Projet INFO-F-302 : Gestion des horaires de train

MAHIA Jérôme, BONAERT Gregory

May 18, 2018

1 Sémantique des variables booléennes

- sur_voie

  La variable booléenne sur_voie

- $dans\_gare$

  La variable booléenne $dans\_gare$

- $fait\_trajet$

  La variable booléenne $fait\_trajet$

2 Formules des contraintes

- i est un instant, qui respecte toujours $0 \leq i < TimeSlots$
- g est une gare
- t est un train
- c_g est la capacité de la gare g

2.1 Contraines explicites

- Pour toute paire de gare (g, g'), pour chaque fenêtre horaire de durée TimeWindow, 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.

$$\bigwedge_{(g,g'),g\neq g'} \bigwedge_{TimeWindow[i_s,i_e]} \bigvee_{i\in[i_s,i_e]} \bigvee_{i<i'\leq min(TimeSlots-1,i+TravelDuration)} \bigvee_{t} [T_{t,i,g,i',g'}]$$

  Les Time Windows utilisées sont non-glissantes (elles ne se superposent pas).

- Pour prévenir tout risque de collision, une voie de chemin de fer entre deux gares, appelé segment, peut accueillir au plus un train.

$$\bigwedge_{t_1\neq t_2} \bigwedge_{i} \bigwedge_{(g,g'),g\neq g'} [\neg V_{t_1,i,(g,g')} \vee \neg V_{t_2,i,(g,g')}]$$

- 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.

$$\bigwedge_{t} \bigwedge_{i} \bigwedge_{(g_1,g_2),(g_2,g_3)} [\neg V_{t,i,(g_1,g_2)} \vee \neg V_{t,i+1,(g_2,g_3)} \vee (Rapide(t) \wedge Petite(g_2)]$$

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

$$\bigwedge_{t,g,i>0} \bigwedge_{i+1 \leq i_2 < i+TWait} \left[\left(G_{g,t,i-1} \vee \neg G_{g,t,i} \vee G_{g,t,i_2} \right) \right.$$

- 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.

$$\bigwedge_{(g_1,g_2),t,i>0} \bigwedge_{i+1<=i_2<i+TravelDuration} \left[V((g_1,g_2),t,i-1) \vee \neg V((g_1,g_2),t,i) \vee V((g_1,g_2),t,i_2) \right]$$

- Une gare ne peut accueillir plus de trains que ce que sa capacité le permet.

$$\bigwedge_{g,i} \bigwedge_{t_1,t_2,\cdots,t_{cg},t_{c_g+1}} \bigvee_{1 \leq j \leq c_g+1} \neg G_{t_j,g,i}$$

2.2 Contraintes implicites

- Tout train doit être quelque part

$$\bigwedge_{t,i} \left[\bigvee_{g} \left(G_{g,i,t} \right) \vee \bigvee_{(g_1,g_2)} \left(V_{(g_1,g_2)i,t} \right) \right]$$

- Un train ne peut pas être dans 2 gares différentes au même moment

$$\bigwedge_{g_1,g_2,i,t,g_1 \neq g_2} \left[\neg G_{g_1,i,t} \vee \neg G_{g_2,i,t} \right]$$

- Un train ne peut pas être sur 2 segments différents au même moment

$$\bigwedge_{(g_1,g_2),(g_3,g_4),i,t,(g_1,g_2) \neq (g_3,g_4)} \left[\neg V_{(g_1,g_2),i,t} \vee \neg V_{(g_3,g_4),i,t} \right]$$

- Un train ne peut pas être sur un segment et une gare au même moment

$$\bigwedge_{g_1,(g_2,g_3),i,t} \left[\neg G_{g_1,i,t} \vee \neg V_{(g_2,g_3),i,t} \right]$$

- Lorsqu'on sort d'un segment A-B, soit on arrive à gare B soit on est sur un segment B-C

$$\bigwedge_{(g_1,g_2),t,i>0} \left[\left(\neg V_{(g_1,g_2),i-1,t} \vee V_{(g_1,g_2),i,t} \right) \vee \left(G_{g_2,t,i} \vee \bigvee_{g_3,\ g_3 \text{ liée à } g_2} \left(V_{(g_2,g_3),i,t} \right) \right) \right]$$

- Lorsqu'un train sort d'une gare A, il doit être sur un segment qui part de A

$$\bigwedge_{g_1,t,i>0} \left[\left(\neg G_{g_1,i-1,t} \vee G_{(g_1,i,t)} \right) \vee \left(\bigvee_{g_2,\ g_1 \text{ liée à } g_2} V_{(g_1,g_2),i,t} \right) \right]$$

- Contraintes supplémentaires afin de rendre la variable booléenne $fait\_trajet$ cohérente

  Première partie:

$$\bigwedge_{t,g,g',i,i',g\neq g',i<i'} (\neg T_{t,i,g,i',g'} \vee G_{g,i,t}) \wedge (\neg T_{t,i,g,i',g'} \vee G_{g',i',t})$$

  Deuxième partie:

$$\bigwedge_{t,g,g',i,i',g\neq g',i<i'} \neg G_{g,i,t} \vee \neg G_{(g',i',t} \vee T_{t,i,g,i',g'}$$

3 Questions supplémentaires

3.1 Remplacement de la première contrainte

Remplaçons la première contrainte par:

> 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 $NbChange = 0$.

Pour modéliser cette contrainte, nous avons besoin de trouver un moyen d'exprimer la notion "il y a un trajet partant de A à l'instant i et arrivant en B à l'instant j en utilisant k trains". Voici comment le modéliser:

$$Chemin_k(g_0, i_{0,D}, g_k, i_{k,A}) \equiv \bigvee_{t_0,t_1,\ldots,t_{k-1}} \bigvee_{g_1,g_2,\ldots,g_{k-1}} \bigvee_{i_{0,D}<i_{1,A}<i_{1,D}<\ldots<i_{k-1,A}<i_{k-1,D}<i_{k,A}} \bigwedge_{0<j<k-1} T_{t_j,g_j,i_{j,D},g_{j+1},i_{j+1,A}}$$

Dans cette expression $i_{g,A}$ représente l'instant d'arrivée à la gare g et $i_{g,A}$ l'instant de départ de la gare g. g_0 est la gare de départ et g_k est la gare d'arrivée.

Une fois qu'on a cette notion, il est assez facile d'écrire la nouvelle contrainte:

$$\bigwedge_{(g,g'),g\neq g'} \bigwedge_{TimeWindow[i_s,i_e]} \bigvee_{i\in[i_s,i_e]} \bigvee_{i<i'\leq min(TimeSlots-1,i+TravelDuration)} \bigvee_{1\leq k\leq 1+NbChange} Chemin_k(g, i, g', i')$$

Note: on utilise $Chemin_k$ plutôt que son expression équivalente par clarté.

3.2 Proposer deux améliorations rendant le modèle plus réaliste.

3.2.1 Faire que les time windows puissent se superposer

Si on permet aux time windows de se superposer (fenêtres glissantes), cela permet d'assurer un traffic plus fluide et efficace.

Pour modéliser cette amélioration, il faut modifier la contrainte 1 de façon à ce que les time windows utilisées puissent se superposer. Voici la modélisation:

$$\bigwedge_{(g,g'),g\neq g'} \bigwedge_{TimeWindow[i_s,i_e]} \bigvee_{i\in[i_s,i_e]} \bigvee_{i<i'\leq min(TimeSlots-1,i+TravelDuration)} \bigvee_{t} [T_{t,i,g,i',g'}]$$

Les TimeWindows acceptées sont toutes les timeWindows tel que

$$0 \leq i_s < i_e = i + TimeWindow < TimeSlots$$

3.2.2 Limiter la quantité de retard admise ou autoriser le retard

Par défaut on autorise un retard illimité sur les voies, ce qui n'est pas très réaliste. Pour améliorer cela, on autorise au maximum 100% de retard. Par exemple, si il faut 20 minutes pour parcourrir une voie, on autorise au maximum un temps de trajet de 40 minutes.

De manière plus générale, si la voie a comme temps de trajet minimum TravelDuration, le train peut rester au maximum 2 x TravelDuration sur la voie. Voici la contrainte qui modélise cette amélioration:

$$\bigwedge_{(g_1,g_2),i>0,t} \bigwedge_{i+1\leq i_2<i+TravelDuration} (\neg V_{g_1,g_2,t,i-1} \wedge V_{g_1,g_2,t,i}) \implies \neg(\bigwedge_{i+1<i'\leq i+2*TravelDuration} V_{g_1,g_2,t,i'})$$

Ceci est équivalent (en forme normale conjonctive) à

$$\bigwedge_{(g_1,g_2),i>0,t} \bigwedge_{i+1\leq i_2<i+TravelDuration} (V_{g_1,g_2,t,i-1} \vee \neg V_{g_1,g_2,t,i}) \vee (\bigvee_{i+1<i'\leq i+2*TravelDuration} \neg V_{g_1,g_2,t,i'})$$

4 Bonus