Université Libre de Bruxelles

Master in cybersecurity

Informatique Fondamentale [INFOF302]

Projet: Logique Propositionnelle et Utilisation de l'Outil MiniSat

Hanquin Benjamin - 000495418 Tchiagou Tékédo Loïc - 000489252

Contents

1 Question 1

1.1 Énoncé

Pour toute instance du problème (i.e., la configuration initiale), construire une formule ΦI en FNC de la logique propositionelle tel que ΦI est satisfaisable si et seulement si il existe une solution au problème. La construction de cette formule doit être expliquée dans le rapport en détail.

1.2 Variables

Nous utiliserons $V_{v,t,i,j}$ qui est vrai lorsque la case supérieure gauche d'une voiture v ce trouve en position i,j au tour t. N_t, N_v, N_n seront utilisés respectivement comme "nombre de mouvements, nombre de voitures et taille du tableau":

- v : identifiant d'une voiture
- t : temps (tour de jeu)
- i : abscisse début de la voiture
- j : ordonnée de la voiture
- la largeur, la longueur et l'orientation du véhicule sont permanente, donc on peut les récupérer avec l'id du véhicule

Nous utiliserons $M_{v,t,d}$ pour un mouvement qui est vrai lorsque on déplace une voiture v dans la direction d à l'instant t :

- v : identifiant d'une voiture
- t : temps (tour de jeu)
- d : direction de la voiture : U(p) D(own), R(ight), L(eft)

(nous pourrions aussi utiliser i et j comme identifiant de la voiture)

NB : Lorsque nous utilisons par exemple $\bigwedge_{i,j=0}^{N_n}$. Le i et le j sont des valeurs distinctes qui s'incrémentent indépendamment, nous utilisons cette écriture pour la simplicité et la lisibilité.

1.3 Construction de la formule

1. Configuration initiale

(a) Toutes les voitures doivent être comprises dans le plateau (pas dépasser des bornes du tableau). Où P étant le plateau, ce qui signifie que, i,j,i+largeur et j+hauteur sont en dehors du plateau lorsque c'est inférieur à 0 ou supérieur à la taille du tableau moins 1.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{i,j}^{N_n} \neg V_{v,t,i,j}$$

Avec

- $i, j, i + v.largeur, i + v.hauteur > 0$
- $i, j, i + v.largeur, i + v.hauteur < N_n$

(b) Aucune voiture ne peuvent partager une même case (se chevaucher).

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} V_{v,t,i,j} \implies \neg V_{v',t,i',j'}$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} \neg V_{v,t,i,j} \lor \neg V_{v',t,i',j'}$$

Où $(v \neq v'\ et\ i + v.largeur - 1 < i'\ et\ j + v.hauteur - 1 < j')\ ou\ (i' + v'.largeur - 1 < i\ et\ j' + v'.hauteur - 1 < j)$

(c) Les voitures au tour 0 doivent correspondre aux voitures en input. Car[] est le tableau des voitures.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{i,j}^{N_n} \neg V_{v,t,i,j}\ quand\ i, j \neq car[v].x, car[v].y$$

2. Clauses du jeu

(a) Il faut au moins un mouvement à chaque coup.

$$\bigwedge_{t=0}^{N_t} (\bigvee_{v=0}^{N_v} \bigvee_{d=0}^{4} M_{v,t,d})$$

(b) Il faut au maximum un seul mouvement à la fois. C'est à dire que si un mouvement est fait dans une direction, un autre mouvement ne peut être fait au même temps t dans une autre direction ou d'une autre voiture.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{d,d'=0}^{4} M_{v,t,d} \implies \neg M_{v',t,d'}\ avec\ v, d \neq v', d'$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{d,d'=0}^{4} \neg M_{v,t,d} \vee \neg M_{v',t,d'} \; avec \; v, d \neq v', d'$$

(c) Il faut au moins chaque voiture à chaque tour de jeu.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} (\bigvee_{i,j=0}^{N_n} V_{v,t,i,j})$$

(d) Lorsqu'une voiture se déplace son déplacement ne peut excéder une case. Donc entre 2 tours, une voiture ne peut pas avoir plus de 2 cases de différence.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{i,i',j,j'=0}^{N_n} V_{v,t,i,j} \implies \neg V_{v,t+1,i',j'}$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{i,i',j,j'=0}^{N_n} \neg V_{v,t,i,j} \vee \neg V_{v,t+1,i',j'}$$

Où $|i - i'| + |j - j'|$ ne peut excéder 1.

(e) Un déplacement n'est possible que dans la direction du véhicule (si horizontale Right et Left, si verticale Up et Down). Où vV est une voiture verticale et vH est horizontale.

$$\bigwedge_{v=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (V_{vH,t,i,j} \implies \neg M_{v,t,Up} \wedge \neg M_{v,t,Down}) \wedge (V_{vV,t,i,j} \implies \neg M_{v,t,Right} \wedge \neg M_{v,t,Left})$$

$$\equiv$$

$$\bigwedge_{v=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (\neg V_{vH,t,i,j} \vee (\neg M_{v,t,Up} \wedge \neg M_{v,t,Down})) \wedge (\neg V_{vH,t,i,j} \vee (\neg M_{v,t,Right} \wedge \neg M_{v,t,Left}))$$

$$\equiv$$

$$\bigwedge_{v=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (\neg V_{vH,t,i,j} \vee \neg M_{v,t,Up}) \wedge (\neg V_{vH,t,i,j} \vee \neg M_{v,t,Down}) \wedge$$

$$(\neg V_{vV,t,i,j} \vee \neg M_{v,t,Right}) \wedge (\neg V_{vV,t,i,j} \vee \neg M_{v,t,Left})$$

(f) Un déplacement n'est possible que si au moins une des deux cases adjacentes disponible (voir règle précédente) est libre (pas de voiture).

i. Pas de collision vers le haut, il ne faut pas de voiture verticale au dessus (en colonne i commencant en j-o avec o qui doit être égal a v'.hauteur) ni de voiture horizontale sur la case du dessus de la voiture (donc qui peut commencer en i-r sur la ligne j-1 avec r qui doit être plus petit ou égal à v'.largeur-1).

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (M_{v,t,Up} \wedge V_{v,t,i,j}) \implies (\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j-1}) \wedge (\bigwedge_{o=0}^{N_n} \neg V_{v',t,i,j-o})$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (\neg M_{v,t,Up} \vee \neg V_{v,t,i,j}) \vee ((\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j-1}) \wedge (\bigwedge_{o=0}^{N_n} \neg V_{v',t,i,j-o}))$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} \bigwedge_{r=0}^{N_n} \bigwedge_{o=0}^{N_n} (\neg M_{v,t,Up} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i-r,j-1}) \wedge (\neg M_{v,t,Up} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i,j-o})$$

Avec

- $v \neq v'$
- $v'.largeur - 1 >= r$
- $v'.hauteur == o$
- $i - r \ et \ j - o >= 0$

ii. Pas de collision vers le bas, il ne faut pas de voiture sur la case du dessous de la voiture (la voiture peut commencer en i-r sur la ligne j+v.hauteur avec r qui doit être plus petit ou égal a la v'.largeur -1).

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (M_{v,t,Down} \wedge V_{v,t,i,j}) \implies (\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j+v.hauteur})$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} \bigwedge_{r=0}^{N_n} (\neg M_{v,t,Down} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i-r,j+v.hauteur})$$

Avec

- $v \neq v'$
- $v'.largeur - 1 >= r$

- $i - r \;\; >= 0$

iii. Pas de collision vers la droite, il ne faut pas de voiture sur la case à droite de la voiture (donc en i+v.largeur sur la ligne j-r avec r qui doit être plus petit ou égal à v'.hauteur-1).

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (M_{v,t,Right} \wedge V_{v,t,i,j}) \implies (\bigwedge_{r=0}^{N_n} \neg V_{v',t,i+v.largeur,j-r})$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} \bigwedge_{r=0}^{N_n} (\neg M_{v,t,Right} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i+v.largeur,j-r})$$

Avec

- $v \neq v'$
- $v'.hauteur - 1 >= r$
- $j - r \;\; >= 0$

iv. Pas de collision vers la gauche, il ne faut pas de voiture verticale à gauche (en colonne i-1 commencant en j-o avec o qui doit être plus petit ou égal à v'.hauteur-1) ni de voiture horizontale sur la case à gauche de la voiture (donc qui peut commencer en i-r sur la ligne j avec r qui doit être égal à v'.largeur).

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (M_{v,t,Left} \wedge V_{v,t,i,j}) \implies (\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j-1}) \wedge (\bigwedge_{o=0}^{N_n} \neg V_{v',t,i-1,j-o})$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (\neg M_{v,t,Up} \vee \neg V_{v,t,i,j}) \vee ((\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j}) \wedge (\bigwedge_{o=0}^{N_n} \neg V_{v',t,i-1,j-o}))$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} \bigwedge_{r=0}^{N_n} \bigwedge_{o=0}^{N_n} (\neg M_{v,t,Left} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i-r,j}) \wedge$$
$$(\neg M_{v,t,Left} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i-1,j-o})$$

Avec

- $v \neq v'$

- $v'.largeur == r$
- $v'.hauteur - 1 >= o$
- $i - r\ et\ j - o >= 0$

(g) Toutes les voitures qui ne sont pas concernées par un mouvement ne se déplace pas. Avec $v, i, j \neq v', i', j'$.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} \bigwedge_{d=0}^{N_d} (M_{v,t,d} \wedge V_{v,t,i,j}) \implies (V_{v',t,i',j'} \iff V_{v',t+1,i',j'})$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} \bigwedge_{d=0}^{N_d} \neg(M_{v,t,d} \wedge V_{v,t,i,j}) \vee (V_{v',t,i',j'} \iff V_{v',t+1,i',j'})$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} \bigwedge_{d=0}^{N_d} (\neg M_{v,t,d} \vee \neg V_{v,t,i,j}) \vee ((\neg V_{v',t,i',j'} \vee V_{v',t+1,i',j'}) \wedge (V_{v',t,i',j'} \vee \neg V_{v',t+1,i',j'}))$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} \bigwedge_{d=0}^{N_d} (\neg M_{v,t,d} \vee \neg V_{v,t,i,j} \vee \neg V_{v',t,i',j'} \vee V_{v',t+1,i',j'}) \wedge$$

$$(\neg M_{v,t,d} \vee \neg V_{v,t,i,j} \vee V_{v',t,i',j'} \vee \neg V_{v',t+1,i',j'})$$

(h) Si on déplace une voiture et qu'elle est à une position, il faut qu'à l'instant t+1 elle se soit déplacé dans la bonne direction. Nous avons alors :

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{d=0}^{N_d} \bigwedge_{i,j,i',j'=0}^{N_n} (M_{v,t,d} \wedge V_{v,t,i,j}) \implies V_{v,t+1,i',j'}$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{d=0}^{N_d} \bigwedge_{i,j,i',j'=0}^{N_n} \neg(M_{v,t,d} \wedge V_{v,t,i,j}) \vee V_{v,t+1,i',j'}$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{d=0}^{N_d} \bigwedge_{i,j,i',j'=0}^{N_n} \neg M_{v,t,d} \vee \neg V_{v,t,i,j} \vee V_{v,t+1,i',j'}$$

Nous avons donc pour les déplacements :

- Up : i' = i et j' = j-1
- Down : i' = i et j' = j+1
- Right : i' = i+1 et j' = j
- Left : i' = i-1 et j' = j

(i) Si une voiture se déplace entre deux tours, il faut un mouvement dans la bonne direction. Nous avons alors :

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{d=0}^{N_d} \bigwedge_{i,j,i',j'=0}^{N_n} (V_{v,t,i,j} \wedge V_{v,t+1,i',j'}) \implies M_{v,t,d}$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{d=0}^{N_d} \bigwedge_{i,j,i',j'=0}^{N_n} \neg(V_{v,t,i,j} \wedge V_{v,t+1,i',j'}) \vee M_{v,t,d}$$

$$\equiv$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{d=0}^{N_d} \bigwedge_{i,j,i',j'=0}^{N_n} \neg V_{v,t,i,j} \vee \neg V_{v,t+1,i',j'} \vee M_{v,t,d}$$

Nous avons donc pour les déplacements :

- Up : i' = i et j' = j-1
- Down : i' = i et j' = j+1
- Right : i' = i+1 et j' = j
- Left : i' = i-1 et j' = j

3. Configuration finale

(a) Il faut que la voiture rouge ait été en contact au moins une fois avec la porte. (Au moins une fois car si on continue de bouger des voitures après ce soit toujours bon). La clause est : $\bigvee_{t=0}^{N_t} V_{v,t,i,j}$ où i+v.largeur = N_n et j=y de la voiture rouge.

2 Question 2

2.1 Énoncé

Implémenter à l'aide de minisat et tester sur les exemples joints à cet énoncé (en respectant le format indiqué à la section 4). Vous pouvez supposer que toute solution requerra au maximum 100 déplacements (d'une case).

2.2 Réalisation

Voir Main.cpp

3 Question 3

3.1 Énoncé

Généralisez le problème aux véhicules de taille de 2 à N-1. Détaillez cette généralisation dans votre rapport et adaptez votre implémentation.

3.2 Réalisation

Étant donné que nous ayons à chaque fois pris les

4 Question 4

4.1 Énoncé

Une variante du problème rush hour ajoute des blocs fixes à l'intérieur du parking. Adaptez votre formule afin de prendre en compte ces blocs qui posent obstacle au déplacements des véhicules. Détaillez cette modi- fication dans votre rapport et adaptez votre implémentation.

4.2 Explications

4.2.1 Variable

Il faudra ajouter une nouvelle variable pour les murs qui sera : $W_{w,i,j}$ où w est l'ID du mur et où i et j sont les abscisses et les ordonnées du plateau. L'Id du mur n'est pas utile dans ce cas ci. Mais pourrait être utile si une règle indiquait que un mur pourrait être plus grand que de taille 1x1. Avec N_w étant le nombre de murs.

4.2.2 Clauses

Une fois que toutes les propositions ont été ajoutées.

Il faut initialisé que les murs doivent correspondre aux murs en input. fixed[] étant le tableau des murs.

$$\bigwedge_{w=0}^{N_w} \bigwedge_{i,j}^{N_n} \neg W_{w,i,j} \ quand \ i,j \neq fixed[w].x, fixed[w].y$$

Et l'inverse

$$\bigwedge_{w=0}^{N_w} \bigwedge_{i,j}^{N_n} W_{w,i,j} \ quand \ i,j = fixed[w].x, fixed[w].y$$

Il faut vérifier que les voitures ne soit pas superposées a un mur en configuration initiale.

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} V_{v,t,i,j} \implies \neg W_{w,i',j'}$$

$$\bigwedge_{t=0}^{N_t} \bigwedge_{v,v'=0}^{N_v} \bigwedge_{i,j,i',j'=0}^{N_n} \neg V_{v,t,i,j} \vee \neg W_{w,i',j'}$$

Où $((i' < i \vee i' > i + v.largeur - 1) \wedge (j' < j \vee j' > j + v.hauteur - 1)$

Il faut ensuite vérifier qu'un déplacement est valide (voir clause 2.f) en prenant en compte les murs également, un déplacement ne peut pas être fait vers un mur. Si on repart de notre formule vérifiant qu'il n'y a pas de collision vers le haut.

- Pas de collision vers le haut, il ne faut pas de voiture verticale au dessus (en colonne i commencant en j-o avec o qui doit être égal a v'.hauteur) ni de voiture horizontale sur la case du dessus de la voiture (donc qui peut commencer en i-r sur la ligne j-1 avec r qui doit être plus petit ou égal à v'.largeur-1). On ajoute maintenant qu'il n'y ai également pas de mur au dessus (i,j-1).

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (M_{v,t,Up} \wedge V_{v,t,i,j}) \implies (\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j-1}) \wedge (\bigwedge_{o=0}^{N_n} \neg V_{v',t,i,j-o}) \wedge (\bigwedge_{w=0}^{N_w} \neg W_{w,i,j-1})$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v} \bigwedge_{t=0}^{N_t} \bigwedge_{i,j=0}^{N_n} (\neg M_{v,t,Up} \vee \neg V_{v,t,i,j}) \vee ((\bigwedge_{r=0}^{N_n} \neg V_{v',t,i-r,j-1}) \wedge (\bigwedge_{o=0}^{N_n} \neg V_{v',t,i,j-o}) \wedge (\bigwedge_{w=0}^{N_w} \neg W_{w,i,j-1}))$$

$$\equiv$$

$$\bigwedge_{v,v'=0}^{N_v}\bigwedge_{t=0}^{N_t}\bigwedge_{i,j=0}^{N_n}\bigwedge_{r=0}^{N_n}\bigwedge_{o=0}^{N_n}\bigwedge_{w=0}^{N_w}(\neg M_{v,t,Up} \lor \neg V_{v,t,i,j} \lor \neg V_{v',t,i-r,j-1})\land$$
$$(\neg M_{v,t,Up} \lor \neg V_{v,t,i,j} \lor \neg V_{v',t,i,j-o}) \land (\neg M_{v,t,Up} \lor \neg V_{v,t,i,j} \lor \neg W_{w,i,j-1})$$

Avec

- $v \neq v'$
- $v'.largeur - 1 >= r$
- $v'.hauteur == o$
- $i - r$ et $j - o >= 0$

- Donc en suivant le même principe on obtient pour le reste :
  - Collision vers le bas :
$$\bigwedge_{v,v'=0}^{N_v}\bigwedge_{t=0}^{N_t}\bigwedge_{i,j=0}^{N_n}\bigwedge_{r=0}^{N_n}\bigwedge_{w=0}^{N_w}(\neg M_{v,t,Down} \lor \neg V_{v,t,i,j} \lor \neg V_{v',t,i-r,j+v.hauteur})\land$$
$$(\neg M_{v,t,Down} \lor \neg V_{v,t,i,j} \lor \neg W_{w,i,j+v.hauteur})$$
  - Collision vers la droite :
$$\bigwedge_{v,v'=0}^{N_v}\bigwedge_{t=0}^{N_t}\bigwedge_{i,j=0}^{N_n}\bigwedge_{r=0}^{N_n}\bigwedge_{w=0}^{N_w}(\neg M_{v,t,Right} \lor \neg V_{v,t,i,j} \lor \neg V_{v',t,i+v.largeur,j-r})\land$$
$$(\neg M_{v,t,Right} \lor \neg V_{v,t,i,j} \lor \neg W_{w,i+v.largeur,j})$$
  - Collision vers le bas :
$$\bigwedge_{v,v'=0}^{N_v}\bigwedge_{t=0}^{N_t}\bigwedge_{i,j=0}^{N_n}\bigwedge_{r=0}^{N_n}\bigwedge_{o=0}^{N_n}\bigwedge_{w=0}^{N_w}(\neg M_{v,t,Left} \lor \neg V_{v,t,i,j} \lor \neg V_{v',t,i-r,j})\land$$
$$(\neg M_{v,t,Left} \lor \neg V_{v,t,i,j} \lor \neg V_{v',t,i-1,j-o}) \land (\neg M_{v,t,Left} \lor \neg V_{v,t,i,j} \lor \neg W_{w,i-1,j})$$

5 Question 5

5.1 Énoncé

Etant donné une instance de départ, adaptez votre code afin de générer les K meilleures solutions (en nombre de déplacements). Expliquez la modification dans votre rapport. Cette option sera activée par le flag '-k NOMBRE' qui indiquera combien de solutions doivent être trouvées.

5.2 Explications

Une fois qu'on a trouvé le premier K, on ajoute comme clause que au moins un des mouvements effectués doit être différent que la (les) solution(s) précédente(s). Et pour les suivantes, on ajoute au fur et a mesure les solutions effectuées comme clause.

6 Manquant

Pour améliorer ce projet, si nous avions pu faire tout ce que nous voulions, nous aurions fait :

- Une recherche dichotomique pour trouver le meilleur nombre de tour pour réaliser un problème du rush hour
- Corrigé la clause initiale de superposition
- Codé la question 5

7 Projet

Ce projet fut très intéressant, bien que ça ait l'air simple au premier coup d'oeil, nous avons eu beaucoup de difficulté à en arriver où nous en somme arrivé.