule tournée à l'exception des sommets d et a . L'équation (1.3) assure qu'un sommet n'est visité qu'une seule fois par une tournée, en d'autre terme elle permet d'éliminer les sous-tours sur une tournée. La limite imposée sur le temps de trajet est assurée par les contraintes (1.4).

1.4.2 Modélisation sous forme d'un programme linéaire en nombres entiers

Le TOP peut être formulé en programmation linéaire en nombres entiers (PLNE) avec les variables de décision y_{ir} et x_{ijr} :

$y_{ir} = 1$ si le client i est servi par le véhicule r et 0 sinon.

$x_{ijr} = 1$ si l'arc (i, j) est utilisé par le véhicule r pour visiter le client i puis le client j et 0 sinon.

$$max \sum_{i \in V^-} \sum_{r \in F} y_{ir} P_i \quad (1.5)$$

$$\sum_{r \in F} y_{ir} \leq 1 \quad \forall i \in V^- \quad (1.6)$$

$$\sum_{j \in V^a} x_{djr} = \sum_{j \in V^d} x_{jar} = 1 \quad \forall r \in F \quad (1.7)$$

$$\sum_{i \in V^a \setminus \{k\}} x_{kir} = \sum_{j \in V^d \setminus \{k\}} x_{jkr} = y_{kr} \quad \forall k \in V^-, \quad \forall r \in F \quad (1.8)$$

$$\sum_{i \in V^d} \sum_{j \in V^a \setminus \{i\}} C_{ij} x_{ijr} \leq L \quad \forall r \in F \quad (1.9)$$

$$\sum_{(i,j) \in U \times U} x_{ijr} \leq |U| - 1 \quad \forall U \subseteq V^-, |U| \geq 2, \forall r \in F \quad (1.10)$$

$$x_{ijr} \in \{0, 1\} \quad \forall i \in V, \forall j \in V, \forall r \in F \quad (1.11)$$

$$y_{ir} \in \{0, 1\} \quad \forall i \in V^-, \forall r \in F$$

La fonction objectif (1.5) consiste à maximiser la somme des profits collectés. Les contraintes (1.6) assurent que chaque client est servi au plus par un véhicule, alors que les contraintes (1.7) et (1.8) garantissent la connectivité de chaque tournée. La limite imposée sur le temps de trajet est assurée par les contraintes (1.9), les sous-tours sont interdits par les contraintes (1.10) et les contraintes (1.11) traduisent celles d'intégrité.

1.5 Conclusion

Dans ce chapitre, nous avons détaillé notre Problème TOP et ses variantes, puis nous avons présenté quelques propriétés de TOP. Et à la fin, nous avons effectué deux formes de modélisation du TOP à deux dépôts.

Dans le chapitre suivant, nous voulons résoudre le problème TOP par une méthode exacte cutting-plane et une méthode approchée basée sur l'algorithme génétique. Et ensuite, nous aurons faire une certaine comparaison de ses résultats obtenus avec les résultats obtenus par Hao et Miller-Hooks obtenus par l'heuristique recherche TABOU.

Chapitre 2

La résolution du problème de tournées de véhicules sélectif

Le TOP consiste à déterminer un ensemble de m tournées élémentaires associées à m véhicules contenant chacune les sommets d et a comme sommet de départ et sommet d'arrivée. Les m tournées sont identifiées de telle sorte à respecter le temps de trajet imposé par chaque véhicule et le profit total récolté par l'ensemble des m tournées soit maximiser. Une des caractéristiques du TOP est donc que tous les sommets peuvent ne pas être visités.

Dans ce chapitre, Nous présentons dans la première section une approche basée sur un algorithme de cutting-plane. Et dans la deuxième section nous proposons une approche basée sur les algorithmes génétique pour la résolution de ce problème et les résultats obtenus.

2.1 Une approche basée sur un algorithme de cutting-plane

Dans cette section, nous présentons une approche basée sur un algorithme de cutting-plane [18], nous définissons tout d'abord la méthode de cutting-plane, puis nous présentons les différentes opérations effectuées pour atteindre la solution. Certains calculs supplémentaires sont nécessaires lors de cet algorithme afin de générer des coupes supplémentaires qui peuvent être ajoutés au modèle pour accélérer son processus de solution.

cutting-plane : Cette méthode a été développée par Gemory [20] pour résoudre le pro-

blème linéaire en nombres entiers (PLNE). Par définition, PLNE est un programme d'optimisation où la fonction objectif, les contraintes de base et les inégalités valides sont toutes représentées comme des expressions linéaires. L'idée principale de l'algorithme de cutting-plane est d'ajouter des contraintes supplémentaires afin de réduire l'espace de recherche d'origine sans exclure la solution entière optimale.

L'objectif de cette approche est l'accélération du processus de résolution alors l'idée est d'ajouter plein de coupes pour réduire d'avantage l'espace de recherche (espace des solutions faisables), et puis l'obtention de la solution optimale sera plus rapide. Ces coupes peuvent être des propriétés de dominance ou d'inégalités validées qui sont ajoutées au modèle dans lequel on a intérêt à améliorer certaines caractéristiques spécifiques.

2.1.1 Élimination des sous-tours

C'est en prendre en compte les contraintes (1.10) dans la formulation alors l'algorithme nécessite un temps d'exécution très long dont la complexité est :

Soit U un sous ensemble de clients et F l'ensemble de véhicules. Pour U fixé on a

$$T(n) = (\text{card}(U))^2 \times \text{card}(F).$$

Et comme on a $\forall U \subset V^-$ avec $\text{card}(U) > 2$, alors

$$T(n) = C_n^2 \times 2^2 \times m + C_n^3 \times 3^2 \times m + \dots + C_n^n \times n^2 \times m = m \times \sum_{k=2}^n C_n^k k^2$$

Donc pour des raisons pratiques, ces contraintes sont éliminées de la formulation en vue de résoudre le problème.

Face à ce problème, Racha EL-HAJJ propose dans sa thèse ([18]) un autre type de contraintes plus fortes que les contraintes (1.10). Ces contraintes appelées GSEC (Generalized Subtour Elimination Constraints) introduits par Fischetti et al. [13], et ajoutées au modèle au besoin à la place de (1.10), assurent que chaque client servi par le véhicule r est connecté aux dépôts et ne forme pas des cycles.

Pour représenter ces contraintes, en utilisons les définitions suivantes. Soit S un ensemble de sommets, en introduisons l'ensemble $\delta(S)$ pour désigner les arcs connectant les sommets de S avec ceux en dehors et l'ensemble $\gamma(S)$ pour désigner les arcs connectant les sommets de S entre eux.

Propriété 2.1. *GSEC* :

$$\sum_{(u,v) \in \delta(S)} x_{uvr} \geq 2y_{ir}, \quad \forall S \subset V, \{d, a\} \subseteq S, \forall i \in V \setminus S, \forall r \in F \quad (2.1)$$

Propriété 2.2. *Contraintes équivalentes aux GSEC :*

$$\sum_{(u,v) \in \gamma(S)} x_{uvr} \leq \sum_{i \in S \setminus \{d,a\}} y_{ir} - y_{jr} + 1, \quad \forall S \subset V, \{d,a\} \subseteq S, \forall j \in V \setminus S, \forall r \in F \quad (2.2)$$

$$\sum_{(u,v) \in \gamma(U)} x_{uvr} \leq \sum_{i \in U} y_{ir} - y_{jr}, \quad \forall U \subset V^-, \forall j \in U, \forall r \in F \quad (2.3)$$

Inégalités (2.1) sont contraintes de connectivité en disant que chaque coupe séparant deux nœuds visités (i et j) doit être franchie au moins deux fois. Sont des versions plus fortes de (1.7) et (1.8)

Les contraintes (2.2) et (2.3) remplacée les contraintes (1.10), permet d'éliminé les sous-tours. Cette forme de contraintes à l'avantage d'avoir moins de coefficients non nuls, d'où il est plus approprié pour l'approche de cutting-plane.

Dans un première temps le modèle de base est construit en utilisant les contraintes (1.6), (1.9), (1.11) et la fonction objectif, après la procédure de résolution commence, une solution réalisable est générée. En suite l'algorithme de Tarjan [21] est appliqué sur cette solution pour vérifier si elle contient des sous-tours, si ce n'est pas le cas alors la solution est considérée comme optimale et l'algorithme est terminée. Dans le cas contraire les contraintes (2.1), (2.2) et (2.3) déduites à partir de la solution obtenue, et ajoutée au modèle linéaire pour éliminer les sous-tours.

Remarque 2.1. Même si en arrive à remplacer les contraintes (1.10) par les contraintes GSEC la complexité reste toujours élevée, à ce stade on peut penser à autres contraintes plus efficaces.

2.1.2 Propriétés de dominance

Les propriétés de dominance sont ajoutées au modèle, soit pour éliminer certaines solutions réalisables qui ne sont pas optimales, ou réduire la symétrie de l'espace de recherche par favoriser quelques structures dans les solutions optimales. D'une manière générale, ceci implique une accélération du processus de résolution. Ces coupes comprennent des inégalités d'élimination de symétrie, des bornes sur le profit et le nombre de clients qu'on peut servir, les coupes des clients obligatoires et enfin les coupes qui éliminent les clients et les arcs inaccessibles.

a) Élimination de la symétrie

Les tours des solutions optimales peuvent être triés selon des critères spécifiques, à savoir la somme des profits recueillis, le nombre des clients ou de la longueur du tour.

Alors dans notre étude on se focalise sur les profits des tours en ordre croissant en ajoutant la contrainte (2.4) au modèle.

Propriété 2.3. *Elimination de la symétrie en se basant sur les profits :*

$$\sum_{i \in V^-} y_{i(r+1)} P_i - \sum_{i \in V^-} y_{ir} P_i \leq 0, \quad \forall r \in F \setminus \{m\} \quad (2.4)$$

Sans cette contrainte, chaque solution réalisable aurait un profit différent par son tour, y aurait au moins $(m! - 1)$ solutions réalisables, mais si en ajoutant cette contrainte seulement une parmi les $m!$ solutions serait réalisable, par exemple, celui ayant les profits de ses tournées dans l'ordre décroissant, d'où l'espace de recherche est largement réduite.

Remarque 2.2. Même si l'ajout de cette contrainte réduit l'espace de recherche, elle augmente la complexité car pour trier les profits de tous les tournées il faut à chaque fois comparer le profit de la tournée précédent par celui de la tournée suivante de tell sort à trouver une solution dont les profits des tournées dans l'ordre décroissant.

b) Les clients et les arcs inaccessibles

Une façon simple pour réduire la taille du problème est de servir que les clients et les arcs accessibles.

- Un client est considéré inaccessible si en servant seulement que le client entre les deux dépôts, ou la longueur de la tournée résultante dépasse la limite L .
- Un arc est inaccessible lorsque la longueur de la tournée reliant directement les dépôts à cet arc dépasse L .

Tous les clients et les arcs inaccessibles sont éliminés au début du modèle en utilisant la propriété suivante.

Propriété 2.4. *Si i un client inaccessible (resp. (i, j) un arc inaccessible), alors :*

$$\sum_{r \in F} y_{ir} = 0 \quad (2.5)$$

$$\sum_{r \in F} x_{ijr} = 0 \quad (2.6)$$

c) Les bornes sur les bénéfices et le nombre des clients servis

Afin de réduire encore l'espace de recherche, on proposé une propriété de dominance efficace qui limite les caractéristiques de chaque tour ou sous-ensemble des tours.

L'idée est de résoudre les petites instances (plus facile à résoudre), puis extraire les informations nécessaires pour le processus de sélection.

Avant d'aller dans les détails de ces propriétés, on définit quelques notations. Soit X une instance du TOP avec m véhicules, nous utilisons X_I pour désigner la même instance dans laquelle le profit de chaque client est considéré égal à 1. Nous définissons également X^g comme étant l'instance X dans laquelle le nombre de véhicules est réduit à g ($g \leq m$). D'autre part, nous utilisons $LB(X)$ (resp. $UB(X)$) pour désigner une borne inférieure (resp. supérieure) du TOP sur l'instance X .

Nous avons alors les inégalités valides suivantes.

Propriété 2.5. *Bornes sur le profit :*

$$\sum_{r \in H} \sum_{i \in V^-} y_{ir} P_i \leq UB(X^{|H|}), \forall H \subset F \quad (2.7)$$

$$\sum_{r \in H} \sum_{i \in V^-} y_{ir} P_i + UB(X^{m-|H|}) \geq LB(X), \forall H \subseteq F \quad (2.8)$$

Inégalités (2.7) sont triviales puisque la somme des profits de toutes les tours $|H|$ ne peut pas dépasser le profit optimal de l'instance avec exactement $|H|$ véhicules ou au moins une borne supérieure de cette instance. Pour les inégalités (2.8) travaillent dans le sens inverse en appliquant une limite inférieure au profit de chaque tour et chaque sous-ensemble des tours.

Propriété 2.6. *Bornes sur le nombre de clients qu'on peut servir :*

$$\sum_{r \in H} \sum_{i \in V^-} y_{ir} \leq UB(X_I^{|H|}), \forall H \subset F \quad (2.9)$$

$$\sum_{i \in V^-} y_{ir} \geq LB(\bar{X}_I^1), \forall r \in F \quad (2.10)$$

De la même façon que (2.7), le nombre de clients par tour ou par sous-ensemble de tours sont limité en utilisant les inégalités (2.9). Pour l'inégalité (2.10) on appliquant une borne inférieure sur le nombre de client de chaque tournée.

Dans un premier temps, on ajoute les coupes des bornes sur les profits et le nombre de clients servis, en calcule le nombre maximale (resp. minimale) pour le profit et le nombre de client servi de chaque tournée et chaque sous ensemble de tournées, ces bornes seront ajoutée comme bornes sup (resp. inf) lors de la résolution du problème d'origine pour réduit la taille de l'espace de recherche et éliminer des solutions asymétriques par exemple l'une avec une tournée ayant de nombreux client et les autres tours presque vide.

Les valeurs de LB dans (2.8) peuvent être calculées à partir d'une heuristique [6]. Les valeurs de UB dans (2.7), (2.8) et (2.9) sont obtenues à partir de la résolution de $X^{|H|}$ et $X^{m-|H|}$ en utilisant par exemple la PLNE. De même que pour les bornes inférieures LB , nous cherchons seulement des bornes supérieures UB de qualité satisfaisant. Dans (2.10) $LB(\bar{X}_I^1)$ est une borne inférieure obtenue en procédant à la résolution du modèle linéaire en nombres entiers décrit ci-dessus où il s'agit de minimiser la fonction objectif en rajoutant les contraintes (2.7) et (2.8) avec $|H| = 1$.

Exemple 2.1. Considérons une instance X de TOP avec $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ l'ensemble des clients et $F = \{V_1, V_2, V_3, V_4\}$ l'ensemble de véhicule. Soit T_1, T_2, T_3 et T_4 les tournées respectivement de V_1, V_2, V_3 et V_4 .

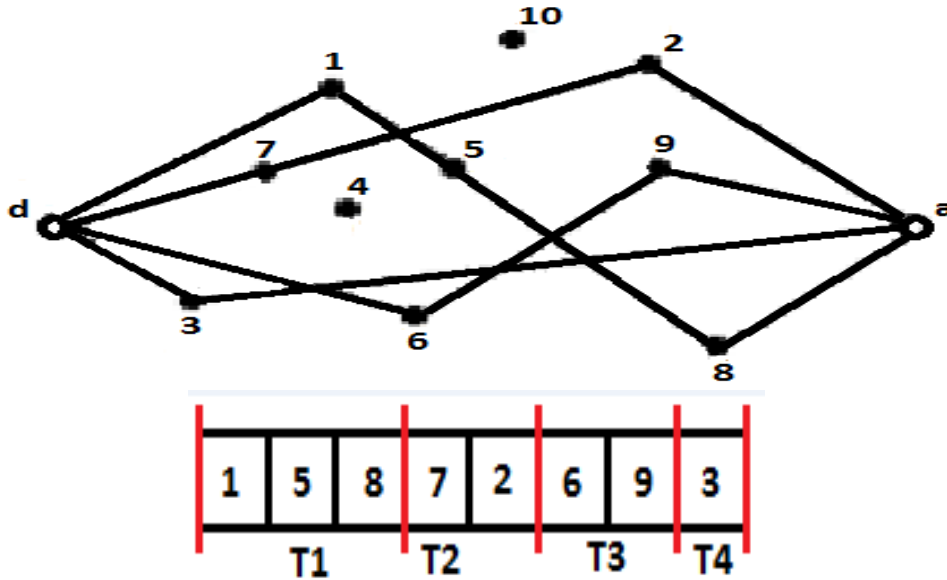


Figure 2.1 – Exemple d'une solution du TOP

Dans la premier itération de l'algorithme de cutting-plane en calcule $UB(X_I^1)$ la borne supérieur sur le nombre de client servi d'une tournée (supposons $UB(X_I^1) = 2$) et en ajoute la constraint (2.9) avec $|H| = 1$ au modèle c'est-à-dire que la somme de profit de chaque tournée ne peut pas dépasser la borne $UB(X_I^1)$ or dans la figure (2.1) on a le nombre de client de la tournée T_1 dépasse 2 alors la solution n'est pas réalisable. Supposons qu'on a trouvé une solution réalisable s'elle est optimal en arrêt et l'algorithme est terminé sinon en passe à la deuxième itération, en calcule $UB(X_I^2)$ la borne supérieur sur le nombre de client de tous les sous-ensembles avec deux véhicule et comme on a 4 véhicule alors on a $C_4^2 = 6$ combinaison possible des sous-ensembles avec deux véhicules, on répété la même procédure jusqu'à l'itération $m - 1 = 3$, c'est la solution n'a pas été atteint dans ce cas-là

en calcule $UB(X_I^3)$ et $LB(X_I^1)$ une borne inférieur sur le nombre de client servi d'une tournée, cette borne doivent être calculé après la résolution de tous les sous problème ($|H| \leq m - 1 = 3$) puisque la valeur $UB(X^{m-1}) = UB(X^3)$ et nécessaire pour calculer $LB(X_I^1)$. C'est l'ajout de tous ces coupe n'a pas cédé à une solution optimale en ajout d'autre types de contraintes supplémentaire.

- Remarque 2.3.** – A chaque fois qu'on ajoute une borne supérieur sur le nombre de client servi d'une tournée ou sous ensemble de tournée en ajout aussi une borne supérieur sur les profits d'une tournées ou sous ensemble de tournée.
- Les valeurs requises de LB et UB sont d'abord calculées pour l'instance avec $|H| = 1$, les valeurs obtenues sont utilisées dans les coupes supplémentaires pour résoudre les autres instances ($|H| \leq m$).
 - Les bornes supérieures et inférieures requises dans les inégalités (2.7), (2.8) et (2.9) sont associés à un tableau de taille m tandis que la borne inférieure de (2.10) est associé à une seule variable. Le tableau et la variable sont initialisés avec les valeurs par défaut, par exemple 0 pour les bornes inférieures, somme des profits ou le nombre de clients pour les bornes supérieures, et sont progressivement mis à jour avec les valeurs calculées obtenues en résolvant les instances associées.

d) Clients obligatoires

Pour chaque instance, certains clients peuvent être considérés comme des clients obligatoires et sont forcés d'être visités afin d'atteindre la solution optimale. Pour localiser ces clients, nous appliquons Définition (2.1) tout en utilisant la notation suivante.

Étant donnée une instance X du TOP avec i un client de X , nous définissons $X \setminus \{i\}$ pour désigner par exemple X avec i retiré.

Définition 2.1. Un client i de X est obligatoire si $UB(X \setminus \{i\}) < LB(X)$.

Où $LB(X)$ est calculée en utilisant une heuristique décrite dans Dang et al. [6]. Calcule de $UB(X \setminus i)$ est effectuée en utilisant le même algorithme que celui utilisé pour résoudre par exemple X , mais avec les contraintes supplémentaires $Y_{ir} = 0 \forall r \in F$.

Une fois, un client est détecté comme obligatoire, la propriété suivante est utilisée pour garantir le service de ce client par la flotte.

Propriété 2.7. Si i est un client obligatoire de X alors :

$$\sum_{r \in F} y_{ir} = 1 \quad (2.11)$$

2.1.3 Les inégalités valides basées sur les incompatibilités

Si deux clients donnés sont trop éloignés l'un de l'autre à cause de la limitation de temps de trajet, alors ces deux clients ne peuvent jamais être servis dans la même tournée. Cette observation nous conduit à la notion d'incompatibilité entre les clients. De plus, l'idée peut également être appliquée entre les autres composantes du problème, à savoir client-tour, client-arc ou arc-arc. Dans ce travail, nous nous concentrons sur deux incompatibilités particulières : entre les clients et entre les arcs.

a) Graphes d'incompatibilité

Les incompatibilités entre les clients et entre les arcs sont modélisées à l'aide de deux graphes. On note par $G_E^{Inc} = (E, E_E^{Inc})$ le graphe d'incompatibilité entre les arcs et par $G_{V^-}^{Inc} = (V^-, E_{V^-}^{Inc})$ le graphe d'incompatibilité entre les clients, une arête entre deux clients de ce graphe veut dire que ces deux clients sont incompatibles.

Soit S un sous-ensemble de sommets de V^- (ou d'arcs de E), nous désignons par $MinTraj(S)$ la longueur du plus court chemin de d à a contenant tous les sommets (ou les arcs) de S . Alors les graphes des incompatibilités sont initialisés comme suit.

Définition 2.2. Soit i et j deux clients de V^- telles que $MinTraj(\{i, j\}) > L$, Alors $[i, j] \in E_{V^-}^{Inc}$. De même, soit (i, j) et (u, v) de E satisfont $MinTraj((i, j), (u, v)) > L$, Alors $[(i, j), (u, v)] \in E_E^{Inc}$.

En plus de la définition (2.2), les graphes des incompatibilités sont étendus par les clients et les arcs plus incompatibles. Cela se fait en appliquant la définition (2.3) en utilisant les notations suivantes.

Étant donné deux clients i et j de X , nous utilisons $X \cup \{[i \sim j]\}$ pour désigner par exemple X où les deux clients sont forcés d'être servi par le même véhicule. De même, par exemple $X \cup \{[(i, j) \sim (u, v)]\}$ désigne l'instance modifiée de X dans laquelle des arcs (i, j) et (u, v) sont utilisés par le même véhicule.

Définition 2.3. Afin d'étendre le graphe d'incompatibilité entre les clients $G_{V^-}^{Inc} = (V^-, E_{V^-}^{Inc})$, un arc $[i, j]$ peut être ajouté à $E_{V^-}^{Inc}$ si $UB(X \cup \{[i \sim j]\}) < LB(X)$. De la même manière

pour le graphe d'incompatibilités entre les arcs $G_E^{Inc} = (E, E_E^{Inc})$, si $UB(X \cup \{[(i, j) \sim (u, v)]\}) < LB(X)$, Alors $[(i, j), (u, v)]$ est ajoutée au E_E^{Inc} . Pour calculer les valeurs requises pour UB . Cela peut être fait en utilisant PLNE sur les modèles suivants proposés dans la propriété 2.8. .

Propriété 2.8. *Considérons une instance X de TOP et deux clients i, j . Par conséquent, le modèle linéaire de $X \cup \{[i \sim j]\}$ est obtenue en ajoutant à celle de X les contraintes suivantes :*

$$\sum_{r \in F} y_{ir} = \sum_{r \in F} y_{jr} = 1 \quad (2.12)$$

$$y_{ir} = y_{jr}, \forall r \in F \quad (2.13)$$

De la même manière pour modéliser $X \cup \{[(i, j) \sim (u, v)]\}$. Les contraintes supplémentaires sont :

$$\sum_{r \in F} x_{ijr} = \sum_{r \in F} x_{uvr} = 1 \quad (2.14)$$

$$x_{ijr} = x_{uvr}, \forall r \in F \quad (2.15)$$

b) Coupes des cliques

Rappelons qu'une clique dans un graphe orienté est une sous-ensemble de sommets qui sont deux à deux adjacent. Ainsi, servir un client (ou utiliser un arc) d'une clique par un véhicule implique que tous les autres clients (ou arcs) de la clique d'être servi par le même véhicule. Par conséquent, chaque véhicule ne peut servir (ou utiliser) au plus un sommet (arête) de la clique. Sur la base de cette observation, la propriété suivante est titulaire pour G_E^{Inc} et $G_{V^-}^{Inc}$

Propriété 2.9. *Soit K (resp. Q) une clique dans $G_{V^-}^{Inc}$ (resp. G_E^{Inc}) :*

$$\sum_{i \in K} y_{ir} \leq 1, \forall r \in F \quad (2.16)$$

$$\sum_{[u,v] \in Q} x_{uvr} \leq 1, \forall r \in F \quad (2.17)$$

Lorsqu'on détermine une clique de ce graphe, alors nous avons en train de déterminer l'ensemble de clients qui sont tous incompatibles entre eux et ne peuvent pas être servis dans la même tournée, du coup chaque client de cette clique doit être servi dans une tournée séparé.

Remarque 2.4. Une clique est maximale si on ne peut pas trouver de clique qui la contient d'une façon stricte. Une clique maximum est une clique maximale dont laquelle son cardinal est le plus grand sur le graphe.

d) coupes d'ensembles indépendants

Les incompatibilités entre les clients et les arcs sont encore exploités à partir de la taille de l'ensemble indépendant maximal extrait de sous-graphe d'incompatibilité.

Les coupes d'ensembles indépendantes sont basées sur l'idée suivante. Considérons $G_{V^-}^{Inc}$ comme un exemple de graphe et soit S un sous-ensemble de V^- , on définit α_s être la taille d'un ensemble indépendant maximal de $G_{V^-}^{Inc}(S)$, le sous-graphe induite par S . Alors il est clair que au plus de α_s clients de S peuvent être servis dans la même tournée, par exemple $\sum_{i \in S} y_{ir} \leq \alpha_s$ est une coupe valable pour toute visite r , c'est-à-dire que chaque véhicule visité au plus α_s clients.

En outre, si l'on considère N_i l'ensemble des sommets voisins d'un sommet i dans $G_{V^-}^{Inc}(S)$, alors nous pouvons ajouter la coupe $\alpha_i y_{ir} + \sum_{j \in N_i} y_{jr} \leq \alpha_i$ au lieu de la précédente, où α_i est une courte notation pour α_{N_i} . Cette coupe particulière intègre la relation entre i et N_i , ainsi que les informations sur l'ensemble indépendant maximum de N_i . La même idée peut être généralisée à G_E^{Inc} , où l'on note N_{ij} l'ensemble des arcs voisins d'un arc (i, j) dans G_E^{Inc} , et les inégalités suivantes résument les coupes valides.

Propriété 2.10. *Coupes d'ensembles indépendants :*

$$\alpha_i y_{ir} + \sum_{j \in N_i} y_{jr} \leq \alpha_i, \forall i \in V^-, \forall r \in F \quad (2.18)$$

$$\alpha_{ij} x_{ijr} + \sum_{(u,v) \in N_{ij}} x_{uvr} \leq \alpha_{ij}, \forall (i, j) \in E, \forall r \in F \quad (2.19)$$

Remarque 2.5. – Un ensemble indépendant est un ensemble des sommets deux à deux non adjacents. Ensembles indépendants maximal et maximum sont définis de la même manière que pour les cliques.

- Trouver une clique ou indépendant maximum est NP-Difficile [14]. Cependant, les inégalités ci-dessus sont également valables pour α étant une borne supérieure de la taille d'un ensemble indépendant maximum. Une partition des sommets en cliques disjointes fournit une borne supérieure de la taille d'un ensemble indépendant maximum. Encore, l'algorithme efficace pour trouver une clique maximum peut être utilisé pour faire une telle partition, puis de calculer la borne supérieure.

Ces coupes de clique et ensemble indépendant sont alors générés à partir de $G_{V^-}^{Inc}$ et G_E^{Inc} et utilisés comme contraintes générales ajouté au modèle. Pour chaque sommet du graphe d'incompatibilité associé, nous déterminons une clique maximale contenant le sommet à l'aide de la méta-heuristique décrit dans Dan et Moukrim [4]. D'autre part,

en utilisant le même algorithme heuristique, une partition de chaque N_i (resp. N_{ij}) en cliques disjointes sont construits, par exemple, trouver une clique maximum, puis retirez ses sommets et continuer sur le graphe restant. Alors une limites supérieures pour α_i (resp α_{ij}) peuvent être calculées.

2.1.4 Algorithme de cutting-plane :

L'objectif de notre algorithme cutting-plan est d'améliorer le modèle linéaire de base en ajoutant quelques coupes supplémentaires à chaque fois , pour accélérer son processus de résolution afin d'atteindre la meilleure solution.

Les étapes de résolution appliquées dans l'algorithme sont les suivantes :

Dans un premier temps, le modèle de base est construit en utilisant les contraintes (1.6), (1.9) et (1.11) avec la fonction objectif (1.5) et des coupes supplémentaires initiales. Nous notons qu'avant de commencer la procédure de résolution, certains calculs initiaux sont effectués pour générer les coupes supplémentaires. Les composants non pertinents de X , à savoir les clients et les arcs inaccessibles, sont d'abord détectés et retirés du modèle. Ensuite, les graphes d'incompatibilité entre les clients et les arcs sont initialisés. Sur la base de ces coupes, les cliques et les ensembles indépendants sont calculés et ajoutés au modèle. Enfin, des coupes d'élimination de la symétrie sont ajoutées et la procédure de résolution commence. Une solution réalisable est générée à l'aide d'une heuristique de Dang et al. [5] et a fourni au solveur de PLNE comme solution de départ.

Algorithm 2.1: Cutting-Plane algorithm

Input: Instance X , additional cuts $D(X)$, timer TM
Output: Bound $UB(X)$, solution $SOL(X)$, indicator $Opt(X)$

```

1 begin
2   solver MIPS;
3   step  $\leftarrow$  1;
4    $Opt(X) \leftarrow$  false;
5    $UB(X) \leftarrow$  sum of profits of all customers of  $X$ ;
6    $SOL(X) \leftarrow$  a feasible solution of  $X$  [voir 5]; Référence
7    $LB(X) \leftarrow P(SOL(X))$ ;
8   MIPS.model( $X, D(X)$ ) (voir Section 1.4.2 et 2.1);
9   MIPS.initialize( $SOL(X)$ );
10  repeat
11     $\{UB, SOL, Opt\} \leftarrow$  MIPS.solve( $TM$ );
12    if ( $UB < UB(X)$ ) then  $UB(X) \leftarrow UB$ ;
13    if ( $Opt = true$ ) then
14      if ( $P(SOL) < UB(X)$ ) then  $UB(X) \leftarrow P(SOL)$ ;
15       $\{T_r\}_{r \in F} \leftarrow$  extract subtours from  $SOL$ ;
16       $\{S_r\}_{r \in F} \leftarrow$  extract tours from  $SOL$ ;
17      if ( $P(\bigcup_{r \in F} S_r) > LB(X)$ ) then
18         $SOL(X) \leftarrow \{S_r\}_{r \in F}$ ;
19         $LB(X) \leftarrow P(SOL(X))$ ;
20      if ( $|\bigcup_{r \in F} T_r| = 0$ ) or ( $LB(X) = UB(X)$ ) then
21         $Opt(X) \leftarrow true$ ;
22      else
23        MIPS.add(GSEC( $\{T_r\}_{r \in F}$ )) (voir Section 2.2.1);
24        MIPS.add(FindClique( $G_V^{Inc}[\bigcup_{r \in F} (T_r \cup S_r)]$ )) (voir Section 2.1.3);
25        MIPS.add(FindClique( $G_E^{Inc}[\bigcup_{r \in F} (T_r \cup S_r)]$ )) (voir Section 2.1.3);
26        if ( $OriginalProblem = true$ ) then
27           $D(X) \leftarrow$  CE algorithm( $X, step$ );
28          MIPS.add( $D(X)$ );
29          step  $\leftarrow$  step + 1;
30  until ( $Opt(X) = true$ ) or ( $TM.expired()$ );

```

Algorithme 2.1 : Algorithme de cutting-plane

Algorithme 2.1 prend en entrée une instance X et un ensemble de coupes supplémentaires $D(x)$, il exige également un solveur de programmation mixte en nombres entiers et une minuterie pour fonctionner. L'algorithme renvoie une borne supérieure $UB(X)$, une solution réalisable $SOL(X)$ et un indicateur booléen $Opt(X)$ indiquant l'optimalité de $SOL(X)$ avant l'expiration de la minuterie.

Nous supposons également que le solveur de programmation mixte en nombres entiers peut être adapté pour supporter les opérations suivantes :

model pour construire le modèle entier linéaire basé sur X et $D(X)$.

initialize fournir une solution de départ pour le solveur.

add pour compléter le modèle avec des réductions supplémentaires.

Solve pour essayer de résoudre le modèle jusqu'à l'expiration d'un temporisateur.

Remarque 2.6. La variable globale notée **OriginalProblem** est utilisée dans l'algorithme principal cutting-plan pour indiquer si le problème initial est résolu ou l'un des problèmes modifiés nécessaires pour générer les coupes supplémentaires. Égale 1 si le problème est résolu et 0 sinon.

Pour générer ces coupes, des calculs supplémentaires sont effectués au cours de la procédure de résolution en appelant l'algorithme CE décrit ci-dessous. Cet algorithme est appelé uniquement lorsque le problème d'origine est résolu. Dans l'itération suivante, de l'algorithme principal, le même processus de résolution est répété avec le modèle modifié. La boucle de résolution est répétée jusqu'à ce qu'une solution réalisable sans sous-tour est obtenue, qui sera la solution optimale, ou jusqu'à ce que le délai prédéfini est atteint. Dans ce dernier cas, une borne supérieure est renvoyée à l'instance au lieu de la solution optimale.

Algorithm 2.2: CE algorithm.

Input: Instance X , step s
Output: additional cuts $D(X)$

```

1 begin
2   if  $s \leq m - 1$  then
3     {UB, SOL, Opt}  $\leftarrow$  Cutting-Plane( $X^s$ ,  $D(X)$ ,  $TM_1$ );
4      $D(X) \leftarrow$  update from {UB, SOL, Opt};
5     {UB, SOL, Opt}  $\leftarrow$  Cutting-Plane( $X^s$ ,  $D(X)$ ,  $TM_1$ );
6      $D(X) \leftarrow$  update from {UB, SOL, Opt};
7     if ( $s = m - 1$ ) then
8       solver MIPS;
9       MIPS.model( $X^s$ ,  $D(X)$ );
10      {UB, SOL, Opt}  $\leftarrow$  MIPS.solve( $TM_1$ );
11       $D(X) \leftarrow$  update from {UB, SOL, Opt};
12   if  $s = m$  then
13     for  $i \in V^-$  do
14       {UB, SOL, Opt}  $\leftarrow$  Cutting-Plane( $X \setminus \{i\}$ ,  $D(X)$ ,  $TM_1$ );
15       if  $UB < LB(X)$  then
16         mandatory  $\leftarrow i$ ;
17      $D(X) \leftarrow$  update from mandatory;
18   if  $s = m + 1$  then
19     for  $(i, j) \in E$  do
20       {UB, SOL, Opt}  $\leftarrow$  Cutting-Plane( $X \cup \{i \sim j\}$ ,  $D(X)$ ,  $TM_1$ );
21       if  $UB < LB(X)$  then
22         update  $G_{V^-}^{Inc}$ ;
23       for  $(u, v) \in E$  do
24         {UB, SOL, Opt}  $\leftarrow$  Cutting-Plane( $X \cup \{(i, j) \sim (u, v)\}$ ,  $D(X)$ ,
25            $TM_1$ );
26         if  $UB < LB(X)$  then
27           update  $G_E^{Inc}$ ;
28    $D(X) \leftarrow$  findCliques( $G_{V^-}^{Inc}$ ,  $G_E^{Inc}$ );

```

Algorithme 2.2 : Algorithme CE

Les premiers types de coupes ajoutées au modèle supplémentaires sont les bornes sur les profits et le nombre de clients servis dans chaque ensemble de la tournée. Pour les premières $m - 1$ itérations de l'algorithme principal, l'algorithme CE est appelé à chaque itération générer toutes les coupes des bornes déduites du sous-problème correspondant. Pour chaque sous problème avec s véhicules, la borne supérieure pour les profits et le nombre de clients desservis par s véhicules sont calculés en utilisant le même algorithme de **cutting-plan**. Les contraintes correspondantes sont alors générées à partir de ces bornes supérieures et ajoutées à l'ensemble des coupes supplémentaires $D(X)$. Après avoir résolu le dernier sous-problème avec $m - 1$ véhicules, la borne inférieure pour le nombre de clients desservis par chaque véhicule est calculé depuis la résolution de ce problème de minimisation particulier on a besoin de quelques bornes inférieures déduites de $m - 1$

sous-problèmes précédents.

A la fin, pour certaines grandes instances, ni les bornes sur les profits et le nombre de clients servis, ni les clients obligatoires sont en mesure de les résoudre. A cette itération les graphes d'incompatibilité entre les clients et entre les arcs sont étendus pour générer les cliques et les ensembles indépendants, Ensuite les coupes correspondantes sont ajoutées à l'ensemble des coupes supplémentaires $D(X)$.

Remarque 2.7. Nous notons que pour générer les trois types de coupes supplémentaires, l'algorithme principal de cutting-plan est appelé avec une limite de temps configurée par minuterie TM_1 .

2.2 La résolution de TOP grâce à l'algorithme génétique

Dans cette section nous nous intéressons à la résolution du problème de tournées sélectives en utilisant l'algorithme génétique.

Les algorithmes génétiques découlent de la famille des algorithmes évolutionnistes. Leur objectif est d'obtenir une solution approchée d'un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte pour le résoudre en un temps raisonnable. Les algorithmes génétiques s'appuient sur des techniques dérivées de l'évolution naturelle : sélection, croisement, mutation, etc.

2.2.1 Codage utilisé :

Comme les chromosomes binaires sont inadaptes pour les problèmes de séquençement, nous avons adopté un codage réel où le chromosome est une séquence de nœuds (le dépôt exclu) sans paramètre délimiteur. Nous mentionnons la fin et les différentes caractéristiques (le temps de trajet parcourue, profit collecté) des tournées dans un tableau accompagnant le chromosome.

Sur la figure (2.2) le chromosome représente les trois tournées suivantes : la tournée (dépôt, 2, 20, 15, dépôt) de longueur L_1 et de profit P_1 , la tournée (dépôt, 3, 10, 6, 4, 19, 8, 30, dépôt) de longueur L_2 et de profit P_2 et la troisième tournée de longueur L_3 et de profit P_3 (dépôt, 25, 7, dépôt).

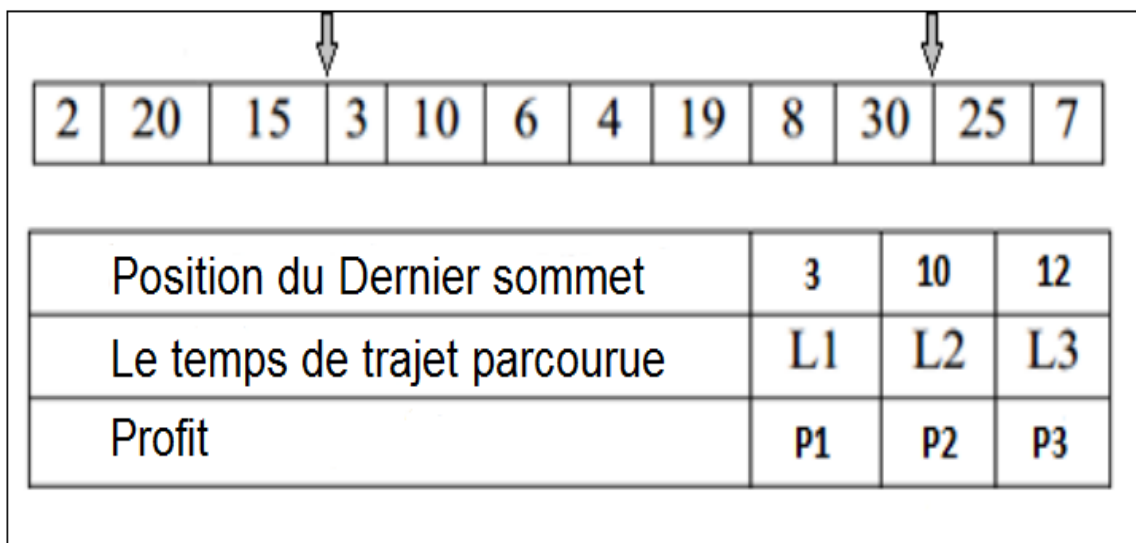


Figure 2.2 – Un chromosome représentant 3 tournées

Remarque 2.8. – Les flèches sur la figure (2.2) indiquent les limites entre les tournées.

Dans la suite ceci nous permet de donner uniquement les chromosomes sans les tableaux associés.

- Le caractère sélectif du problème rend variable la taille des chromosomes puisque le nombre total de clients visités n'est pas fixe.

2.2.2 Création de la population initiale :

Cette phase permet de générer une population d'individus qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global.

Pour construire la population initiale, nous avons développé une heuristique qui construit progressivement chaque chromosome de la manière suivante : à une itération donnée, le chromosome comporte plusieurs tournées dont le dernier tournées est en cours de construction, et son dernier gène est appelé sommet courant. L'algorithme doit choisir le prochain sommet à ajouter dans la tournée parmi les successeurs possibles du sommet courant dans le graphe. La sélection parmi les candidats se fait aléatoirement mais en favorisant les sommets ayant le plus grand profit. Elle repose sur un principe de roulette, où chaque sommet occupe un secteur proportionnel à son profit. Lorsqu'un sommet est sélectionné, il faut vérifier qu'il peut être visité en respectant les contraintes. Si ce n'est pas le cas, l'algorithme retire ce sommet de la roulette puis réitère la procédure pour tester un autre successeur. Si aucun successeur ne permet de vérifier les contraintes, la tournée s'achève

et l'algorithme tente d'en créer une nouvelle. Au fur et à mesure de la construction, la liste des sommets déjà insérés dans le chromosome est tenue à jour pour éviter de les tester une seconde fois. De plus, à chaque ajout d'un sommet dans une tournée, les caractéristiques de celle-ci (le temps de trajet parcourue, profit collecté) sont mises à jour.

Deux évènements peuvent achever la construction d'un chromosome. Soit le nombre de tournées déjà construites est égal au nombre de véhicules disponibles. Soit il ne reste aucun successeur qui vérifie les contraintes.

2.2.3 Sélection :

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais.

Afin de choisir les chromosomes (solutions) qui seront en mesure de contribuer à la création de la nouvelle population, nous avons adopté un mode de sélection qui consiste à attribuer à chaque individu une probabilité de sélection proportionnelle à son évaluation (fitness) (fonction objectif) et à la somme des évaluations des individus (Goldberg 1994). Si $f(i)$ est la force de l'individu i (fonction objectif), alors la fonction $S(i)$ (2.20) représente la probabilité de sélectionner ce chromosome i .

$$S(i) = \frac{f(i)}{\sum_{j=1}^N f(j)} \quad i \in [1, N]. \quad (2.20)$$

Avec N : la taille de la population.

De plus, pour éviter toute disparition du meilleur individu nous adoptons une stratégie élitiste en conservant les deux meilleurs individus de la population.

2.2.4 Croisement :

Le croisement a pour objectif d'enrichir la diversité de la population en manipulant la combinaison des chromosomes. Généralement, les croisements sont envisagés avec deux parents et génèrent deux enfants.

A partir de deux parents (solutions) choisis aléatoirement en fonction de leurs évaluations (fitness), nous essayons de générer un fils (une solution) qui soit réalisable. Ce type de croisement a été inspiré des travaux de Blaton et al. destinés au VRPTW (Blaton et al., 1993) et repose sur les étapes suivantes :

- a) considérer un gène candidat de chacun des deux parents. Un gène est candidat s'il n'a pas déjà été inséré ou rejeté pour cause de violation de contraintes et si tous les

gènes précédents dans le chromosome ont été insérés ou rejetés.

- b) insérer à la fin de la tournée fils en cours de construction le gène (client) ayant le plus grand profit et répondant aux contraintes (le temps de trajet). Si le gène initialement choisi ne satisfait pas les contraintes, tester l'insertion de l'autre gène.
- c) répéter cette procédure en considérant les nouveaux gènes candidats des deux parents.
- d) compléter la tournée par une procédure similaire à celle décrite en a) et b) avec comme critères de sélection la distance et la satisfaction des contraintes, et ce jusqu'à l'achèvement de la tournée en cours de construction. Si aucun des deux gènes ne convient, achever la tournée à cette étape.

Exemple 2.2.

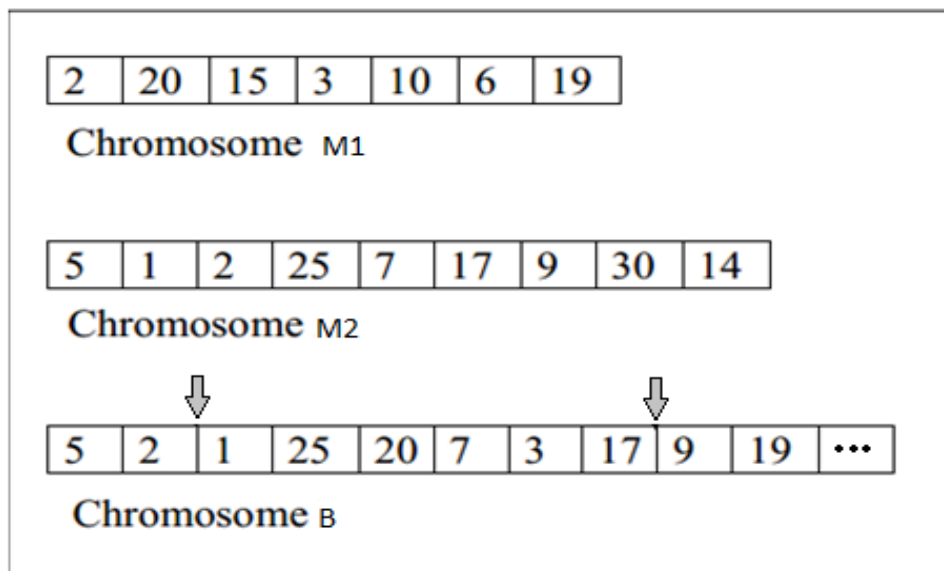


Figure 2.3 – Exemple de croisement entre deux parents.

Nous illustrons notre croisement par l'exemple sur la figure (2.3) Supposons que le chromosome B est le résultat du croisement des chromosomes $P1$ et $M2$, dont chacun est composé de 3 tournées, et que les tournées obtenues sont $(d, 5, 2, 1, a)$, $(d, 25, 20, 7, 3, 17, a)$ et $(d, 9, 19, ?, a)$ avec d et a représentant les dépôts. Ce chromosome est obtenu de la manière suivante : Nous comparons les profits de 5 et 2. Celui de 5 est supérieur alors nous l'insérons comme premier site de la tournée (tout sommet inséré sera marqué + pour éviter plusieurs visites d'un sommet). Nous comparons ensuite 2 du chromosome $M1$ et 1 du chromosome $M2$. Si 2 respecte les contraintes et comme son profit est supérieur à celui de 1, il est ajouté à la tournée fils juste après 5 dans le chromosome B .

Les deux sommets candidats suivants sont 20 et 1. Supposons cette fois que le profit de

1 est supérieur à celui de 20 mais que ni le sommet 20 du chromosome $M1$ ni 1 du chromosome $M2$ ne respectent les contraintes. Alors nous mettons fin à la tournée en cours de construction et nous recommençons la construction d'une nouvelle tournée en essayant d'insérer ces mêmes sommets 1 et 20. Le sommet 1 est favorisé grâce à son profit, et ainsi de suite jusqu'à atteindre la fin de l'un des deux chromosomes parents. Supposons que nous ayons inséré le sommet 9 du chromosome $M2$ et le dernier sommet 19 du chromosome $M1$. Il reste les sommets 30 et 14 du chromosome $M2$ à insérer. Dans ce cas, si l'ajout du sommet 30 à la tournée (*dépôt*, 9, 19, ...) ne viole pas les contraintes, nous l'effectuons et nous testons l'insertion du sommet 14. Sinon nous mettons fin à cette tournée et commençons la construction d'une nouvelle tournée s'il reste un autre véhicule libre.

2.2.5 Mutation :

L'opérateur de mutation consiste à remplacer, de façon aléatoire, un gène au sein d'un chromosome par un autre.

Au cours du processus d'évolution, la mutation effectue une exploration plus large de l'espace de recherche, afin d'éviter toute convergence prématurée ou disparition de la diversité en apportant de l'innovation dans la population. Nous avons adopté la méthode d'amélioration 2-opt (Laporte et al., 1998) en lui faisant subir une légère modification. La procédure originale est équivalente à une inversion d'une chaîne de sommets du chromosome dont les extrémités sont choisies aléatoirement. Dans notre cas, après l'inversion nous risquons de violer les contraintes. C'est pourquoi nous éliminons de la tournée les sommets ne respectant pas ces contraintes. Nous ne conservons le chromosome (voir figure (2.4)).

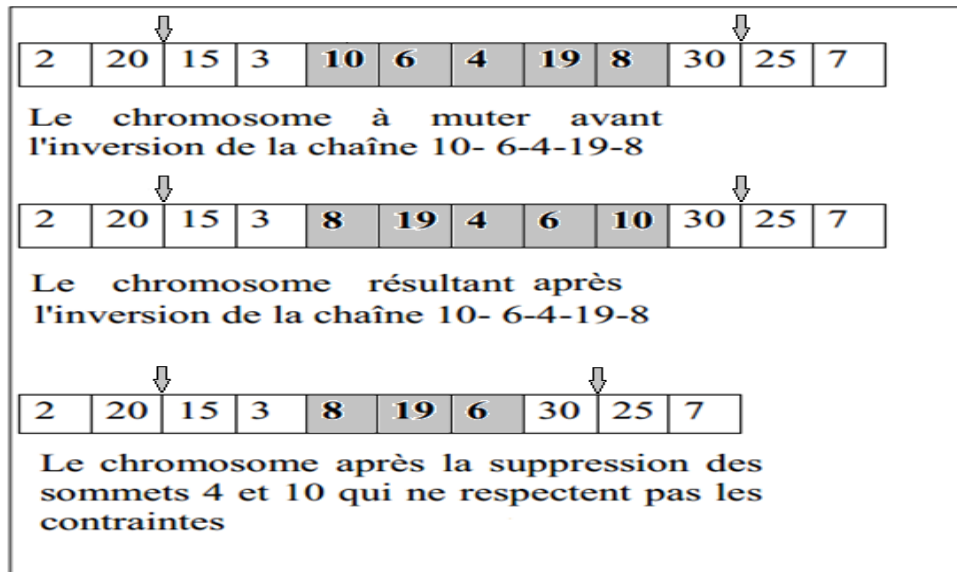


Figure 2.4 – Exemple de mutation.

2.2.6 Résultats Numériques :

L'algorithme proposé a été programmés en langage C. Et nous avons testé notre heuristique sur une machine Intel CORE i3, 2.2GHZ Ghz et de 4096 MB RAM. Nous avons utilisé les instances benchmark publiées par Chao et al. [9]. Ces instances sont de différentes tailles. Pour n et m fixés un sous-ensemble d'instances est obtenu en variant la longueur maximale L d'une tournée.

Approche par l'algorithme génétique

Les paramètres considérés sont les suivants :

1. Le nombre d'individus dans la population.
2. Le taux de mutation.
3. Le taux de croisement.

Après le test de notre approche, nous avons trouvé que les paramètres qui ont donné un bon résultat par rapport aux autres sont les suivants :

1. Le nombre d'individus 20
2. Le taux de mutation 0.02
3. Le taux de croisement 0.06

Les tableaux ci-dessous présentent un extrait des résultats obtenus après les tests de l'approche sur un ensemble d'instances classiques du problème, de taille $n = 102$ avec $m =$

2.2. La résolution de TOP grâce à l'algorithme génétique

4, 3, 2 et pour différentes longueurs maximales L . HMH et AG désignent respectivement les résultats obtenus par Hao et Miller-Hooks [25] et les résultats obtenus par notre approche

m	L	AG		HMH	
		P	Sec	P	Sec
4	65	463	11,85	643	68,6
	60	450	12,29	576	31,8
	55	425	12,46	503	44,9
	50	376	11,85	462	23,6
	45	297	11,68	359	20,6
	40	211	11,65	285	17,2
	35	196	11,48	217	10
	30	141	11,07	164	5,5
	25	129	11,35	123	3,3
	20	64	11,17	79	0,1
	10	30	3,8	30	0,1
3	80	588	12,4	681	69,5
	73,3	553	13,08	632	94,4
	66,7	472	12,01	563	107,7
	60	368	11,15	481	36
	53,3	290	12,58	416	34
	46,7	274	11,8	344	21,1
	40	128	10,73	247	23,7
	33,3	120	11,98	175	12
	26,7	109	11,48	117	3,7
	20	93	14	79	2,3
	13,3	46	3,6	46	0,1

2	110	685	13,44	702	120,1
	100	516	13,92	638	118,7
	90	502	13,61	578	84,4
	80	500	13,52	521	53
	70	395	14,05	459	74,6
	60	333	13,05	382	42,7
	50	198	11,77	290	37,7
	40	161	12,38	190	16,3
	30	148	12,24	110	8,6
	20	70	18,61	64	2,8
	10	30	3,71	30	0,1

Figure 2.5 – Extrait des résultats numériques.

Les significations des en-têtes des colonnes sont comme suit :

- **m** : nombre des tournées ;
- **n** : nombre de client ;
- **L** : longueur maximale d'une tournée ;
- **P** : profit total de la meilleure solution ;
- **Sec** : Temps d'exécution en secondes.

Comparaison des résultats obtenus par l'algorithme génétique avec celle proposée par Hao et Miller-Hooks obtenus par l'heuristique recherche TABOU.

Après l'analyse des résultats numériques obtenus par notre approche et mis à part les nouvelles solutions qu'elle a, nous avons remarqué qu'elle présente des inconvénients.

D'une part, les solutions obtenues par notre approche sont en moyenne moins bonnes que les solutions obtenues par l'autre méthode.

D'autre part, notre heuristique requiert des temps **Sec** élevés par rapport à l'autre méthode pour les cinq première valeur de L , elle donne des solutions en un temps d'exécution nettement inférieur à celui mis par HMH pour les autres valeur de L même si les profits de HMH et plus élevés que celle de notre approche.

2.3 Conclusion :

Dans ce chapitre, nous avons présenté deux méthodes pour la résolution de TOP, la première est une méthode exacte pour résoudre ce problème sur la base de l'algorithme

2.3. Conclusion :

de coupe-plan, et la deuxième est une méthode approchée basée sur l'algorithme génétique. En fin, nous pouvons dire que la dernière méthode de résolution du TOP que nous avons proposée est en moyenne meilleure que l'autre méthode de recherche TABOU, il suffit d'améliorer notre approche telles que la mutation, l'application d'autres métaheuristiques pour diminuer le temps d'exécution.



Deuxième partie

Dimension des ordres bipartis

Introduction

Les notions d'ordre, de classement, de rangement sont présentes dans de multiples activités ou situations humaines : hiérarchies administratives ou sociales, organigrammes, ordonnancements de tournois sportifs, ordres de pré-séance, de succession ou de préférences, notions d'ordre, ordres du jour, classements scolaires ou audiovisuels, ordre alphabétique, lexicographique, etc. On n'en finirait pas d'énumérer toutes les situations où interviennent des ordres.

Il n'est donc pas étonnant, compte tenu du développement de l'utilisation des mathématiques dans la modélisation de multiples phénomènes, de trouver de nombreux domaines où les mathématiques de l'ordre sont présentes. Toutefois, celles-ci sont relativement récentes. Certes, en mathématiques, la notion d'ordre des grandeurs est connue depuis longtemps et c'est au seizième siècle qu'apparaissent pour la première fois les symboles $>$ et $<$ pour désigner « plus grand que » et « plus petit que » 1. Mais la notion abstraite d'ordre défini comme un type particulier de relation transitive n'a été élaborée qu'entre les années 1880 et 1914 par des mathématiciens et/ou logiciens comme Peirce, Peano, Schröder, Cantor, Dedekind, Russel, Huntington, Scheffer et Hausdorff, dans le contexte de la formalisation de l'« algèbre de la logique » (i.e. de l'algèbre de Boole) et celui de la création de la théorie des ensembles (avec l'étude des « types d'ordre »).

Dans ce travail, nous ne nous intéressons qu'aux ordres finis, c'est-à-dire ayant un nombre fini d'éléments. Ces structures sont très étudiées et ces études peuvent prendre différentes formes comme par exemple une approche par la dimension. Le fait de se limiter aux ensembles ordonnés finis n'est pas simplement justifié par le souci de garder une taille raisonnable à cet ouvrage. Il le situe dans le champ des « mathématiques discrètes », dont l'importance n'est plus à démontrer de nos jours. Dans ce cadre, déjà très large, nous avons privilégié les notions et résultats qui nous paraissent essentiels, notamment du fait

de leurs usages dans de très nombreuses modélisations : extensions linéaires d'un ordre, fermetures, chaînes et antichaînes avec les théorèmes de Dilworth, dimensions d'un ordre, ordres d'intervalles, etc.

Le but du troisième chapitre est d'abord de donner et d'illustrer les notions fondamentales utilisées pour décrire, étudier, raisonner sur les ensembles ordonnés, notions qui seront utilisées et/ou développées dans le quatrième chapitres.

Dans le quatrième chapitre, au début on va donner les paramètres fondamentaux d'un ensemble ordonné. La première partie de ce chapitre sera donc consacrée aux théorèmes de Dilworth(1950) établit l'égalité pour un ensemble ordonné quelconque du nombre maximum d'éléments incomparables de cet ensemble ordonné et du nombre minimum de chaînes en lesquelles on peut partitionner l'ensemble de ses éléments. C'est un résultat central car, d'une part, il porte sur un ensemble ordonné arbitraire et permet d'y résoudre un problème rencontré dans des situations variées (par exemple, en recherche opérationnelle, informatique ou géométrie du plan) et, d'autre part, il est relié à beaucoup d'autres résultats célèbres de la combinatoire. Le corps de ce chapitre sera essentiellement constitué de l'étude de la dimension des ordres bipartis et on s'intéresse plus particulièrement aux au dimension 2. Et en fin, nous présentant quelques rappelles sur la complexité, après on montre qu'un ordre et de dimension k pour $k \geq 3$ est un problème NP-complet.

Ensembles ordonnés

Nous présentons dans ce chapitre quelques notions élémentaires sur les ensembles ordonnés, notion utilisée à de nombreuses reprises dans la suite de ce travail.

3.1 Relation d'ordre

Définition 3.1. (Relation binaires)

Une relation binaire d'un ensemble E vers un ensemble F est une partie R de $E \times F$. Si $(x, y) \in R$ on dit que x est en relation avec y et on note xRy .

Si $(x, y) \notin R$ on dit que x n'est pas en relation avec y .

Dans le cas particulier où $E = F$ on dit que R est une relation binaire définie sur E

Définition 3.2. On dit qu'une relation binaires R définie sur un ensemble E est :

1. réflexive si $(\forall x \in E) xRx$.
2. antiréflexive si $(\forall x \in E) \text{ on n'a pas } xRx$.
3. symétrique si $(\forall (x, y) \in E^2) xRy \Leftrightarrow yRx$
4. transitive si $(\forall (x, y, z) \in E^3) xRy \text{ et } yRz \Rightarrow xRz$
5. antisymétrique si $(\forall (x, y) \in E^2) xRy \text{ et } yRx \Leftrightarrow x = y$.

Définition 3.3. Une relation binaire d'un ensemble E est une relation de préordre si elle est réflexive et transitive. C'est une relation d'ordre si en plus elle est antisymétrique.

Une relation d'ordre est total si $\forall (x, y) \in E^2 xRy$ ou yRx . Elle est dit partielle dans le cas contraire.

Définition 3.4. Un couple $P = (E, \leq_P)$ formé d'un ensemble et d'une relation d'ordre noté \leq_P est appelé ensemble ordonné. Si \leq_P est totale l'ensemble est dit totalement ordonné.

Exemple 3.1. Soit $E = \{a, b, c, d, e\}$ et $P = (E, R)$ l'ensemble ordonné où R est l'ordre suivant sur E :

$$R = \{(a, b), (a, e), (c, b), (c, d), (c, e), (d, e), (a, a), (b, b), (c, c), (d, d), (e, e)\}.$$

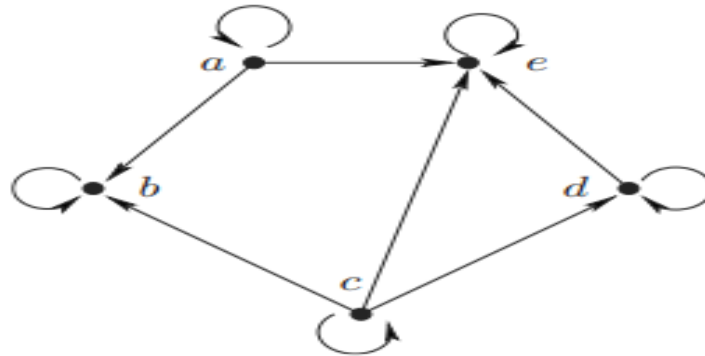


Figure 3.1 – Un ensemble ordonné P représenté par un réseau dans le plan.

Terminologie : Soit (E, \leq_P) un ensemble ordonné fini et $(x, y) \in E^2$

- On dit que x et y sont comparable dans P si $(x \leq_P y)$ ou $(y \leq_P x)$ et on note $x \sim_P y$. Ils sont dits incomparables dans le cas contraire et on note $x \parallel_P y$.
- Si $(x \leq_P y)$ on dit que x est inférieur à y ou y est supérieur à x
- Si $(x <_P y)$ on dit que x est strictement inférieur à y ou y est strictement supérieur à x .
- Pour tout x on note par $U_p(x)$ l'ensemble $\{y : y \in E \text{ et } x \leq_p y\}$ et $L_p(x)$ l'ensemble $\{y : y \in E \text{ et } y \leq_p x\}$.
- On dit que y est un successeur immédiat de x (ou y couvre x) si x est strictement inférieur à y et il n'existe pas de z tel que x soit strictement inférieur à z et z soit strictement inférieur à y . On dit alors que x est un prédécesseur immédiat de y ou (x est couvert par y).

L'ensemble ordonné P de l'exemple(3.1) a cinq couples de couverture :

$$a < b, a < e, c < b, c < d \text{ et } d < e.$$

Définition 3.5. Soit A une partie non vide de E , on note par $P \setminus A$ ou $P[A]$, l'ordre induit par P sur A ; il est défini par : $P[A] = (A, \leq_{P[A]})$ où pour tout x, y de A , $x \leq_{P[A]} y$

ssi $x \leq_p y$.

3.1.1 Minorants, majorants, minimaux et maximaux

Définition 3.6. Soit $P = (E, \leq_P)$ un ensemble ordonné et A une partie non vide de E .

- $a \in A$ est un élément maximal de A s'il n'existe pas de y dans A tel que $a \leq_P y$ et $y \neq a$.
- $b \in A$ est un élément minimal de A s'il n'existe pas de y dans A tel que $y \leq_P b$ et $y \neq b$.

L'ensemble ordonné P de l'exemple(3.1) a deux éléments minimaux, a et c , et deux éléments maximaux, b et e .

Définition 3.7. Soit $P = (E, \leq_P)$ un ensemble ordonné et A une partie non vide de E .

- On dit qu'un élément x de E est un majorant (resp. minorant) de A si ($y \in A$) $y \leq_P x$ (resp. $x \leq_P y$).
- Si l'ensemble des majorants (resp. minorants) de A est non vide, on dit que A est majorée (resp. minorée).

Une partie A de E est dite bornée si elle est à la fois majorée et minorée.

Définition 3.8. Soit (E, \leq_P) un ensemble ordonné et A une partie non vide de E .

- On dit que m est un minimum (ou plus petit élément) de A si m est un minorant de A et $m \in A$. on note $m = \min(A)$.
- On dit que M est un maximum (ou plus grand élément) de A si M est un majorant de A et $M \in A$. on note $M = \max(A)$.

Remarque 3.1. Si A admet un plus petit (resp. un plus grand) élément, alors cet élément est unique.

3.1.2 Diagramme de Hasse d'un ordre

Un ensemble ordonné $P = (E, \leq_P)$ où E est fini peut être représenté par un diagramme où la réflexivité et la transitivité sont implicites.

Le diagramme de hasse d'un ensemble ordonné s'obtient en procédant de proche en proche à partir des éléments minimaux et de leurs successeur immédiats.

Alors on peut le représenter graphiquement avec les contraintes suivantes :

- Les sommets sont les éléments de E ;
- Si $x \prec y$ on place x plus bas que y sur le diagramme ;
- Si $x \prec y$ et qu'il n'y a pas de $z \in E$ tel que $x \preceq z \preceq y$, on met une *arête* entre x et y .

Exemple 3.2. Le diagramme de Hasse pour l'ordre \subset sur $E = \{1, 2, 3\}$ et :

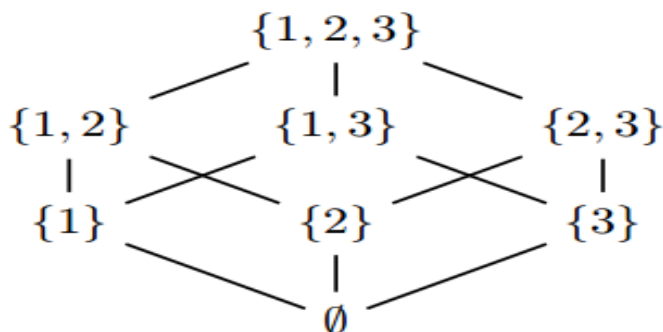


Figure 3.2 – Le diagramme de Hasse pour l'ordre \subset .

3.1.3 Isomorphisme et dualité

En mathématiques la notion d'isomorphisme entre deux structures est fondamentale. Elle permet de montrer que deux ensembles d'objets de nature totalement différente peuvent vérifier les mêmes propriétés. Dans le cas de structures d'ordre, s'y ajoute la notion d'anti-isomorphisme (ou de dualité) tout aussi importante.

Définition 3.9. Deux ensembles ordonnés $P = (X, \leq_P)$ et $Q = (Y, \leq_Q)$ sont dits isomorphes (ou de même type) s'il existe une bijection f de X sur Y vérifiant :

$$x \leq_P y \Leftrightarrow f(x) \leq_Q f(y)$$

Définition 3.10. Deux ensembles ordonnés $P = (X, \leq_P)$ et $Q = (Y, \leq_Q)$ sont dits anti-isomorphes (ou duaux) s'il existe une bijection f de X sur Y vérifiant, pour tous $x, y \in X$:

$$x \leq_P y \Leftrightarrow f(x) \geq_Q f(y)$$

Exemple 3.3.

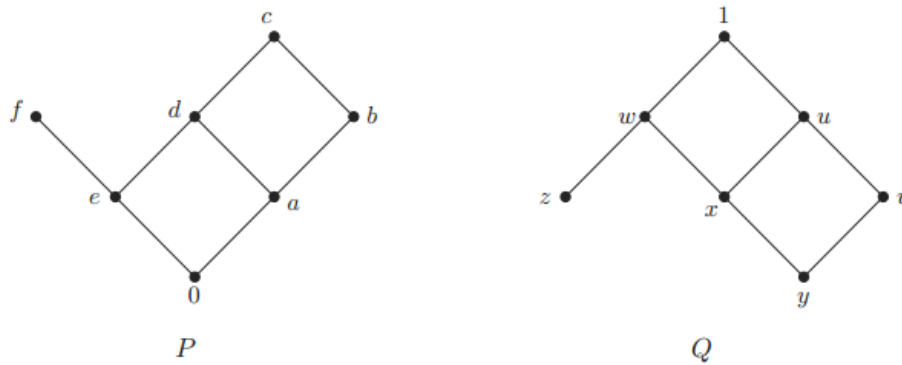


Figure 3.3 – Deux ensembles ordonnés P et Q duaux.

Un cas particulièrement intéressant d'anti-isomorphisme est obtenu en considérant l'ensemble ordonné $P_d = (X, \leq_{P_d})$ dual d'un ensemble ordonné $P = (X, \leq)$ et défini par :

$$x \leq_{P_d} y \Leftrightarrow y \leq x$$

Exemple 3.4.

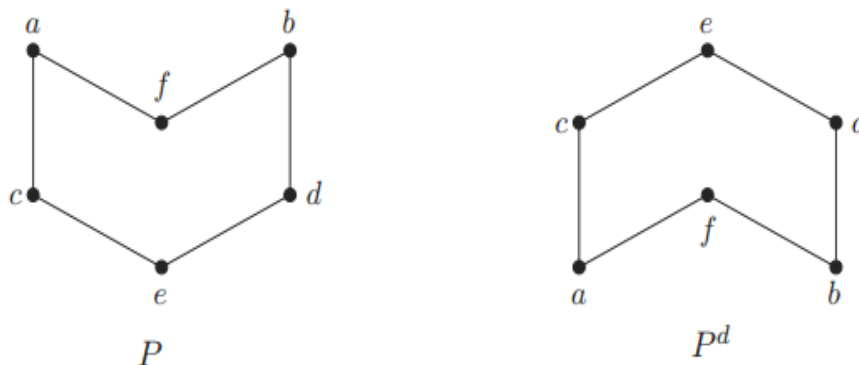


Figure 3.4 – Un ensemble ordonné P et son ensemble ordonné dual P^d .

Définition 3.11. (élément isolé d'un ordre)

Un élément d'un ordre P est dit isolé s'il est à la fois un maximum et un minimum de P .

3.1.4 Chaînes, Antichaînes et paramètres fondamentaux

Définition 3.12. Une partie A de X est une chaîne (resp. antichaîne) de P si $P[A]$ est une chaîne (resp. antichaîne).

Une chaîne (resp. antichaîne) est dit maximale s'il n'existe pas de chaîne (resp. antichaîne) qui la contienne de manière stricte.

Une chaîne C (resp. antichaîne) est dite maximum s'il n'existe pas de chaîne (resp. antichaîne) contenant plus d'éléments que C .

Dans l'exemple(3.1) $\{ae, cb, d\}$ constitue une partition en trois chaînes et $\{ac, d, be\}$ une partition en trois antichaîne de P .

Définition 3.13. (Hauteur)

Soit P un ordre fini, la hauteur de P , noté $h(P)$, est la longueur maximale d'une chaîne de P , soit $h(P) := \text{Max}_{\mathbb{N}}(\{|A| - 1 : A \subseteq E : A \text{ est une chaîne de } P\})$.

La hauteur d'un ordre est nulle si et seulement si cet ordre est une antichaîne.

Définition 3.14. (Largeur)

Soit P un ordre fini, la largeur de P , noté $\omega(P)$, est le cardinal maximal d'une antichaîne de P , soit $w(P) := \text{sup}\{|A| : A \text{ est une antichaîne de } P\}$.

Le seul moyen d'avoir un ordre de largeur nulle est que $E = \emptyset$

3.2 Conclusion

Dans ce chapitre, nous avons présenté les notions élémentaires sur les ensembles ordonnés, les relations binaires, et on a défini la relation d'ordre et quelques rappels de certaines définitions utilisées dans le chapitre suivant.

Dimension des ordres bipartis

Il apparaît nécessaire au début de ce chapitre de définir quatre paramètres fondamentaux attachés à un ensemble ordonné P . La première partie de ce chapitre sera donc consacrée au théorème de Dilworth, la deuxième partie traitera la notion usuelle de dimension des ensembles ordonnés et qui peut être ainsi définie à travers la notion d'extensions linéaires d'un ordre.

A la fin, ce chapitre traitera l'ordre de dimension deux et on s'intéresse plus particulièrement aux ordres bipartis de dimension deux, après on montre que le problème de décider si un ordre de dimension k pour $k \geq 3$ est un problème NP-complète.

4.1 Le théorème de Dilworth

Quatre paramètres fondamentaux attachés à un ensemble ordonné P est :

- La largeur $\omega(P)$ d'un ensemble ordonné n'est autre que la taille d'une antichaîne maximum.
- L'étendue $\kappa(P)$ de P est le nombre maximum d'éléments d'une chaîne de P ($h(P) = \kappa(P) - 1$ Ce paramètre est donc la longueur maximum d'une chaîne de P).
- $\gamma(P)$ le nombre minimum d'antichaînes dans une partition de P en antichaînes.
- $\theta(P)$ le nombre minimum de chaînes dans une partition de P en chaînes.

Remarque 4.1. Pour tout ensemble ordonné P , on a $\kappa(P) \leq \gamma(P)$ et $\omega(P) \leq \theta(P)$, car deux éléments de P ne peuvent appartenir simultanément à une même chaîne et à une même antichaîne. Le propos de cette section est d'établir que ces inégalités sont en fait des égalités (théorèmes (4.1) et (4.2)) et d'en tirer les premières conséquences. Nous commençons par montrer la première de ces égalités, ce qui est relativement simple.

Théorème 4.1. [18] *Pour tout ensemble ordonné P , on a l'égalité $\kappa(P) = \gamma(P)$, i.e si k est la taille d'une chaîne maximum de $P = (X, \leq_p)$ alors on peut partitionner X en k antichaînes.*

Preuve 4.1. Nous construisons une partition de P en un nombre h d'antichaînes A_1, A_2, \dots, A_h de la façon suivante : A_1 est l'ensemble des éléments minimaux de P et, pour tout $k > 1$, A_k est l'ensemble des éléments minimaux du sous ensemble ordonné $P_k = P \setminus (\cup_{1 \leq j < k} A_j)$. Chaque A_k est une antichaîne de P_k , donc de P et, pour tout $k > 1$, tout élément x_k de A_k couvre au moins un élément x_{k-1} de A_{k-1} (sinon on aurait trouvé $x_k \in A_{k-1}$). On peut donc construire une chaîne couvrante $C = x_1 \prec x_2 \prec \dots \prec x_h$ de P à h éléments. On a donc, d'une part, $\gamma(P) \leq h$, puisque P a été partitionné en h antichaînes et, d'autre part, $h \leq \kappa(P)$, puisque la chaîne C a h éléments. D'où l'inégalité $\gamma(P) \leq \kappa(P)$ et, comme l'inégalité inverse a été observée ci-dessus, on obtient l'égalité $\gamma(P) = \kappa(P)$.

Cette preuve est constructive, en ce sens qu'une partition de P en $\gamma(P)$ antichaînes est effectivement déterminée, ainsi qu'une chaîne de P ayant un nombre maximum $\kappa(P)$ d'éléments. Cette chaîne se présente comme une suite $x_1, x_2, \dots, x_{\kappa(P)}$, avec, pour $k = 1, \dots, \kappa(P)$, $x_k \in A_k$ et, pour $k = 2, \dots, \kappa(P)$, $x_{k-1} \prec x_k$.

En revanche, l'égalité $\omega(P) = \theta(P)$, qui fait l'objet du théorème de Dilworth, n'est pas obtenue ci-dessous de manière constructive.

Théorème 4.2. (R.P. Dilworth)

Soit $P = (X, \leq_p)$ un ensemble ordonné fini. Alors le nombre minimum de chaînes pour partitionner P est égale à sa largeur ($\omega(P) = \theta(p)$).

Preuve 4.2. La propriété est vraie pour $|P| \leq 1$.

supposons la vraie pour tous les ensemble ordonnés de taille k .

soit P tel que $|P| = k + 1$ et de largeur n . Distinguons deux cas :

1^{er} cas : P contient une antichaîne maximum A qui est distincte de $\max(P)$ et de $\min(P)$.

Soient $A' = \{y \in X ; y \leq_p x \text{ pour au moins un } x \text{ dans } A\}$ et $A'' = \{y \in X ; y \geq_p x \text{ pour au moins un } x \text{ dans } A\}$. On a $A' \neq X$ et $A'' \neq X$ puisque $A \neq \max(P)$ et $A \neq \min(P)$.

De plus, on a $A' \cap A'' = A$, $A' \cup A'' = X$ et $\omega(P) = \omega(p(A')) = \omega(p(A''))$.

Par hypothèse de récurrence, $P(A')$ et $p(A'')$ peuvent être partitionnés en $\omega(P)$ chaînes. soit C' et C'' de telles partition. chaque élément a de A est dans un seul C'_a de C' et dans un seul C''_a de C'' .

Clairement, pour chaque A la réunion $C_a = C'_a \cup C''_a$ est une chaîne, et la famille de ces chaînes forme une partition de P .

2^{ème} cas : tout antichaîne maximum est formée soit des éléments minimaux soit des éléments maximaux. Dans ce cas, choisissons une chaîne maximal C de P . Elle contient forcément un élément minimal et un élément maximal de P , donc $\omega(P(X \setminus C)) = \omega(P) - 1$, et par récurrence on peut partitionner $X \setminus C$ en $\omega(p) - 1$ chaînes de $P(X \setminus C)$. En ajoutant C , on obtient une décomposition de P en $\omega(P)$ chaînes.

Remarque 4.2. On peut noter que le théorème de Dilworth reste valable dans le cas d'un ensemble ordonné infini ayant une largeur finie. Cependant, le théorème tombe en défaut pour les ensembles ordonnés ayant une largeur infinie, même si toute antichaîne est de taille finie.

Par exemple, dans la figure ci-dessous, l'ordre a 4 comme largeur ($\{a, b, e, f\}$ est une antichaîne), et peut être couvert par 4 chaînes (e.g., $\{0, a, d, 1\}$, $\{b\}$, $\{e\}$, $\{c, f\}$).

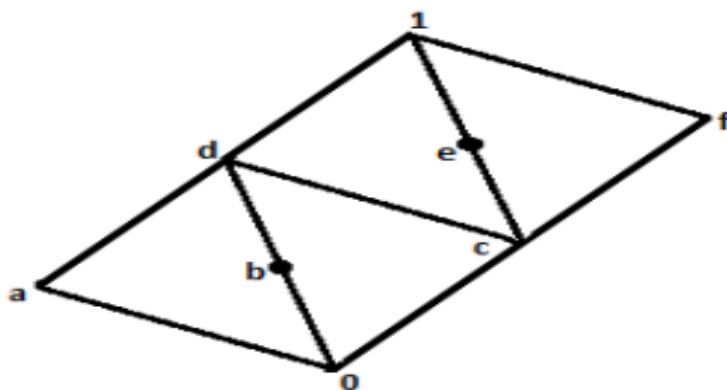


Figure 4.1 – Une illustration du théorème de Dilworth.

Soient $P = (X, \leq_p)$ un ensemble ordonné et I une partie non vide de X . On dit que I est un intervalle dans P si :

$$\forall z \in X, [(x, y \in I) \text{ et } (x \leq_p z \leq_p y)] \Rightarrow z \in I.$$

Par exemple, si $a \leq b$, l'ensemble $z \in X : a \leq_p z \leq_p b$ est un intervalle dans P que l'on note $[a, b]$.

On dira que y couvre x dans l'ensemble ordonné P , et on note $x < y$, chaque fois que $x \leq y$ et $[x, y] = \{x, y\}$.

Définition 4.1. (Graphe simple)

Un graphe simple est la donnée du couple $G = (S, A)$, S est appelé ensemble de sommets du graphe et A est appelé ensemble des arêtes du G .

Définition 4.2. (Graphe orienté)

Un graphe orienté est un couple $G = (X, E)$ où X est un ensemble et E est une relation binaire de base X . Les éléments de X sont appelés sommets de G et les éléments de E sont appelés arcs de G .

A chaque ensemble ordonné $P = (X, \leq_p)$, on associe quatres graphes :

- $G_1 = (X, U)$, graphe défini par : $\{x, y\} \in U$ si et seulement si $x \sim_p y$ et $x \neq y$, c'est le graphe de comparabilité ;
- $G_2 = (X, E)$, graphe orienté défini par $(x, y) \in E$ si et seulement si $x <_p y$ (on oublie la réflexivité). c'est le graphe de la relation ;
- $G_3 = (X, V)$, graphe orienté défini par : $(x, y) \in V$ si et seulement si y couvre x , c'est le graphe de Hasse ;
- $G_4 = (X, F)$, graphe simple défini par : $\{x, y\} \in F$ si et seulement si y couvre x ou x couvre y , c'est le graphe de couverture.

4.2 Extensions et générateurs

Soient $P = (E, \leq_p)$ et $Q = (E, \leq_Q)$ deux ensembles ordonnés, on définit l'intersection de P et Q par : $\forall x, y \in E^2, (x, y) \in P \cap Q \Rightarrow x \leq_p y$ et $x \leq_Q y$.

On dit que Q est une extension de P si $x \leq_p y$ implique $x \leq_Q y$ pour tous $x, y \in E$. Si de plus Q est un ordre total on dit que Q est une extension linéaire de P .

Théorème 4.3. (*Szpilrajn extension*)[23]

Soient $P = (E, \leq_p)$ un ensemble ordonné et $x, y \in E$ tels que $x \parallel_p y$. Alors, il existe une extension Q de P telle que $x \leq_Q y$.

Preuve 4.3. On définit une relation sur E , notée \leq_Q , définie par : $z \leq_Q z'$ si et seulement si $z \leq_p z'$ ou $z \leq_p x$ et $y \leq_p z'$. Il est facile de voir que $Q = (X, \leq_Q)$ est un ensemble ordonné contenant P .

Exemple 4.1.

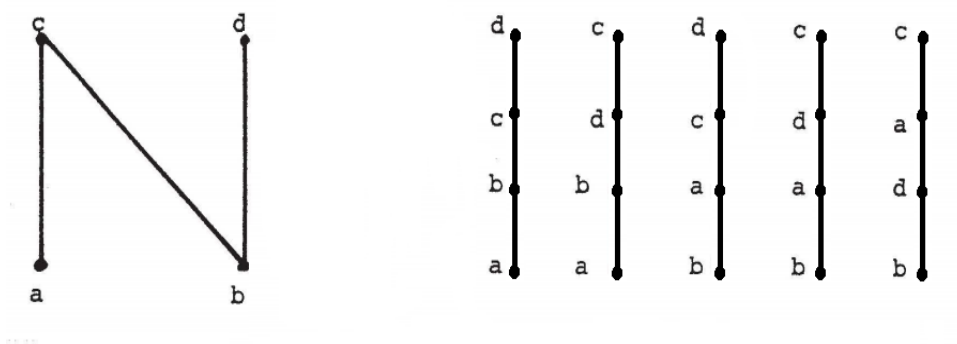


Figure 4.2 – Un ordre et ses cinq extension linéaires.

L'ensemble des extensions de P ordonné par inclusion admet P comme élément minimal. Les éléments maximaux pour cet ordre inclusion sont des ordres totaux d'après le lemme précédent. Cet ensemble, noté $L(p)$, est appelé ensembles des extensions lineaires de P .

D'après le lemme précédent, il est facile de voir $P = \cap L(p)$.

Le lemme ci-dessous s'obtient sans difficulté en invoquant l'antisymétrie et la transitivité des ordres.

Lemme 4.1. *Si P est un ordre sur un ensemble X , P n'a aucun circuit.*

4.3 Dimension d'un ordre

La notion de dimension se retrouve dans plusieurs objets mathématiques tels que les espaces euclidiens, les variétés, les espaces vectoriels et bien d'autres. Cette notion peut être adaptée avec succès aux ensembles ordonnés, la plus célèbre étant introduite en 1941 par B. Dushnik et E. W. Miller ([2]).

Définition 4.3. Soient $P = (X, \leq_P)$ Un ensemble ordonné et $R = \{L_1, \dots, L_n\}$ Une famille d'extension linéaires sur X , R est appelée générateur de P si $P = \cap_{i=1}^n L_i$.

Proposition 4.1. Soient $P = (X, \leq_P)$ un ensemble ordonné et $R = \{L_1, \dots, L_n\}$ une famille d'extensions linéaires de P . Les propriétés suivantes sont équivalentes :

1. R est un générateur de P
2. $P = \cap_{i=1}^n L_i$
3. Pour tous $x, y \in X$ tels que $x \parallel_P y$, il existe $1 \leq i \leq n$ tel que $x <_{L_i} y$
4. Pour tous $x, y \in X$ tels que $x \parallel_P y$, il existe $1 \leq i, j \leq n$ tels que $x <_{L_i} y$ et $y <_{L_j} x$.

Preuve :

(1) \Rightarrow (2) Par définition.

(2) \Rightarrow (3) $\forall x, y \in X$ tels que $x \parallel_p y$, supposon que $\forall i \quad y <_{L_i} x$ Or $P = \bigcap_{i=1}^n L_i \Rightarrow y <_P x$ absurde avec $x \parallel_p y$.

(3) \Rightarrow (4) $\forall x, y \in X$ tels que $x \parallel_p y$, d'après (3) il existe $1 \leq i \leq n$ tel que $x <_{L_i} y$ et il existe $1 \leq j \leq n$ tel que $y <_{L_j} x$ d'ou (4).

(4) \Rightarrow (1) Montrou que $P \subset \bigcap_{i=1}^n L_i$, soit $x, y \in X$ tels que $x <_P y$ alors d'après (4) $\forall 1 \leq i \leq n \quad x <_{L_i} y \Rightarrow P \subset \bigcap_{i=1}^n L_i$.

Montrou d'abord que $\bigcap_{i=1}^n L_i \subset P$, soit $x, y \in X$ tels que $x <_{L_i} y \forall 1 \leq i \leq n$ supposon que $x \parallel_p y$ alors d'après (4) il existe $1 \leq i \leq n$ tels que $y <_{L_i} x \Rightarrow x <_P x$ absurde.

Définition 4.4. La dimension d'un ensemble ordonné $P = (X, \leq_p)$, notée $\dim(P)$, est le plus petit entier n tel qu'il existe une famille $R = \{L_1, \dots, L_n\}$ d'extentions linéaires de P telle que $P = \bigcap R = \bigcap_{i=1}^n L_i$. Une telle famille est appelée réalisateur ou base.

- Remarque 4.3.**
- Un ensemble ordonné est de dimension 1 si et seulement si est un ordre total.
 - Un ensemble ordonné $P = (X, \leq_p)$ ayant au moins deux éléments qui est antichaîne est de dimension 2. En effet, $R = \{L, L^d\}$ est un générateur de P où L un ordre total quelconque sur X .
 - Tout ensemble ordonné P a même dimension que son dual. en effet, $\{L_1, \dots, L_n\}$ est générateur de P si et seulement si $\{L_1^d, \dots, L_n^d\}$ est un générateur de P^d .

Exemple 4.2.

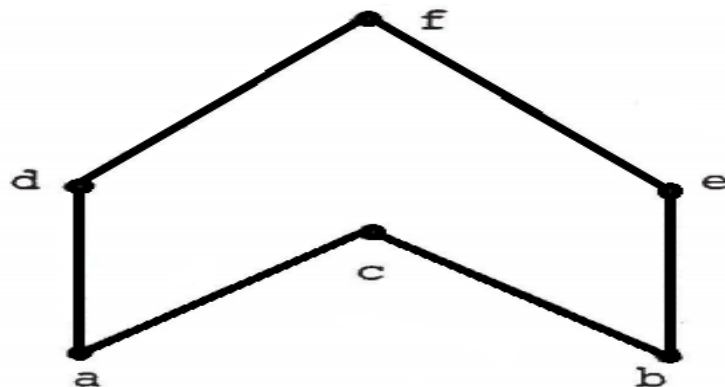


Figure 4.3 – Un ordre de dimension 3, une base étant $\{adbecf, adbcef, beacdf\}$.

Proposition 4.2. Soient $P = (X, \leq_p)$ un ensemble ordonné et Y une partie non vide de X . Alors, $\dim(P \setminus Y) \leq \dim(P)$.

Preuve : Si $\{L_1, \dots, L_n\}$ est générateur de P , alors $\{L_1 \setminus Y, \dots, L_n \setminus Y\}$ est générateur de $P \setminus Y$.

Proposition 4.3. Soient $P = (X, \leq_p)$ un ensemble ordonné et $x \in X$ Alors, $\dim(P) - 1 \leq \dim(P - x) \leq \dim(P)$

Preuve : Le seconde inégalité est évidente d'après la proposition précédents.

Pour la première, considérons $\{\tau_1, \dots, \tau_n\}$ un réalisateur de P -x.

Soient $U_p(x) = \{y : y \in X \text{ et } x <_P y\}$, $D_p(x) = \{y : y \in X \text{ et } y <_p x\}$ et $I_p = \{y : x \parallel_P y\}$.

On pose $L_0 = \tau_1(D_p(x) \cup I_p(x))x\tau_1(U_p(x))$, $L_1 = \tau_1(D_p(x))x\tau_1(U_p(x) \cup I_p(x))$.

Pour $i \in \{2, \dots, n\}$, on définit L_i à partir de τ_i en insérant x juste après le dernier élément de $D_p(x)$.

Clairement les L_i sont des extension linéaires de P .

Montrons que $P = \bigcap_{i=0}^n L_i$.

Si $x \parallel_p y$, alors $y <_{L_0} x$ et $x <_{L_1} y$.

Prenons $a, b \in X - \{x\}$ tels que $a \parallel_p b$.

Supposons que $a <_{\tau_i} b$ pour $i \in \{2, \dots, n\}$, on a donc $b <_{\tau_1} a$.

Montrons que $b <_{L_0} a$ ou $b <_{L_1} a$.

On a trois cas :

Si $a, b \in D_p(x)$, $b <_{\tau_1} a \Rightarrow b <_{L_0} a$ ou $b <_{L_1} a$.

Si $a, b \in I_p(x)$, $b <_{\tau_1} a \Rightarrow b <_{L_0} a$ ou $b <_{L_1} a$.

Si $a, b \in U_p(x)$, $b <_{\tau_1} a \Rightarrow b <_{L_0} a$ ou $b <_{L_1} a$.

Si $a, b \in D_p(x) \cup I_p(x)$, $b <_{\tau_1} a \Rightarrow b <_{L_0} a$ ou $b <_{L_1} a$.

Si $a, b \in U_p(x) \cup I_p(x)$, $b <_{\tau_1} a \Rightarrow b <_{L_0} a$ ou $b <_{L_1} a$.

donc dans tous les cas on a $b <_{L_0} a$ ou $b <_{L_1} a$.

Remarque 4.4. La dimension est donc monotone (i.e, la dimension d'un sous-ordre est plus petit que l'ordre de départ.

4.3.1 Ordres de dimension deux

Définition 4.5. Une extension linéaire L de P est non séparante si $\forall x, y, z \in X$, $x <_p y$, $x \parallel_p z, y \parallel_p z \Rightarrow z <_L x$ ou $y <_L z$

Théorème 4.4. Un ordre est de dimension inférieur à deux si et seulement si il possède une extension linéaire non séparante.

Preuve : Trivial si $\dim(P) = 1$. Supposon donc $\dim(P) = 2$.

Soient L_1 et L_2 deux extensions linéaires telles que $P = \bigcap_{i=1}^2 L_i$. Il suffit que L_1 est non séparante.

Supposons qu'il existe $x, y, z \in X$, telles que $x <_{L_1} z <_{L_1} y$ avec $x <_P y$, $x \parallel_P z$ et $y \parallel_P z$. $x \parallel_P z$ et $x <_{L_1} z$ donc $z <_{L_2} x$; $x <_P y$, donc $z <_{L_2} y$. Or $z <_{L_1} y$, donc $z <_P y$, contradiction.

Condition suffisante

Soit L_1 une extension linéaire non séparante de P , soit L' définie par : $x <_{L'} y$, ssi $x <_P y$ ou $x \parallel_P y$ et $y <_{L_1} x$.

L' est un graphe orienté complet antisymétrique. Pour montrer que L' est une extension linéaire, il suffit de vérifier que L' n'a pas de circuit.

Si L' a un circuit, alors elle a un circuit de longueur trois. En effet, si $\mu = (x_1, \dots, x_k, x_1)$ est un circuit avec $k \geq 4$, alors (x_1, x_2, x_3, x_1) est un circuit, soit $(x_1, x_3, \dots, x_k, x_1)$ est un circuit de plus court.

Soit (a, b, c, a) un circuit de L' . Ce ne peut être un circuit de P . Donc, il y a au plus deux arcs de P dans le circuit. S'il y'en a deux, par exemple (a, b) et (b, c) , alors par transitivité de P , $(a, c) \in P$ contradiction avec $(c, a) \in L'$. S'il en a un seul, par exemple $(a, b) \in P$, alors $(b, c), (c, a) \in L'$ contradiction avec le fait que L_1 est non séparante.

S'il n'en a aucun, (a, c, b, a) est un circuit de L' , ce qui contredit que L' est un ordre. L' est donc une extension linéaire de P .

Définition 4.6. – Soit $G = (X, E)$ un graphe simple. On appelle orientation transitive de G , une application ϕ qui à toute arête $\{a, b\}$ associe l'un des arcs (a, b) ou (b, a) de sorte que le graphe orienté obtenu soit le graphe d'une relation d'ordre partiel.

- On dira que G est un graphe de comparabilité s'il admet une orientation transitive.
- Le graphe d'incomparabilité de $P = (X, \leq_P)$ est $IP = (X, E)$ où $\{x, y\} \in E$ si et seulement si $x \parallel_P y$.

Théorème 4.5. *Un ordre est de dimension inférieure à deux si et seulement si IP est un graphe de comparabilité.*

Preuve :

Condition nécessaire

Si $\dim(P) = 1$, IP n'a pas d'arêtes, c'est un graphe de comparabilité.

Si $\dim(P) = 2$, on considère une extension linéaire L non séparante et on définit une

orientation F sur IP en posant $xy \in F$ si $x \parallel_P y$ et $x <_L y$.

Montrons que F est une orientation transitive.

supposons que $xy, yz \in F$.

Si $x \parallel_P z$, on a $x <_L y <_L z$, d'où $xz \in F$.

Si $x \perp_P z$, alors $x <_p z$ puisque $x <_L z$. Contradiction avec le fait que L est une extension linéaire non séparante.

Condition suffisante

Soit F une orientation de IP . On pose $L = P \cup F$.

Montrons que L est une extension linéaire de P .

L est un tournoi(entre deux sommets distincts il y a un seul arc). S'il y a un circuit, il y a un circuit de longueur 3. Ce circuit ne peut être ni à l'intérieur de P ni de F .

Si (a, b, c, a) a deux arcs dans P . Cela contredit la transitivité de P . de même, Si (a, b, c, a) a deux arcs dans F .

Montrons que L est non séparante.

Supposons par l'absurde qu'il existe $x, y, z \in X$, telles que $x <_L y <_L z$ avec $x <_P z$, $x \parallel_P z$ et $y \parallel_P z$. On a donc $xy, yz \in F$ et $x <_P z$, ce qui contredit le fait que F est transitif. D'où $\dim(p) \leq 2$.

Une caractérisation des ordres de dimension deux :

Théorème 4.6. *La dimension d'un ordre P est inférieure ou égale à 2 si et seulement si P est isomorphe à l'ordre d'inclusion sur une famille d'intervalles d'un ordre total.*

Preuve :

Condition nécessaire

supposons que $\dim(P) = 2$ alors $P = L_1 \cap L_2$

Donc $L_1 = x_1, \dots, x_n$ et $L_2 = y_1, \dots, y_n$

Soit $x, y \in X$ tels que $x <_P y$ alors

il existe i, j avec $i < j$ tels que $x_i = x$ et $y_j = y$ $x_i <_{L_1} y_j$

il existe l, l' avec $l < l'$ tels que $y_l = x$ et $y_{l'} = y$ $y_l <_{L_2} y_{l'}$

soit $L_1 L_2^d = x_1, \dots, x_n y_n, \dots, y_1$ alors $x = x_i$ et $x = y_{n-l+1}$ alors on associe à x l'intervalle $I_x = [i, 2n - l + 1]$ et de même on associe à y l'intervalle $I_y = [j, 2n - l' + 1]$

et par suit $I_y \subset I_x$ d'où P est isomorphe à l'ordre d'inclusion sur une famille d'intervalles d'un ordre total.

condition est suffisante

Nous avons besoin pour la suite de montrer que la condition est suffisante.

Soit un ensemble E et un ordre total O sur E . un intervalle $[i, t]$ de O est défini par son élément initial i et son élément terminal t , avec $i \in E$, $t \in E$, et $(i, t) \in O$. si $e \in E$, $e \in [i, t] \Leftrightarrow (i, e) \in O$ et $(e, t) \in O$.

Définissons deux ordres totaux O_1 et O_2 sur l'ensemble H des intervalles de O . Si $I = [i, t] \in H$ et $I' = [i', t'] \in H$, on a

- $t \neq t'$ et $(t, t') \in O \Rightarrow (I, I') \in O_1$.
- $t = t'$ et $(i', i) \in O \Rightarrow (I, I') \in O_1$.
- $i \neq i'$ et $(i', i) \in O \Rightarrow (I, I') \in O_2$.
- $i = i'$ et $(t, t') \in O \Rightarrow (I, I') \in O_2$.

Avec les ordres O_1 et O_2 ainsi construites, on a bien :

$$[i, t] \subset [i', t'] \Leftrightarrow ([i, t], [i', t']) \in O_1 \cap O_2$$

La restriction de l'ordre d'inclusion à une famille quelconque d'intervalles de O est donc de dimension au plus égale à 2.

4.4 Ensemble ordonné biparti :

Définition 4.7. Un ensemble ordonné P est biparti si l'étendue $\kappa(P)$ de P est égale à 2, c'est-à-dire si P est de hauteur 1.

L'exemple (4.4) donne deux exemples d'ensembles ordonnés bipartis :

- Le premier est l'ensemble ordonné biparti complet $K_{4,3}$, plus généralement, on note $K_{p,q}$ l'ensemble ordonné biparti où chacun des p éléments minimaux est inférieur à chacun des q éléments maximaux.
- Le second exemple est la couronne CR_4 , plus généralement, on appelle couronne CR_p l'ensemble ordonné formé de p éléments minimaux a_1, \dots, a_p et de p éléments maximaux b_1, \dots, b_p vérifiant $a_i < b_{i-1}, b_i$ pour tout $1 < i \leq p$ et $a_1 < b_p, b_1$.

Exemple 4.3.

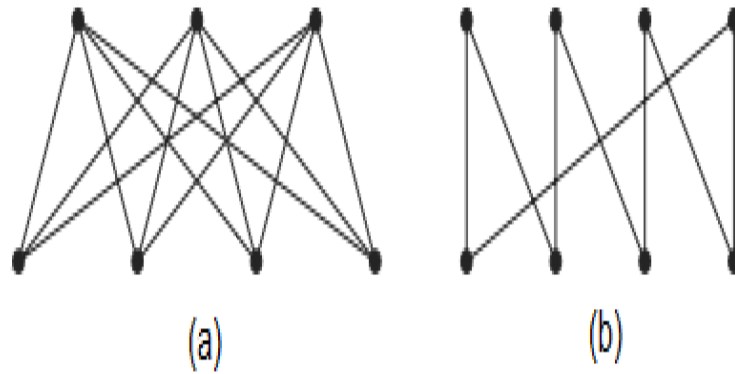


Figure 4.5 – (a) $K_{4,3}$ (b) CR_4

a) Les ordres bipartis de dimensions deux

Les ordres bipartis de dimensions 2 sont des ordres de dimensions deux de hauteur 1, à savoir avec seulement 2 niveaux. Ces ordres et leurs graphes de comparabilité ont été étudiés récemment par rapport à des problèmes qui sont NP-difficile sur les graphes bipartites arbitraires ([1], [12], [10]).

Théorème 4.7. Soit $P = (X, <)$ est un ordre de hauteur 1 avec niveau inférieur $V_1 = \text{Min}(P)$ et de niveau supérieur $V_2 = V - V_1$.

P est de dimension 2 si et seulement si il existe un ordre a_1, a_2, \dots, a_{n_1} de V_1 et b_1, b_2, \dots, b_{n_2} de V_2 de telle sorte que

$$\begin{cases} a_i <_P b_k, a_j <_P b_l \text{ avec } i < j \text{ et } k > l \\ \text{implique } a_i <_P b_l \text{ et } a_j <_P b_k, \end{cases} .$$

Exemple 4.4.

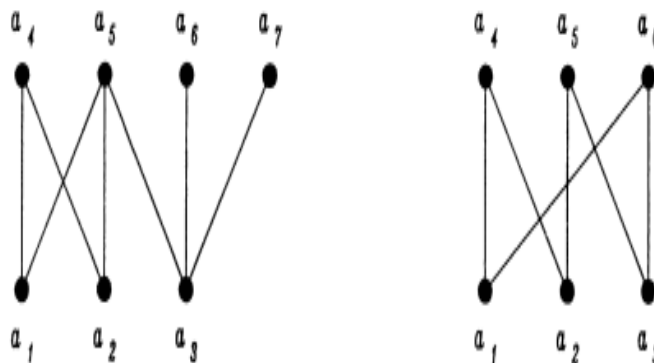


Figure 4.6 – un ordre biparti de dimension 2 et ordre non biparti de dimension 2

4.5 Les algorithmes :

La théorie de la complexité s'intéresse à l'étude formelle de la difficulté des problèmes en informatique. Elle est basée sur les travaux d'Edmonds [11] et de Cook [22]. L'objectif de calculer la complexité d'un algorithme, est d'obtenir un ordre de grandeur du nombre d'opérations élémentaires nécessaires pour que l'algorithme fournisse la solution du problème à l'utilisateur. Ceci permet de comparer la performance des algorithmes. La notion de rapidité d'un algorithme est très importante si l'on envisage des applications pratiques. Toutefois, jusqu'au milieu du XX^e siècle, la notion de rapidité d'un algorithme était vague et ne faisait pas l'objet de recherches pour elle-même.

Finalement le développement de la théorie de la complexité algorithmique depuis le milieu des années 1960 a placé la notion de rapidité d'un algorithme au cœur de questions très difficiles et profondes.

4.5.1 Problèmes, algorithmes, complexité

Un problème est défini par la donnée d'un objet mathématique représenté par un nombre fini de symboles (l'instance ou l'entrée) et d'une question dont la réponse dépend uniquement de l'instance. Si la réponse à la question ne peut être que "oui" ou "non", on parle de problème de décision.

D'une manière simple, un algorithme est une description de longueur finie, dans un langage de programmation, d'une suite finie d'opérations élémentaires (le calcul) ne dépendant que de l'instance et fournissant une réponse à la question (la sortie).

La complexité de l'algorithme est le nombre d'opérations élémentaires nécessaires à l'exécution de l'algorithme pour une instance de taille n (nombre de symboles nécessaire à sa représentation) dans le pire des cas.

Nos estimations de complexité seront toujours données à une constante multiplicative près à l'aide de la notation de Landau \mathcal{O} . Un algorithme est de complexité $\mathcal{O}(f(n))$ où f est une fonction sur N , s'il existe une constante c telle que le nombre d'opérations nécessaire pour exécuter l'algorithme sur une instance de taille n est inférieur à $c \times f(n)$.

Un algorithme est dit (en temps) polynomial si sa fonction de complexité est en $\mathcal{O}(n^k)$ avec $k \geq 1$ fixé. Dans ce cas, l'algorithme est considéré comme efficace.

4.5.2 Les problèmes P, NP, et NP-complets :

- La classe P ou classe Polynomiale : Cette classe regroupe les problèmes qui peuvent être résolus en utilisant un algorithme à temps polynomial.
- La classe NP ou classe Non-déterministe Polynomiale est une classe des problèmes qui sont « vérifiables » en temps polynomial. Cela signifie : si l'on nous donne, d'une façon ou d'une autre, une solution « certifiée », nous pouvons vérifier que cette solution est correcte en un temps qui est polynomial par rapport à la taille de l'entrée.

Définition 4.8. Un problème A est réductible à un problème B s'il existe un algorithme résolvant A qui utilise un algorithme résolvant B .

Définition 4.9. Si l'algorithme résolvant A est polynomial, considérant les appels à l'algorithme résolvant B comme de complexité constante, la réduction est dite polynomiale. On dit que A est polynomialement réductible à B et l'on note $A \leq B$.

Remarque 4.5. La relation \leq est réflexive et transitive, mais pas antisymétrique.

Un problème de la classe NP est dit NP-complet si tous les problèmes de la classe NP peuvent être réduits en temps polynomial à ce problème. Autrement dit, un problème A sera dit NP-Complet s'il respecte les deux propriétés suivantes :

- $A \in NP$
- A est NP-difficile c'est-à-dire : que $B \leq_P A$ pour tout problème $B \in NP$.

La résolution d'un problème NP peut nécessiter l'examen d'un grand nombre (éventuellement exponentiel) de cas mais l'examen de chaque cas doit se faire en temps polynomial. On déduit que $P \subset NP$. La question est de savoir si $NP \subset P$. La différence entre P et NP n'a pas été prouvée, mais la conjecture est considérée comme hautement probable. Ces problèmes constituent le noyau dur des problèmes NP, si bien que si l'on trouvait un algorithme polynomial permettant de résoudre un seul problème NP-Complet, on pourrait en déduire un autre pour tout problème NP .

La réduction polynomiale est la technique de base pour prouver que des problèmes sont NP-complet.

Définition 4.10. Un problème X se réduit en temps polynomial à un problème Y si et seulement si des instances du problème X peuvent être résolues avec un nombre polynomial de calculs élémentaires qui résolvent le problème Y , on note $X \leq_P Y$.

Théorème 4.8. [15]

Déterminer si un ordre est de dimension k , pour $k \geq 3$, est NP-complet.

Avant de décrire la réduction, nous obtenons un aperçu du problème. Soit $P = (N, \leq_P)$ un ensemble ordonné et supposons que N est divisée en deux ensembles S, S' pour qu'il n'y ait pas d'arc de P dirigé à partir d'un noeud de S' à un noeud de S . Soit $B(P)$ être le graphe biparti avec N ensemble de noeuds et un ensemble d'arêtes $\{\{x, y\}/x \in S, y \in S', x \text{ et } y \text{ incomparable}\}$. Considérons maintenant une extension linéaire L de P , et soit \bar{L} le graphe biparti avec N l'ensemble des noeuds et un ensemble des arêtes $\{\{x, y\}/x \in S, y \in S', (x, y) \in L\}$; i.e., L contient les arcs (sans la direction) qui sont dirigés à partir des noeuds de S' aux noeuds de S . Puisque P n'a pas de tels arcs et L est une extension linéaire de P , \bar{L} doit être un sous-graphe de $B(P)$.

Nous disons que deux arêtes $\{x, y\}, \{z, w\}$, d'un graphe sont indépendants si les noeuds x, y, z, w sont distincts et le graphe induit par eux se compose exactement de ces deux arêtes. Supposons que \bar{L} avait deux arêtes indépendants $\{x, y\}, \{z, w\}$ avec $x, z \in S$ et $y, w \in S'$. À partir de la définition de \bar{L} alors, L contiendrait $(y, x), (w, z), (z, y)$ et (x, w) ; i.e., L serait un cycle $y \rightarrow x \rightarrow w \rightarrow z \rightarrow y$ contredisant le fait que L est un ordre linéaire. Ainsi, \bar{L} n'a pas de paire des arêtes indépendants. Nous appelons un graphe biparti avec cette propriété, graphe de la chaîne. Il se caractérise également par la propriété que les sommets voisins Γ_x (ensembles de noeuds adjacents à x) de noeuds x en S sont totalement ordonnés par ensemble inclusion; i.e. pour tout x, y dans S , soit $\Gamma_x \subseteq \Gamma_y$ ou $\Gamma_y \subseteq \Gamma_x$. Supposons maintenant que P est de dimension d , et soient L_1, \dots, L_d être l'extension linéaire de P avec intersection est P . Soit $\bar{L}_1, \dots, \bar{L}_d$ le graphe biparti que nous avons défini ci-dessus. Chaque \bar{L}_i est un sous-graphe de chaîne $B(P)$. Etant donné que l'intersection de L'_i est de P , pour chaque arête $\{x, y\}$ de $B(P)$ avec $x \in S, y \in S'$, l'arc (y, x) doit apparaître dans au moins une des L'_i ; ainsi $\{x, y\}$ est couvert par (apparaît dans) au moins l'un des \bar{L}_i . Pour un graphe biparti G , soit $ch(G)$ le nombre minimum de sous-graphes de chaîne de G qui couvre toute les arêtes de G . Nous avons montré :

Lemme 4.2. $dim(P) \geq ch(B(P))$.

Ainsi, par exemple, si P est la couronne sur des noeuds $\{v_1, \dots, v_k, v'_1, \dots, v'_k\}$ avec des arcs $v_i \rightarrow v'_j$ pour $i \neq j$, et nous prenons $S = \{v_1, \dots, v_k\}, S' = \{v'_1, \dots, v'_k\}$, alors $B(P)$ est constitué des k paires d'arêtes indépendantes $\{v_1, v'_1\}, \dots, \{v_k, v'_k\}$ et $dim(P) \geq ch(B(P)) = k$.

Par réduction de 3-Coloration, qui est NP-complet Ce problème revient à décider s'il existe un coloriage du graphe $G = (S, E)$ utilisant au plus 3 couleurs. Dans G , nous allons construire un ordre partiel P de telle sorte que G peut être coloré avec 3 couleurs si et seulement si $dim(P) \leq 3$.

Soit $V = \{u_1, \dots, u_n\}$ être les nœuds de G , et $E = \{e_1, \dots, e_m\}$ ses arêtes. P est un ordre partiel sur l'union N de deux ensembles disjoints S et S' . S Contient deux nœuds u_{ia}, u_{ib} pour chaque nœud u_i de V , et deux noeuds u_{ik}, u_{jk} pour chaque arête $e_k = \{u_i, u_j\}$ dans E . S' contient les versions apprêtées des nœuds de S ; ainsi, au total N a $4n + 4m$ éléments. L'ordre partiel P est défini comme suit.

$$P = \{(u_{ia}, u'_{it}) / 1 \leq i \leq n, t \neq a\} \cup \{(u_{ib}, u'_{it}) / 1 \leq i \leq n, t \neq b\} \cup \{(u_{ik}, u'_{jl}) / 1 \leq i, j \leq n, 1 \leq k \leq m, l \geq k \text{ ou } l = a \text{ ou } b\} \cup \{(u_{ik}, u'_{jk}) / 1 \leq k \leq m, e_k = \{u_i, u_j\}\} \cup \{(u_{ik}, u'_{jl}) / 1 \leq i, j \leq n, 1 \leq k < l \leq m\}.$$

Notez que tous les arêtes entre S et S' sont dirigés de S à S' . Soit $B(P)$ le graphe biparti défini au-dessus. Soit Q les nœuds u_{ik} en S qui correspondent aux arêtes de G (ie, avec $1 \leq k \leq m$), R le reste des nœuds S (i.e. avec $k = a, b$), et que Q', R' être les sous-ensembles analogues de S' . En termes simples, $B(P)$ a la structure suivante : dans un nœud Q est relié à sa version apprêté et à tous les noeuds de Q' ayant un second indice strictement inférieur ; un nœud R est connecté à sa version apprêté et tous les nœuds de S' avec un premier indice différent. Figure 4.7 nous donnons un exemple de la construction ; une partie seulement de l'ordre partiel est représenté pour plus de clarté.

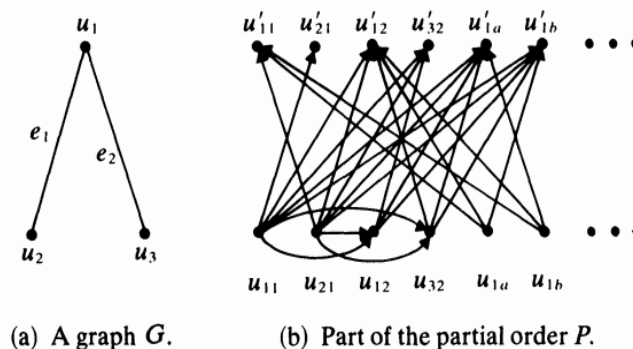


Figure 4.7.

Lemme 4.3. Si $ch(B(P)) \leq 3$ alors G peut être coloré avec au plus 3 couleurs.

Preuve. Supposons que $ch(B(P)) \leq 3$, et que B_1, B_2, B_3 soit trois sous-graphes de chaîne de $B(P)$ que couvrir. Considérons le graphe H_i de $B(P)$ induite par tous les nœuds u_{it} , avec i le premier indice. Il comporte trois composantes connexes : l'arête $\{u_{ia}, u'_{ia}\}$, l'arête $\{u_{ib}, u'_{ib}\}$ et le sous-graphe induit par le u_{ik}, u'_{ik} avec $1 \leq k \leq m$. Depuis un graphe de chaîne ne peut pas contenir deux arêtes indépendants, aucun des B'_j ne peut contenir deux arêtes de différentes composantes de H_i . Depuis H_i a trois composantes, tous les les arêtes du troisième composant sont dans le même B_j ; nœud u_i couleur avec l'index de cette B_j .

Nous devons montrer maintenant que ce soit une coloration juridique de G . Supposons qu'il existe deux noeuds adjacents u_i, u_j avec la même couleur, par exemple la couleur 1. soit $e_k = \{u_i, u_j\}$. De la définition de la coloration que nous avons $\{u_{ik}, u'_{ik}\}, \{u_{jk}, u'_{jk}\} \in B_1$. Mais ces deux arêtes sont indépendants dans $B(P)$, donc aussi dans B_1 , ce qui contredit l'hypothèse que B_1 est un graphe de chaîne.

Lemme 4.4. *Si G peut être coloré avec au plus 3 couleurs alors $d(P) \leq 3$.*

Preuve.[15]

Preuve. immédiatement de lemmes 4.3, 4.4, 4.5 on conclure que le problème est NP-complet.

4.6 Conclusion :

Dans ce chapitre, dans la première section nous avons présenté deux théorèmes de dilworth, dans la deuxième section on a défini la notion usuelle de dimension est appliquée aux ensembles ordonnés. Et en fin nous nous sommes surtout concentrés sur la dimension des ordres bipartis et nous avons montré qu'un un ordre est de dimension k , pour $k \geq 3$, est NP-complet à l'aide de la réduction polynomiale.

Conclusion et générale

Le travail que nous avons présenté concerne la résolution d'un problème d'optimisation rencontré dans le domaine du transport. Ce problème est appelé Problème de Tournées de Véhicules Sélectives ou le plus souvent problème de course d'orientation par équipe. Le TOP consiste à déterminer les tournées de m véhicules partant d'un dépôt et s'achevant à un autre. Chaque tournée doit desservir un sous-ensemble de clients dont chacun est caractérisé par une priorité ou un profit. Le problème consiste à maximiser le profit récolté par les m tournées sachant que chacune d'elles est limitée par une distance maximale. Afin de résoudre le TOP, nous avons conçu une approche par l'algorithme génétique qui sont vraiment robuste, il faut juste le guider pour qu'il puisse nous donner de bons résultats. Le développement de ce travail portera sur l'amélioration de nos procédures, telles que la mutation, l'application d'autres métaheuristiques, et l'extension de notre AG à d'autres variantes du TOP en ajoutant de nouvelles contraintes, la contrainte de capacité des véhicules. Ce problème a été traité récemment et pour la première fois par Archetti et al. (2007a) et nommé Capacitated Team Orienteering Problem CTOP. Nous pouvons encore ajouter la contrainte de temps de service ou la contrainte de fenêtre de temps.

Dans ce travail nous avons aussi utilisé les ensembles ordonnés pour décider si un ordre bipartite est de dimension k pour $k \geq 3$ est NP-complet à l'aide de la réduction polynomiale qui est la technique de base pour prouver que des problèmes sont NP-complets.

Bibliographie

- [1] A. Brandstandt and D. Kratsch, "On the restriction of some NP-complete graph problems to permutation graphs", 1984.
- [2] B. Dushnik, E.W. Miller, "Partially ordered sets", American J. Math. 63, 600-641, 1941.
- [3] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization : Algorithms and Complexity, Dover Publications, 1998.
- [4] D.-C. Dang and A. Moukrim, "Subgraph extraction and metaheuristics for the maximum clique problem", In : Journal of Heuristics 18, 5 2012.
- [5] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "An effective PSO-inspired algorithm for the team orienteering problem", In : European Journal of Operational Research 229.2, pp. 332-344, 2013.
- [6] D.-C. Dang, R. N. Guibadj, and A. Moukrim, A PSO-inspired algorithm for the team orienteering problem, Submitted preprint, 2012.
- [7] D. Feillet, P. Dejax, and M. Gendreau, "Traveling Salesman Problems with Profits", In : Transportation Science 39.2, pp. 188-205, 2005.
- [8] I.-M. Chao, B. Golden, and E. Wasil, "The Team Orienteering Problem", In : European Journal of Operational Research 88, pp. 464-474, 1996.
- [9] J. Spinrad, "Two dimensional partial orders", Ph.D. Thesis, Princeton University, 1982.
- [10] J. Edmonds, "Covers and packings in a family of sets", Bulletin of The American Mathematical Society, 68 :494-499, 1962.
- [11] L.K. Stewart, "Permutation graph structure and algorithms", Thesis, Department of

- Computer Science, University of Toronto, 1985.
- [12] M. Fischetti, J. J. Salazar González, and P. Toth, "Solving the Orienteering Problem through Branch-and-Cut", In : *INFORMS Journal on Computing* 10, pp. 133-148, 1998.
- [13] M. Garey and D. S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness*, Freeman, san Fransisco, 1979.
- [14] M. Kantor and M. Rosenwein, "The orienteering problem with time windows", In : *The Journal of the Operational Research Society* 43.6, pp. 629-635, 1992.
- [15] Mihalis Yannakakis, "Computing the minimum fill-in is NP-complete", *SIAM Journal on Algebraic and Discrete Methods*, 2(1) :77-79, 1981.
- [16] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The Team Orienteering Problem : A survey", In : *European Journal of Operational Reasearch* 209, pp. 1-10, 2011.
- [17] R. El-Hajj, D.-C. Dang, A. Moukrim, "Un algorithme de branch-and-cut pour la résolution du Problème de Tournées Sélectives", Université de Technologie de Compiègne, Département Génie Informatique, France, 2013.
- [18] R.P. Dilworth, A decomposition theorem for partially ordered sets, *Annals of Mathematics*, 51, 161-165, 1950.
- [19] R. E. Gemory, "Outline of an Algorithm for Integer Solutions to Linear Programs and an Algorithm for the Mixed Integer Problem", In : *BULLETIN of the American Mathematical Society* 64.5, pp. 275-278, 1958.
- [20] R. E. Tarjan. "Depth-First Search and Linear Graph Algorithms". In : *SIAM J. Comput.* 1.2, pp. 146-160, 1972.
- [21] S. A. Cook, "The complexity of theorem-proving procedures", In *Proceedings of the third annual ACM symposium on Theory of computing, STOC '71*, pages 151-158, ACM, 1971.
- [22] S. E. Butt and T. M. Cavalier, "A heuristic for the multiple tour maximum collection problem", In : *Computers and Operations Research* 21, pp. 101-111, 1994.
- [23] Szpilrajn, E., "Sur l'extension de l'ordre partiel", *Fundamenta Mathematicae* 16 : 386-389, ISSN 0016-2736, 1930.

- [24] T. Hao, E. Miller-Hooks, "A Tabu Search Heuristic for the Team Orienteering Problem", *Computers and Operation Research*, vol 32, pp 1379-1407, 2005.
- [25] Z. Zhang, O. Che, B. Cheang, A. Lim, and H. Qin, "A memetic algorithm for the multiperiod vehicle routing problem with profit", In : *European Journal of Operational Research* 229, pp. 573-584, 2013.