INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Optimisation du damage d'une station de ski

Stage de 3ème année - Département Génie Mathématique

Théo Guyard

Table of contents

Introduction

Problème

Pour entretenir une station il faut damer les pistes tous les soirs.

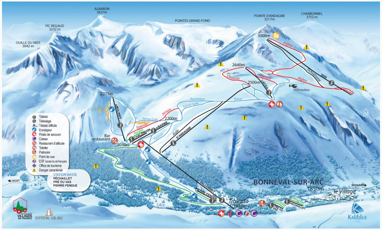
Problème

Quelle est la manière optimale d'effectuer ce damage ?

Hypothèses :

- Chaque dameuse est affectée à un dépôt
- Toutes les pistes sont praticables dans au moins un sens
- On peut atteindre n'importe quelle piste de la station
- Toutes les pistes ne requièrent pas forcément d'être damées
- On ne peut pas surcharger de travail une dameuse
- On peut associer à une piste un coût de passage, un coût de damage et une demande

Objectif



Nombre de véhicules + Plan des pistes + Renseignements sur les pistes

$\Downarrow$

Parcours de chaque dameuse pour la prochaine session de damage

Propriétés et remarques

Solution cherchée

Solution approchée $\rightarrow$ dameuse en trop

On cherche une solution exacte !

Taille du problème

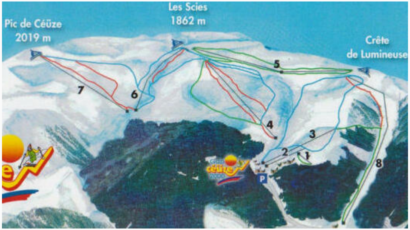
Taille des stations $\rightarrow$ petit réseau

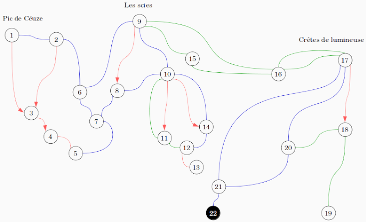
Plus de choix dans les méthodes de résolution

Modélisation

Modélisation sous forme de graphe :

- Jonction de piste / dépôt : sommet
- Piste : arête/arc
- Ensemble d'arcs/arêtes requis
- Effectuer des cycles en partant des dépôts





Problèmes de routage

Problème du postier Chinois (CPP)

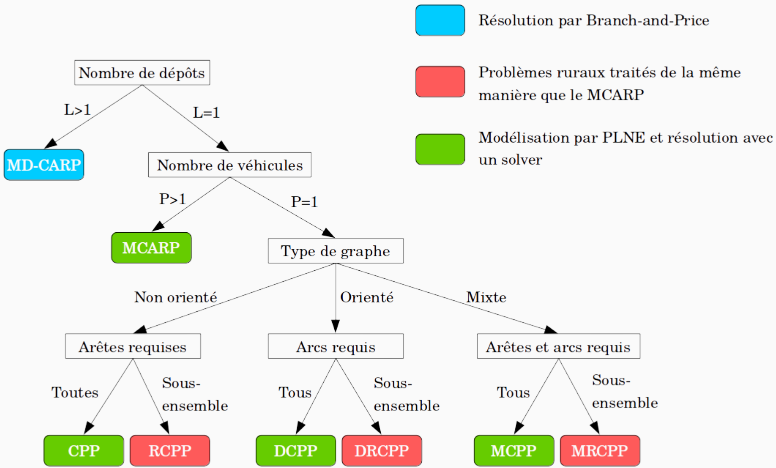
Trouver un cycle qui passe par toutes les arêtes d'un graphe non orienté.
Décliné en problèmes similaires (DCPP, MCPP, rural ...).

Problème de routage avec capacité (CARP)

- Un ou plusieurs véhicules (nombre fixé) et un dépôt
- Demande sur les arêtes
- Les cycles doivent respecter une capacité maximale

MCARP pour un graphe mixte et MD-MCARP pour plusieurs dépôts

Classification des problèmes



Modélisation par PLNE

MCARP : un ou plusieurs véhicules, graphe mixte, dépôt unique

Variables

- $x_{ij}^{p} = \begin{cases} 1 \text{ si le véhicule } p \text{ sert l'arc } (i,j) \in R \\ 0 \text{ sinon} \end{cases}$
- $y_{ij}^{p},\ (i,j) \in A$ est le nombre de fois que le véhicule p passe par l'arc (i,j) sans le servir
- $f_{ij}^{p},\ (i,j) \in A$ est le flot sur l'arc $(i,j) \in A$ correspondant à la demande restante dans le chemin effectué par le véhicule p

Le nombre de fois que le véhicule p passe sur un arc est $x_{ij}^{p} + y_{ij}^{p}$.

Type des variables

$$y_{ij}^p \in \mathbb{N} \qquad \forall (i,j) \in A, \qquad \forall p \in \{1, ..., P\} \tag{1}$$

$$f_{ij}^p \geq 0 \qquad \forall (i,j) \in A, \qquad \forall p \in \{1, ..., P\} \tag{2}$$

Fonction objectif

$$\text{Minimize : } \sum_{p=1}^{P} \left[\sum_{(i,j) \in R} x_{ij}^p c_{ij} + \sum_{(i,j) \in A} y_{ij}^p d_{ij} \right] \tag{3}$$

MCARP

Continuité des routes

$$\sum_{(i,j)\in R} x_{ij}^p + \sum_{(i,j)\in A} y_{ij}^p = \sum_{(j,i)\in R} x_{ji}^p + \sum_{(j,i)\in A} y_{ji}^p \qquad \forall i \in S \qquad \forall p \in \{1, ..., P\} \tag{4}$$

Couverture des arcs/arêtes requis

$$\sum_{p=1}^{P} x_{ij}^p = 1 \qquad \forall (i,j) \in A_R \tag{5}$$

$$\sum_{p=1}^{P} (x_{ij}^p + x_{ji}^p) = 1 \qquad \forall (i,j) \in E_R \tag{6}$$

Élimination des sous-tours et prise en compte de la capacité

$$\sum_{(0,j)\in A} y_{0j}^p + \sum_{(0,j)\in R} x_{0j}^p \leq 1 \tag{7}$$

$$\sum_{(j,i)\in A} f_{ji}^p - \sum_{(i,j)\in A} f_{ij}^p = \sum_{(j,i)\in R} x_{ji}^p q_{ji} \qquad \forall i \in S \setminus 0 \qquad \forall p \in \{1, ..., P\} \tag{8}$$

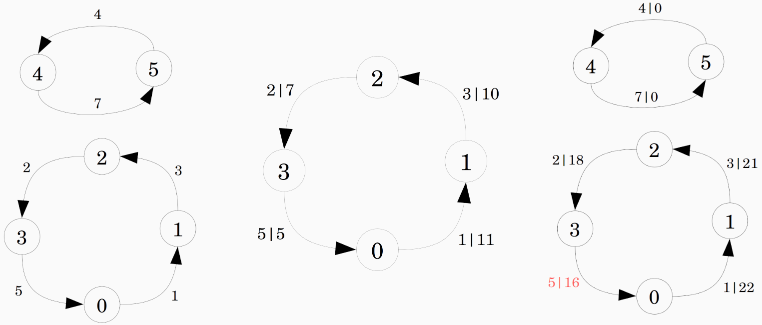
$$\sum_{(0,j)\in A} f_{0j}^p = \sum_{(i,j)\in R} x_{ij}^p q_{ij} \qquad \forall p \in \{1, ..., P\} \tag{9}$$

$$\sum_{(i,0)\in A} f_{i0}^p = \sum_{(i,0)\in R} x_{i0}^p q_{i0} \qquad \forall p \in \{1, ..., P\} \tag{10}$$

$$f_{ij}^p \leq W(x_{ij}^p + y_{ij}^p) \qquad \forall (i,j) \in A \qquad \forall p \in \{1, ..., P\} \tag{11}$$

Élimination des sous-tours

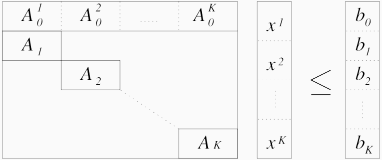
C'est plus clair avec une explication !



Algorithme de Branch-and-Price

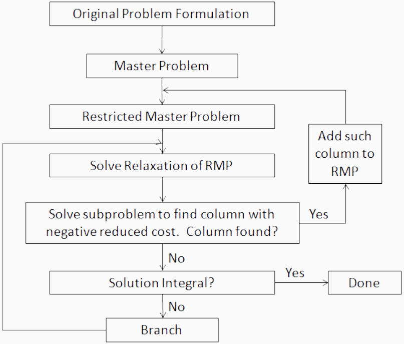
Principe

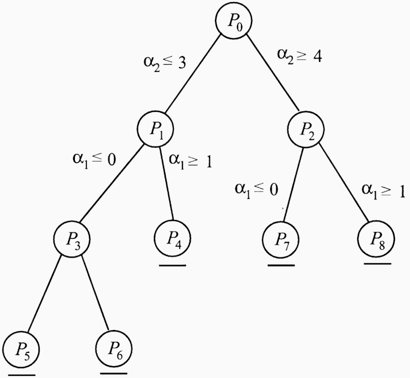
Problème décomposable en problème maître et problèmes esclaves indépendants.



Problème maître réduit : Problème maître avec un sous-ensemble de colonnes.
Coût réduit d'un problème esclave : Pertinence de la prise en compte de la solution esclave dans le problème maître. Il dépend des variables duales du problème maître et des variables du problème esclave.

Principe





- Solutions réelles $\rightarrow$ borne inférieure pour le nœud
- Solutions entière $\rightarrow$ borne supérieure globale
- On ne va pas explorer toutes les branches

Problème maître et problèmes esclaves

Problèmes esclave

Trouver une tournée qui, pour un dépôt donné :

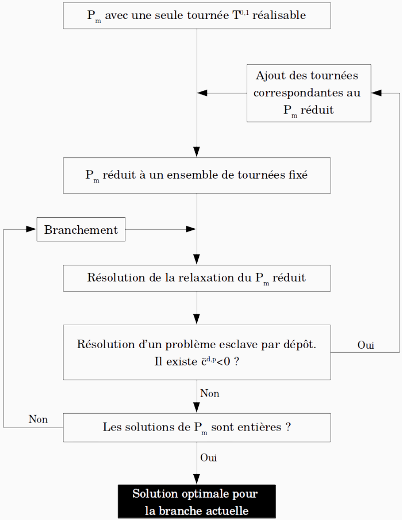
1. Minimise le coût réduit
2. Parte et arrive au dépôt
3. Soit continue
4. Ne crée par de sous-tours
5. Respecte la capacité du véhicule

Problème maître réduit

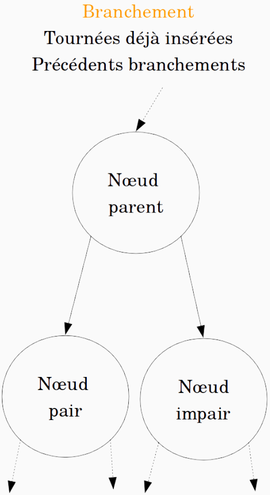
Trouver une combinaison de tournée des problèmes esclave de coût minimum permettant de :

1. Respecter le nombre maximal de véhicules
2. Couvrir tous les arcs requis
3. Couvrir toutes les arêtes requises dans un sens

Algorithme



P_m avec une seule tournée $T^{0,1}$ réalisable
Ajout des tournées correspondantes au P_m réduit
P_m réduit à un ensemble de tournées fixé
Branchement
Résolution de la relaxation du P_m réduit
Résolution d'un problème esclave par dépôt. Il existe $\tilde{c}^{d,p}<0$?
Oui
Non
Non
Les solutions de P_m sont entières ?
Oui
Solution optimale pour la branche actuelle



Branchement
Tournées déjà insérées
Précédents branchements
Nœud parent
Nœud pair
Nœud impair

Tournée initiale

Permet de fixer la première borne supérieure de la résolution :

- On fixe son coût à 10^3 fois la demande totale sur le graphe
- Elle sert tous les arcs/arêtes requis
- Si elle est choisie, c'est que le problème maître est infaisable

Intégrité des solutions

On résout une relaxation du problème maître réduit :

$$
\begin{array}{rc}
\text{Minimize :} & \sum_{T^{dp}} \alpha^{dp} c^{dp} \\
\text{Subject to :} & \dots \\
& \alpha^{dp} \geq 0 \qquad \forall T^{dp}
\end{array}
$$

- On ne peut pas avoir de "fraction" de tournée
- Solutions réelles $\rightarrow$ borne inférieure pour le nœud
- Solutions entière $\rightarrow$ borne supérieure globale

Branchements

Chaque branchement concerne un dépôt et un arc :

- Nœud pair : interdit les tournées du dépôt de servir l'arc
- Nœud impair : oblige au moins une tourné du dépôt à servir l'arc

On ne doit pas effectuer deux fois le même branchement !
Un branchement peut rendre le problème maître réduit infaisable.

Descente et complexité de résolution

La totalité des tournées est transmise lors d'un branchement :

- Le nombre de tournées des nœuds grandit vite
- La résolution du problème maître devient de plus en plus difficile

Plusieurs solutions :

- Explorer quelques fois en profondeur puis en largeur
- Heuristique pour retirer des tournées
- Utiliser la résolution avec un dépôt pour fixer une borne supérieure

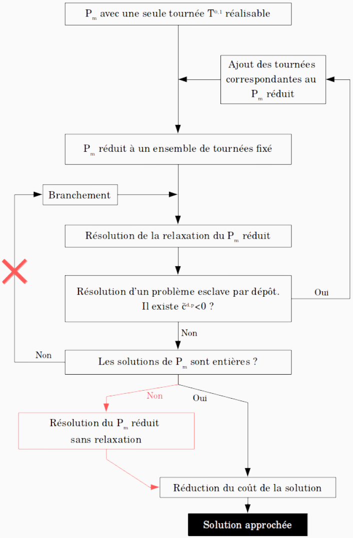
Algorithme modifié et solution approchée

Résoudre un PLNE avec les tournées introduites à la racine :

→ Tournées solutions souvent ajoutées dès la racine

→ Solution approchée

→ Ajout d'une procédure de réduction de coût à la fin de l'algorithme



Résultats

Jeux de données

Trois types de jeux de données :

- Petits jeux de données (`small`)
  - $\rightarrow$ Graphe orienté, non-orienté et mixte
  - $\rightarrow$ Partie ou totalité des arcs/arêtes requis
- Jeux de données pour le CARP (`mval`, `lpr`)
  - $\rightarrow$ Graphe orienté, non-orienté et mixte
  - $\rightarrow$ Partie ou totalité des arcs/arêtes requis
- Jeux de données de stations (`devoluy`, `ceuze`, `greoliere`)
  - $\rightarrow$ Partie de la station, tous les arcs/arêtes requis
  - $\rightarrow$ Station entière, arcs/arêtes en partie requis
  - $\rightarrow$ Station entière, totalité des arcs/arêtes requis
  - $\rightarrow$ Unique ou plusieurs dépôts

Résultats

Problèmes pour un unique dépôt :

Résolution par solver de manière optimale

Jeu de donnée	$\lvert S\rvert$	$\lvert E_R\rvert$	$\lvert E\rvert$	$\lvert A_R\rvert$	$\lvert A\rvert$	P	Algo	c_{tot}	t
devoluy-1	7	12	12	0	0	2	MCARP	60.5	0.033
devoluy-2	13	10	21	0	0	2	MCARP	67	0.068
devoluy-3	13	21	21	0	0	2	MCARP	106.5	0.111
ceuze-1	10	9	9	4	4	2	MCARP	59	0.029
ceuze-2	22	17	26	5	7	2	MCARP	122	0.051
ceuze-3	22	26	26	7	7	2	MCARP	177.5	0.199
greoliere-1	38	19	30	3	5	2	MCARP	148	0.068
greoliere-2	38	37	60	3	5	2	MCARP	239	0.218
greoliere-3	38	60	60	5	5	3	MCARP	384	15.613

Résultats

Problèmes pour un plusieurs dépôts :

Résolution par Branch-and-Price de manière optimale

Jeu de donnée	$\|S\|$	$\|S_D\|$	$\|E_R\|$	$\|E\|$	$\|A_R\|$	$\|A\|$	P	Algo	c_{tot}	t
devoluy-1	8	2	12	12	0	0	2	BnP-opt	57	12.185
devoluy-2	13	2	10	21	0	0	2	BnP-opt	61	9.558
devoluy-3	13	2	21	21	0	0	3	BnP-opt	107.5	236.006
ceuze-1	10	2	9	9	4	4	2	BnP-opt	59	20.445
ceuze-2	22	2	17	26	5	7	2	BnP-opt	114	179.202
ceuze-3	22	2	26	26	7	7	2	BnP-opt	159.5	12684.146
greoliere-1	38	2	19	30	3	5	2	BnP-opt	148	154.679

Résolution par Branch-and-Price de manière approchée

Jeu de donnée	...	P	Algo	c_{tot}	$\tilde{c}_{tot}$	GAP	t
devoluy-1	"	2	BnP-approx	60.5	60.5	5.8%	5.651
devoluy-2	"	3	BnP-approx	61	61	0%	6.088
devoluy-3	"	3	BnP-approx	127.5	119.5	10.0%	12.171
ceuze-1	"	3	BnP-approx	64	62	4.7%	5.053
ceuze-2	"	2	BnP-approx	116	114	0%	26.314
ceuze-3	"	3	BnP-approx	182.5	169.5	5.9%	37.210
greoliere-1	"	3	BnP-approx	163	162	8.6%	18.767

Conclusion

Conclusion

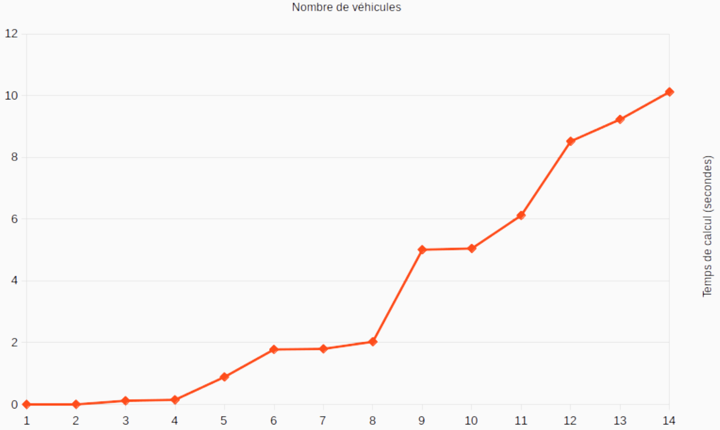
Résolution par solver plus rapide que par Branch-and-Price

- `CPLEX` codé par des équipes entières avec des années d'optimisation de code
- Algorithme de Branch-and-Price codé sur 4 semaines
- Beaucoup d'affichage, pas codé pour être optimisé
- Grande marge de progression pour améliorer la vitesse

Solutions approchées très convenables, parfois optimales

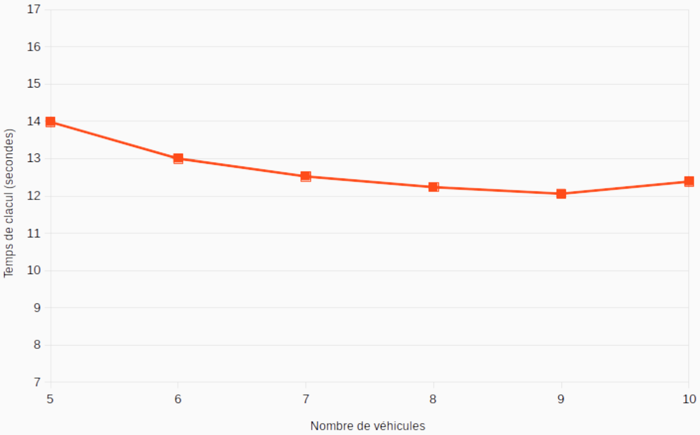
- Vraie nécessité d'une solution optimale ?

Sensibilité aux données



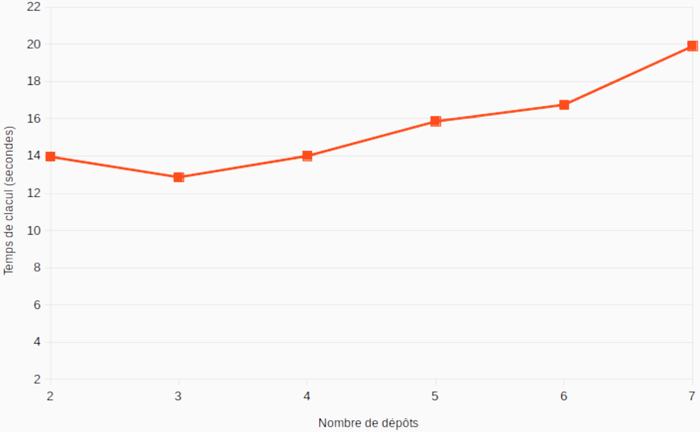
Sensibilité au nombre de véhicules pour la résolution par solver

Sensibilité aux données



Sensibilité au nombre de véhicules pour la résolution par Branch-and-Price

Sensibilité aux données



Sensibilité au nombre de dépôts pour la résolution par Branch-and-Price

Conclusion

Résolution par Branch-and-Price peu sensible aux données du problème, avec un code plus rapide :

- Solution optimale
- Temps raisonnable
- Peu sensible à l'ajout de véhicules ou de dépôts

→ Algorithme adapté à la résolution du problème de damage

Conclusion

- Problème concret avec des contraintes pratiques
- Recherches bibliographiques
- Nouveaux concepts de recherche opérationnelle
- Nouveau langage de programmation
- Utilisation de solvers
- Coder un algorithme de Branch-and-Price de A à Z