SCIENCES SORBONNE UNIVERSITÉ

Sorbonne université

Modèles et Applications en Ordonnancement et optimisation combinAtoire

M2 ANDROIDE

Rapport du projet de production et intégration intégré

lien du git du projet

Etudiants :
Corcoral Thomas
Toullalan Antoine

Année : 2022 - 2023

Table des Matières

1 Introduction

Ce projet est une résolution du problème de production et de distribution intégrée par des méthodes exactes et approchées. Ce projet, codé en Julia, est sous forme d'un notebook pour faciliter la compréhension du code dans lequel nous avons utilisé le solveur CPLEX pour les deux types de méthodes. Le caractère NP-difficile du problème nécessite l'utilisation de méthodes approchée pour avoir des solutions pour des instances de grande taille.
Nous avons aussi évalué l'efficacité de ces méthodes grâces aux instances fournies de taille variables : cette évaluation permet de comparer l'efficacité de nos méthodes exacte et approchées par la visualisations de statistiques correspondantes.

Ce projet se divise donc en trois partie : d'abord le codage d'une méthode de résolution exacte par PLNE mais peu efficace puis le développement de méthodes de résolution approchée reposant sur différentes heuristiques. Enfin l'ajout d'un Branch cut au plne de résolution qui permet de coder une méthode exacte efficace.

Dans ce rapport, nous allons expliquer les choix que nous avons fait pour coder ces méthodes, la construction de notre code et les résultats des évaluations expérimentales

2 | Architecture du code

Nous avons décidé de coder ce projet en partie sur des notebook jupyter afin d'améliorer la compréhension du code. Nous avons crée un fichier julia pour chaque heuristique que nous appelons ensuite dans le notebook, pour la méthode exacte nous avons un fichier julia pour le problème du VRP et un pour le problème du PDI afin que le code du notebook soit compacte et compréhensible.

Pour les méthodes approchées et exactes, nous avons crée une méthode de lecture des instances qui prend en argument le nom du fichier de l'instance et renvoie un dictionnaire qui contient toutes les informations. Pour faciliter la lecture des instances nous avons aussi crée une structure Client qui permet de contenir ses coordonnées, sa demande...

3 Construction des méthodes de résolution heuristique et méthode exacte

Nous avons d'abord codé le PLNE décrit dans le sujet en implémentant les fonctions objectif et les contraintes.Nous avons pu ainsi résoudre de manière optimale le problème du LSP et du VRP, comme décrit dans la partie évaluation expérimentale, ce type de résolution est néammoins très couteux en temps de calcul et ne permet pas d'évaluer des instances de taille supérieure à 14.

Les PLNE codés sont ceux correspondant au LSP (contraintes de 1 à 5) et au VRP (contraintes de 6 à 10), on a utilisé la formulation MTZ. Avec cette méthode nous avons des résuktats optimaux mais la taille des instances ne peut pas dépasser 14. Nous avons ensuite codé 3 heuristiques pour le VRP car ce problème est NP-difficile donc remplacé son PLNE qui augmente considérablement le temps de calcul est primordial pour avoir une méthode de résolution efficace. Les heuristiques codées sont donc : Clark-Wright ("Clark-Wright.jl"), Bin-Packing ("binPacking.jl") et l'heuristique sectorielle ("sectorielle.jl")

4 Validation

4.1 PLNE et méthodes heuristiques

Instance	BP	CW	SECT	PLNE
A_014_#ABS1_15_1.prp	42706.0	42706.0	42706.0	40390.0
A_014_ABS10_15_1.prp	78929.0	78929.0	78929.0	72875.0
A_014_ABS10_15_2.prp	66830.0	66830.0	66830.0	61809.0
A_014_ABS10_15_3.prp	60017.0	60017.0	60017.0	53958.0
A_014_ABS10_15_4.prp	71460.0	71460.0	71460.0	66253.0
A_014_ABS10_15_5.prp	69999.0	69999.0	69999.0	62115.0
A_014_ABS11_15_1.prp	78929.0	78929.0	78929.0	72813.0
A_014_ABS11_15_2.prp	66830.0	66830.0	66830.0	61908.0
A_014_ABS11_15_3.prp	60160.0	60014.0	60038.0	54052.0
A_014_ABS11_15_4.prp	71405.0	71460.0	71460.0	66493.0
A_014_ABS11_15_5.prp	69999.0	69999.0	69999.0	62215.0
A_014_ABS12_15_1.prp	78677.0	78929.0	78677.0	73039.0
A_014_ABS12_15_2.prp	66830.0	66830.0	66830.0	61809.0
A_014_ABS12_15_3.prp	60017.0	60014.0	60017.0	54095.0
A_014_ABS12_15_4.prp	71671.0	71671.0	71671.0	66572.0
A_014_ABS12_15_5.prp	70064.0	70064.0	70064.0	63021.0
A_014_ABS13_15_1.prp	47153.0	47153.0	47153.0	44486.0
A_014_ABS13_15_2.prp	43263.0	43263.0	43263.0	40325.0
A_014_ABS13_15_3.prp	35964.0	35964.0	35964.0	34125.0
A_014_ABS13_15_4.prp	43593.0	43593.0	43593.0	41723.0
A_050_ABS27_50_3.prp	699634.0	699769.0	701452.0	X
A_100_ABS5_100_4.prp	469024.0	450644.0	469190.0	X
B_050_instance1.prp	1.10e6	1.12e6	997695	X
B_100_instance1.prp	1.99e6	1.73e6	2.41e6	X

TABLE 4.1 – Comparaison des couts des différents algorithmes

Instance	BP	CW	SECT	PLNE
A_014_#ABS1_15_1.prp	1.926387007	1.890944508	2.146556564	37.886187371
A_014_ABS10_15_1.prp	1.829061202	1.790297622		4.769033897
A_014_ABS10_15_2.prp	1.750326588	1.726949687	1.847169659	11.7505842060
A_014_ABS10_15_3.prp	1.569661105	1.527180554	1.481423318	2.61285753
A_014_ABS10_15_4.prp	1.598236528	1.569218129	1.750965813	34.3992969
A_014_ABS10_15_5.prp	1.698740424	1.578604553	1.534346029	424.28090642
A_014_ABS11_15_1.prp	1.7056369	1.62563182	1.694804211	13.489697893
A_014_ABS11_15_2.prp	1.618508841	1.589822659	1.56122272	15.01331328
A_014_ABS11_15_3.prp	1.525075653	1.396923848	1.37817487	2.045280162
A_014_ABS11_15_4.prp	1.538525991	1.444517601	1.436966515	341.698009041
A_014_ABS11_15_5.prp	1.651280324	1.569473628	1.595629585	170.484382415
A_014_ABS12_15_1.prp	1.797743362	1.664392471	1.648752193	103.562622198
A_014_ABS12_15_2.prp	1.660897373	1.554696315	1.571285148	7.266050129
A_014_ABS12_15_3.prp	1.42181528	1.3895544	1.40326764	1.881322758
A_014_ABS12_15_4.prp	1.476371172	1.421429392	1.412634166	47.173273628
A_014_ABS12_15_5.prp	1.688484945	1.569410347	1.564008721	600.011769522
A_014_ABS13_15_1.prp	1.561328696	1.614416232	1.57075004	92.504890712
A_014_ABS13_15_2.prp	1.627825697	1.431587022	1.450725527	66.976524696
A_014_ABS13_15_3.prp	1.575090083	1.557748956	1.547908108	7.353446701
A_014_ABS13_15_4.prp	1.553840176	1.556890089	1.518796654	150.605394213
A_050_ABS27_50_3.prp	6.317530464	6.84585293	5.200792413	X
A_100_ABS5_100_4.prp	18.446655863	18.901837419	18.460858187	X
B_050_instance1.prp	46.801568012	45.818584245	51.11262741	X
B_100_instance1.prp	185.25681445	193.95180455	90.241010769	X

Table 4.2 – Comparaison des temps d'execution des différents algorithmes (en secondes)

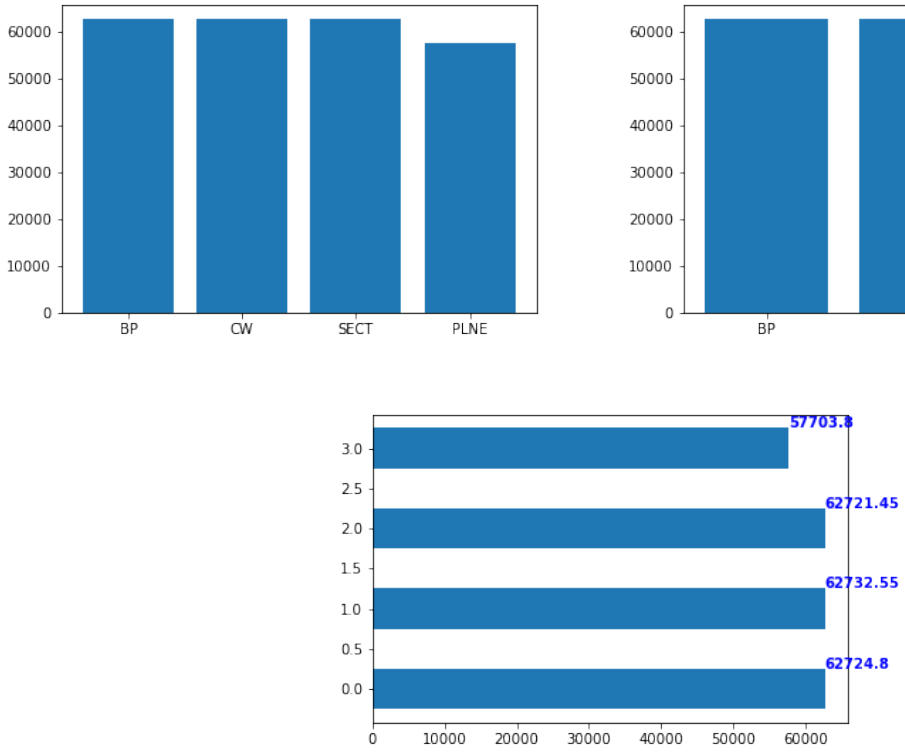

FIGURE 4.1 – Comparaison de la moyenne du cout obtenu sur tous les algorithmes.

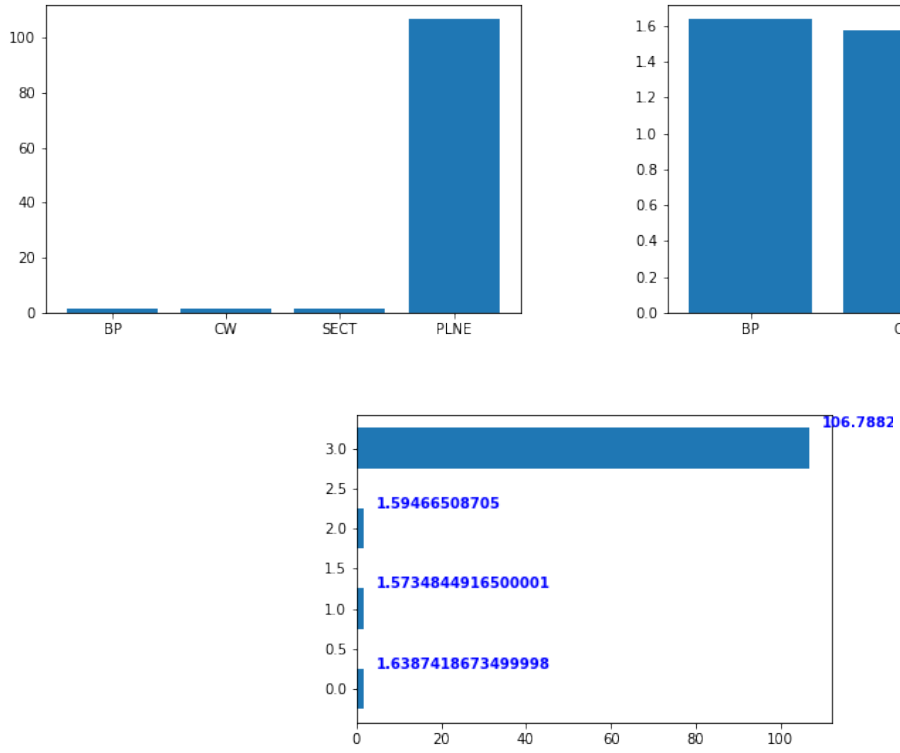

FIGURE 4.2 – Comparaison de la moyenne de temps d'execution obtenu sur tous les algorithmes.

On voit ici que le PLNE a pour toutes les instances (de taille 14) un cout nettement inférieur à celui des heuristiques mais le temps d'execution du PLNE est très supérieur par rapport à ces heuristiques. On voit ainsi que le temps d'execution moyen pour le PLNE est environ 100 fois supérieur au temps d'execution moyen des heuristiques. Le cout moyen des solutions renvoyé par le PLNE (57703) est légèrement inférieure au cout moyen des heuristiques (environ 62700) donc les solutions renvoyées par la méthode exacte sont plus performantes pour le cout.
Nous voyons que les 3 heuristiques ont des temps de calcul et des cout de solution assez similaires même si l'heuristique CW a un temps de calcul légèrement inférieur aux 2 autres. Ainsi Nous voyons l'intérêt d'utiliser des heuristiques pour avoir une méthode de résolution efficace au niveau du temps de calcul.

4.2 Exemple de résultat

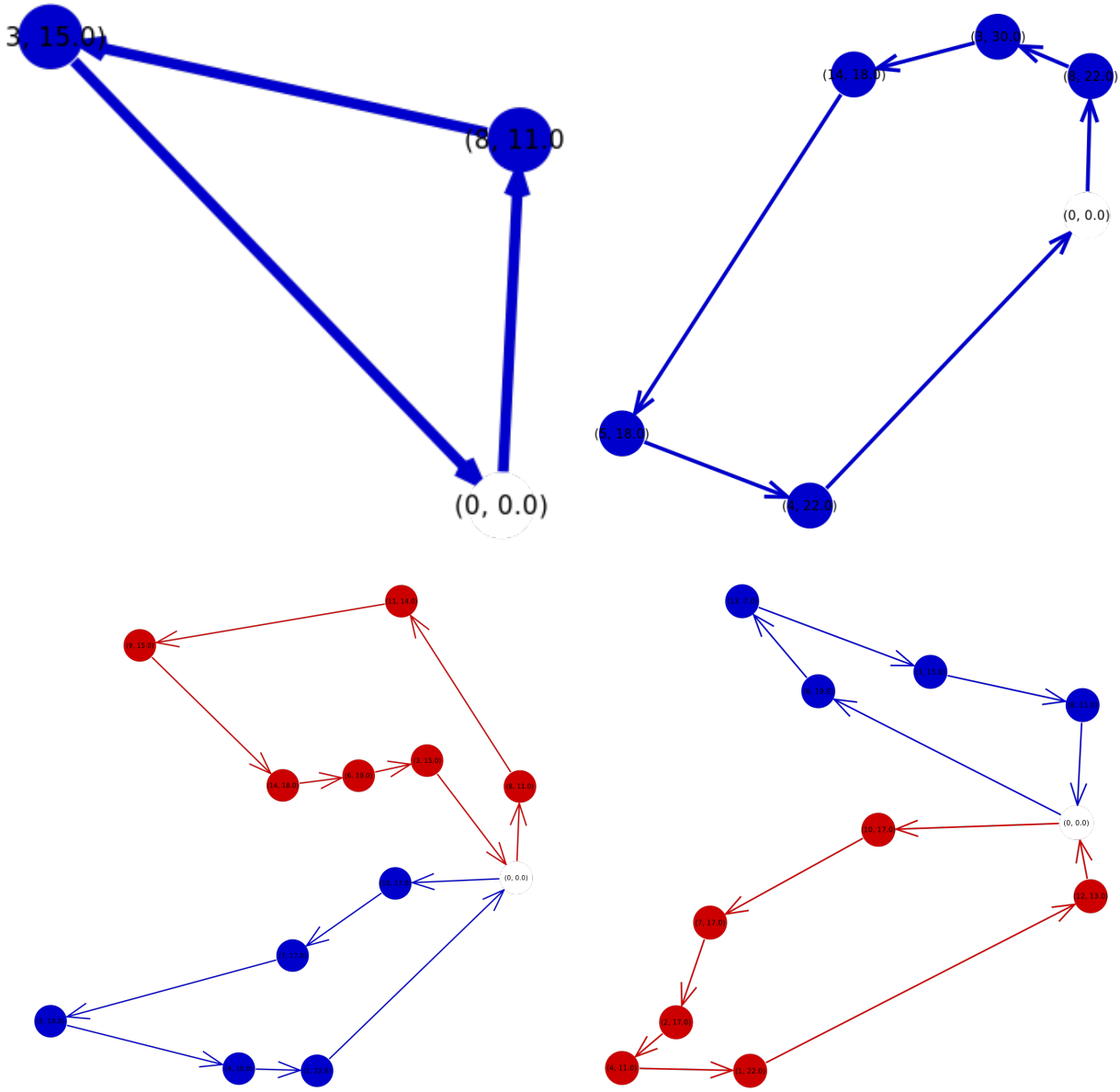

FIGURE 4.3 – Exemple sur l'instance A_014_ABS1_15_3. (temps 2, 3, 5, 6

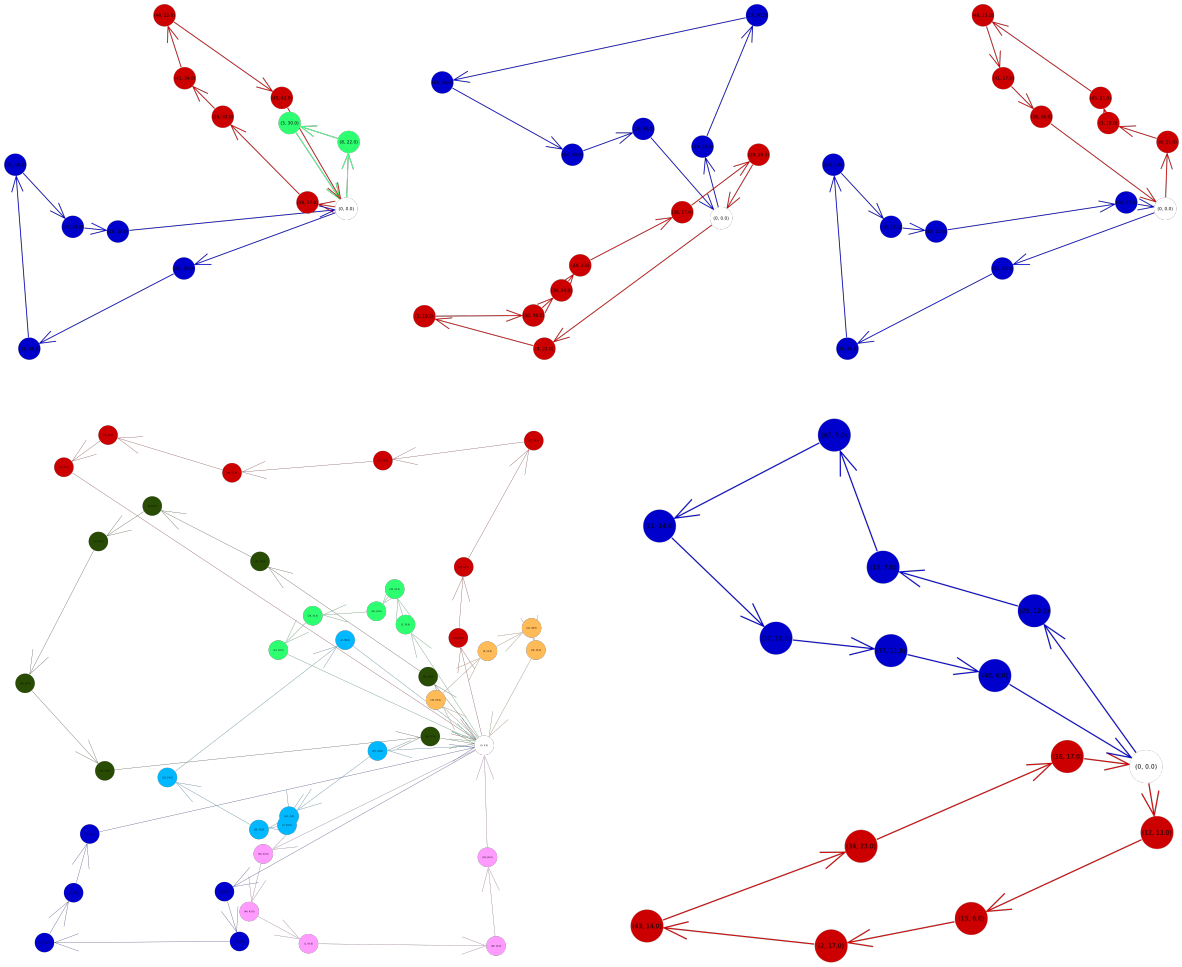

FIGURE 4.4 – Exemple sur l'instance A_050_ABS27_50_3. (temps 2, 3, 4, 5, 6)

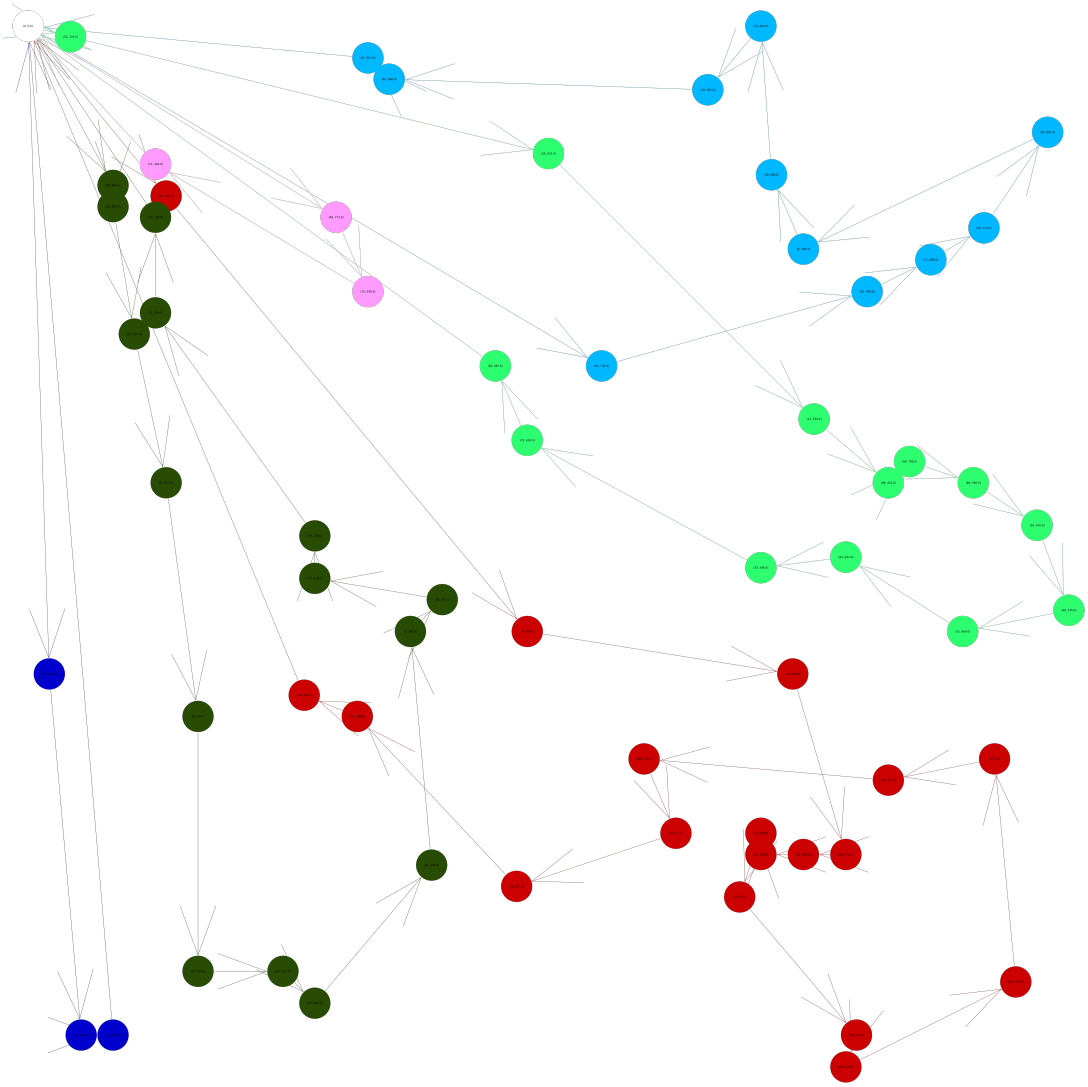

FIGURE 4.5 – Exemple d'un pas de temps sur l'instance B_100_instance1 (temps 1)

Ici chaque ensemble de circuit représente la ou les tournées pour un tick et la tournée de chaque camion est représenté de couleurs différente.
Nous avons donné des exemples de tournées avec l'heuristique CW pour des instances de taille 14,50 et 100.Nous voyons bien que la taille des solution (le nombre de tournées nécessaires à chaque tick) augmente fortement lorsqu'on augmente légèrement le nombre de client.

5 Conclusion

Nous avons donc implémenté des méthodes de résolutions exactes et approchées efficaces du problèmes du PDI en codant des méthodes heuristiques qui renvoie des solutions avec un cout proche de l'optimal (on compare avec les solutions du PLNE). Ces méthodes approchées permettent ainsi d'avoir des solutions performantes pour des instances avec des tailles allant jusqu'à 100 au moins.
Ce projet a permis de prendre en main différentes heuristiques du problème VRP mais aussi d'implémenter des PLNE assez complexe. Nous avons tenter d'implémenter, dans la partie validation expérimentale, un programme test qui execute les méthodes rapprochées et exactes sur toutes les instances de taille 14 puis creer pour chaque méthode 4 point correspondant aux 4 classes dans les instances de taille 14 et chaque point aurait comme coordonnées (temps de calcul moyen pour les instances de cette classe, cout moyen des solutions pour les instances de cette classe) afin de mieux observer les performances des heuristiques et du PLNE en fonction de l'augmentation des cout de trajet et des couts de production. Mais ce test a "crashé" sans doute a cause des temps de calcul très long du PLNE.