

INFO - F - 302 Informatique Fondamentale

Projet : Gestion des horaires de trains
3^{ème} année de bachelier en science informatiques

Année académique 2017-2018

RAPPORT

Auteurs :

Libert Maxime (Matricule : 000397112)
Jacobs Alexandre (Matricule : 000408850)

Table des matières

1	Introductions	1
2	Modélisation	1
2.1	Fonctions booléennes	1
2.2	Variables créées pour l'utilisation du solver-SAT	2
2.3	Contraintes modélisées pour l'utilisation du solver-SAT	2
2.3.1	Explicites	2
2.3.2	Implicites	3
3	Réponses aux Questions	5
3.1	Question 1 : si on remplace la contrainte 1, modélisez	5
3.2	Question 2 : Proposez deux améliorations rendant le modèle plus réaliste	5
3.2.1	Amélioration 1 : Fenêtre glissante	5
3.2.2	Amélioration 2 : Empêcher le retard de plus de TIMEWINDOW	5

1 Introductions

Ce projet s'inscrit dans le cadre du cours INFO-F-302 de l'Université Libre de Bruxelles. Son objectif principal est la modélisation et la résolution d'un problème via un SAT-solver, le problème en question était une gestion des horaires de trains.

2 Modélisation

Cette section explique ce qui a été modélisé; quelles sont les variables utilisées, quelles sont les contraintes prises en compte et quelles sont les fonctions booléennes implémentées pour résoudre notre problématique. Pour une compréhension claire de la modélisation des contraintes et variables, voici quelques définitions d'ensemble et de variables :

- $T \equiv$ Ensemble de trains $= \{t_0 \cdots t_n\}$, où n est le nombre total de trains - 1 .
- $S \equiv$ Ensemble de gares $= \{g_0 \cdots g_m\}$ où m est le nombre total de gares - 1.
- $P \equiv$ Ensemble de paires de gares connectées $= \{paire_1 \cdots paire_m\}$ où m est le nombre total de paire de gares connectées - 1.
- $E \equiv$ Ensemble de trains de capacité de gare + 1 $= \{train_1, train_2, \cdot, train_{capacité_g}, t_{capacité_g+1}\}$
- TIMEWAIT : un nombre de minutes correspondant à l'attente minimale d'un train dans une gare qu'il dessert.
- TIMESLOT : un nombre de minutes correspondant à la durée totale du planning à représenter.
- TIMEWINDOW : un nombre de minutes correspondant à la fenêtre horaire des trajets unitravel.
- TIMEDURATION : un nombre de minutes correspondant à la durée maximale d'un trajet unitravel.
- TravelDuration : un nombre de minutes correspondant à la durée d'un trajet entre deux gares (récupéré sur la map). Vaut -1 si les deux gares ne sont pas connectées et vaut 0 si les deux gares sont les mêmes.

2.1 Fonctions booléennes

1. **bool** isConnected(**int** from, **int** to) : Nous permet de déterminer si deux gares sont liées par une voie, elles le sont si le temps récupéré de la map est supérieur à 0.
2. **bool** isSmallStation(**int** node) : Nous permet de déterminer si une gare est de type "Small"
3. **bool** isFastTrain(**int** train) : Nous permet de déterminer si un train est de type "Fast"

2.2 Variables créées pour l'utilisation du solver-SAT

1. G qui représente la présence d'un train dans une gare à un instant précis

$$\begin{aligned} &\forall \text{ train} \in T, \\ &\forall \text{ gare} \in S, \forall 0 \leq \text{instant} \leq \text{TIMESLOT} \\ &G_{\text{train}, \text{instant}, \text{gare}} = \begin{cases} 1 & \text{si train en gare} \\ 0 & \text{si sinon} \end{cases} \end{aligned}$$

2. V qui représente la présence d'un train sur une voie entre deux gares à un instant précis.

$$\begin{aligned} &\forall \text{ train} \in T, \forall \text{ paire de gares} \in P, \forall 0 \leq \text{instant} \leq \text{TIMESLOT} \\ &V_{\text{train}, \text{paire de gares liée directement par une voie}, \text{instant}} = \begin{cases} 1 & \text{si train sur la voie entre les deux gares} \\ 0 & \text{si sinon} \end{cases} \end{aligned}$$

3. J qui représente le voyage effectué par un train entre deux gares. Il permet de connaître le voyage d'un train d'une gare A vers une gare B ainsi que l'instant de départ et d'arriver.

$$\begin{aligned} &\forall \text{ train} \in T, \\ &\forall \text{ gare}_1, \text{gare}_2 \in S, \forall 0 \leq \text{instant}_1, \text{instant}_2 \leq \text{TIMESLOT} \\ &J_{\text{train}, \text{instant}_1, \text{instant}_2, \text{gare}_1, \text{gare}_2} = \begin{cases} 1 & \text{si train en voyage entre les deux gares} \\ 0 & \text{si sinon} \end{cases} \end{aligned}$$

2.3 Contraintes modélisées pour l'utilisation du solver-SAT

Nous avons implémenté toutes les contraintes qui étaient explicitement décrites dans l'énoncé du projet ainsi que des contraintes implicites.

2.3.1 Explicites

1. Contrainte 1 : Pour toute paire de gare (g,g'), pour chaque fenêtre horaire de durée TimeWindow $[i_{start}, i_{end}]$, il existe un train qui dessert g dans cette fenêtre, puis, plus tard, ce même train desservira g', après une durée de trajet d'au plus TravelDuration.

$$\bigwedge_{g \neq g' \in S} \bigwedge_{[i_{start}, i_{end}]} \bigvee_{i \in [i_{start}, i_{end}]} \bigvee_{i < i' \leq \min[\text{TIMESLOT}, i + \text{TIMEDURATION}]} \bigvee_{\text{train} \in T} J_{\text{train}, i, i', g, g'}$$

2. Contrainte 2 : Pour prévenir tout risque de collision, une voie de chemin de fer entre deux gares, appelée segment, peut accueillir au plus un train.

$$\bigwedge_{\text{train}_1 \neq \text{train}_2 \in T} \bigwedge_{0 \leq i \leq \text{TIMESLOT}} \bigwedge_{(g, g') \in P, g \neq g'} \left[\neg V_{\text{train}_1, i, (g, g')} \vee \neg V_{\text{train}_2, i, (g, g')} \right]$$

3. Contrainte 3 : Les trains de type slow sont des omnibus : ils sont tenus de s'arrêter dans toutes les gares. Les trains de type fast ne sont tenus de s'arrêter que dans les gares de type big.

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i < TIMESLOT} \bigwedge_{(g_1, g_2) \neq (g_2, g_3) \in P} \left[(\neg V_{t,i,(g_1,g_2)} \vee \neg V_{train,i+1,(g_2,g_3)} \vee Rapide(t)) \wedge (\neg V_{t,i,(g_1,g_2)} \vee \neg V_{train,i+1,(g_2,g_3)} \vee Petite(g_2)) \right]$$

4. Contrainte 4 : Afin de permettre aux usagers de monter à bord, tout train desservant une gare doit le faire durant une durée d'au moins TimeWait.

$$\bigwedge_{train \in T} \bigwedge_{0 < i \leq TIMESLOT} \bigwedge_{g \in S} \bigwedge_{i+1 \leq i_2 < i+TIMEWAIT} \left[(G_{train,i-1,g} \vee \neg G_{train,i,g} \vee G_{train,i_2,g}) \right]$$

5. Contrainte 5 : Les trains doivent respecter les temps de trajet prévus par l'infrastructure entre deux gares. On supposera constant, les temps de trajet entre deux gares, ils seront donc donnés en entrée du problème.

$$\bigwedge_{train \in T} \bigwedge_{0 < i \leq TIMESLOT} \bigwedge_{(g_1, g_2) \in P} \bigwedge_{i+1 \leq i_2 < i+TravelDuration} \left[V_{train,i-1,(g_1,g_2)} \vee \neg V_{train,i,(g_1,g_2)} \vee V_{train,i_2,(g_1,g_2)} \right]$$

6. Contrainte 6 : Une gare ne peut accueillir plus de trains que ce que sa capacité le permet.

$$\bigwedge_{0 \leq i \leq TIMESLOT} \bigwedge_{g \in S} \bigwedge_E \bigvee_{train \in E} \neg G_{train,i,g}$$

2.3.2 Implicites

1. Contrainte 1 : Tout train doit être quelque part.

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i \leq TIMESLOT} \left[\bigvee_{g \in S} (G_{train,i,g}) \vee \bigvee_{(g_1, g_2) \in P} (V_{train,i,(g_1, g_2)}) \right]$$

2. Contrainte 2 : Un train ne peut pas être dans 2 gares différentes au même moment.

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i \leq TIMESLOT} \bigwedge_{g_1 \neq g_2 \in G} \left[\neg G_{train,i,g_1} \vee \neg G_{train,i,g_2} \right]$$

3. Contrainte 3 : Un train ne peut pas être sur 2 segments différents au même moment.

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i \leq TIMESLOT} \bigwedge_{(g_1, g_2), (g_3, g_4) \in P, g_1 \neq g_3, g_2 \neq g_4} \left[\neg V_{train,i,(g_1, g_2)} \vee \neg V_{train,i,(g_3, g_4)} \right]$$

4. Contrainte 4 : Un train ne peut pas être sur un segment et une gare au même moment.

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i \leq TIMESLOT} \bigwedge_{g_1 \in S} \bigwedge_{(g_2, g_3) \in P} [\neg G_{train, i, g_1} \vee \neg V_{train, i, (g_2, g_3)}]$$

5. Contrainte 5 : Lorsqu'un train sort d'un segment A-B, soit il arrive à gare B soit on est sur un segment B-C.

$$\bigwedge_{train \in T} \bigwedge_{0 < i \leq TIMESLOT} \bigwedge_{(g_1, g_2) \in P} [(\neg V_{train, i-1, (g_1, g_2)} \vee V_{train, i, (g_1, g_2)}) \vee (G_{train, i, g_2} \vee \bigvee_{g_3 \in S, g_3 \text{ liée à } g_2} V_{train, i, (g_2, g_3)})]$$

6. Contrainte 6 : Lorsqu'un train sort d'une gare A, il doit être sur un segment qui part de A.

$$\bigwedge_{train \in T} \bigwedge_{0 < i \leq TIMESLOT} \bigwedge_{g_1 \in S} [(\neg G_{train, i-1, g_1} \vee G_{train, i, g_1} \vee (\bigvee_{g_2 \in S, g_1 \text{ liée à } g_2} V_{train, i, (g_1, g_2)}))]$$

7. Contrainte 7 : Contraintes pour la cohérence de la variable J (partie 1).

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i \leq TIMESLOT} \bigwedge_{i < i' \leq TIMESLOT} \bigwedge_{g \neq g' \in S} (\neg J_{train, i, i', g, g'} \vee G_{train, i, g}) \wedge (\neg J_{train, i, i', g, g'} \vee G_{train, i', g'})$$

8. Contrainte 8 : Contraintes pour la cohérence de la variable J (partie 2).

$$\bigwedge_{train \in T} \bigwedge_{0 \leq i \leq TIMESLOT} \bigwedge_{i < i' \leq TIMESLOT} \bigwedge_{g \neq g' \in S} \neg G_{train, i, g} \vee \neg G_{train, i', g'} \vee J_{train, i, i', g, g'}$$

3 Réponses aux Questions

3.1 Question 1 : si on remplace la contrainte 1, modélisez

Si on remplace la contrainte 1 par la contrainte suivante : Pour toute paire de gare (g,g'), pour chaque fenêtre horaire de durée TimeWindow, il est possible de prendre un train dans la fenêtre horaire, en gare g, pour aller vers g' après une durée maximale de trajet TravelDuration, le tout en changeant potentiellement de train au plus NbChange fois. La version précédente du problème correspond au cas NbChange=0.

$$\bigwedge_{c \in [0, \text{NbChange}]} \bigvee \left(\bigwedge_{g \neq g' \in S} \bigwedge_{[i_{\text{start}}, i_{\text{end}}]} \bigvee_{i \in [i_{\text{start}}, i_{\text{end}}]} \bigvee_{i < i' \leq \min[\text{TIMESLOT}, i + \text{TIMEDURATION}]} \bigvee_{\text{train} \in T} G_{\text{train}, i, g} \right. \\ \left. \bigvee_{\substack{i < i_{\text{arrivé}_1} < \dots < i_{\text{arrivé}_c} < i' \\ i < i_{\text{départ}_1} < \dots < i_{\text{départ}_c} < i'}} \bigvee_{\text{train}_1, \dots, \text{train}_c \in T} \bigvee_{g_1, \dots, g_c \in S} \bigwedge_{\substack{k \in [1, c-1] \\ \text{train}_{k-1} \neq t_k \\ i_{\text{arrivé}_k} < i_{\text{départ}_k}}} G_{\text{train}_{k-1}, i_{\text{arrivé}_k}, g_k} \wedge G_{\text{train}_k, i_{\text{départ}_k}, g_k} \vee G_{\text{train}_k, i', g'} \right)$$

3.2 Question 2 : Proposez deux améliorations rendant le modèle plus réaliste

3.2.1 Amélioration 1 : Fenêtre glissante

Cette amélioration permet d'être plus réaliste. La modélisation en FNC est simple car pour la mettre en place, il "suffit" de changer la façon dont la contrainte 1 est implémentée (cf : code en annexe (main.cpp)) dans la troisième boucle for, on incrémente iStart de 1 à chaque tour de boucle et non plus de TIMEWINDOW.

$$\bigwedge_{g \neq g' \in S} \bigwedge_{[i_{\text{start}}, i_{\text{end}}]} \bigvee_{i \in [i_{\text{start}}, i_{\text{end}}]} \bigvee_{i < i' \leq \min[\text{TIMESLOT}, i + \text{TIMEDURATION}]} \bigvee_{\text{train} \in T} J_{\text{train}, i, i', g, g'}$$

3.2.2 Amélioration 2 : Empêcher le retard de plus de TIMEWINDOW

Cette amélioration permet d'être plus réaliste dans le sens où si un train arrive avec TIMEWINDOW de retard, l'utilisateur va prendre le train prévu pour la prochaine TIMEWINDOW. Par exemple, si on attend le train Ottignies-Arlon (qui passe normalement une fois par heure) à 13h54 et que le train à une heure de retard, on va prendre le train de 14h54 étant donné que la SNCB va annuler le train que nous voulions prendre au départ. Pour mettre en place cette amélioration, il faut modifier un petit peu la contrainte 5 et rajouter le fait que on ne peut plus être sur la voie après un temps égal à TIMEWINDOW.

$$\bigwedge_{\text{train} \in T} \bigwedge_{0 < i \leq \text{TIMESLOT}} \bigwedge_{(g_1, g_2) \in P} \bigwedge_{i+1 \leq i_2 < i + \text{TravelDuration} + \text{TIMEWINDOW} - 1} \left[V_{\text{train}, i-1, (g_1, g_2)} \vee \neg V_{\text{train}, i, (g_1, g_2)} \vee V_{\text{train}, i_2, (g_1, g_2)} \right] \\ \wedge G_{\text{train}, i_1 + \text{TravelDuration} + \text{TIMEWINDOW}, g_2}$$