Un algorithme de génération de colonnes pour le problème de capacité d'infrastructure ferroviaire

Aurélien Merel[1,2], Sophie Demassey[2], Xavier Gandibleux[1]

Laboratoire d'Informatique de Nantes-Atlantique

[1] Université de Nantes
2 rue de la Houssinière
Nantes, France

[2] École des Mines de Nantes
4 rue Alfred Kastler
Nantes, France

25 février 2010

ROADEF 2010 – Toulouse

1. Problème traité
   - Description du problème de saturation
   - Infrastructure et circulation
   - Modélisation PLNE
2. Résolution
   - Génération de colonnes
   - Filtrage des contraintes
   - Rééquilibrage des variables duales
3. Bilan et perspectives
   - Bilan
   - Perspectives

Problème de saturation

Définition (Delorme, 2003)

Objectif : insérer le **maximum de circulation** issue d'une liste saturante dans une infrastructure, en plus d'une grille pré-existante $\rightarrow$ **marge de capacité** disponible.

Problème traité

Grille pré-existante vide $\rightarrow$ **capacité absolue**.

Infrastructure et circulation

RECIFE

- Infrastructure : Pierrefitte-Gonesse
- Pour chaque train de la liste saturante :
  - Itinéraires possibles et vitesse prédéfinis
  - Heure de sortie nominale + ensemble de retards autorisés, toutes les 15 ou 30s jusqu'au max

$\Rightarrow$ Temps de passages sur un itinéraire **connus relativement à l'heure d'entrée**

Évolution par rapport à RECIFE

Retards à la seconde près $\Rightarrow$ davantage de circulations possibles.

Saturation par affectation d'un **itinéraire** et d'un **retard** (heure d'entrée absolue) à un maximum de trains

Modélisation linéaire : hypothèses et aperçu

Hypothèses

- Infrastructure discrétisée (**zones** de détection physiques z)
- Temps discrétisé (**périodes** p)
- (z, p) = une **ressource unaire** m (un train au plus peut occuper chaque zone à chaque instant)
- Une **route** : ensemble de ressources spatio-temporelles

Modèle linéaire

- Set Packing Problem
- Différent de RECIFE, développé dans Lusby et al. 2006 et Merel et al. 2009
- Résolution de la relaxation linéaire

Modélisation linéaire : Set Packing Problem

Lignes (contraintes)

- **unicité** : chaque train physique ne peut être routé qu'une seule fois
- **ressources** : une contrainte par ressource m

Colonnes (variables)

- Une variable binaire $\Leftrightarrow$ un couple (train, route)
  $\rightarrow$ utilisation d'une contrainte d'unicité et plusieurs contraintes de ressources

Objectif

- Maximiser le nombre de trains routés

Modèle linéaire : exemple

		(t_1, r_1)	(t_1, r_2)	(t_2, r_3)	(t_2, r_4)
unicité	t_1	1	1		
	t_2			1	1
ressources	(z_1, p_1)	1			
	(z_1, p_2)	1	1		
	$\vdots$		1		
	(z_2, p_1)			1	1
	(z_2, p_2)			1	1
	$\vdots$				1

$\cdot x \leq \mathbf{1}$

Résolution par génération de colonnes

Motivations

- Beaucoup de routes possibles pour chaque train
- Nombreux conflits entre les routes
- $\Rightarrow$ Combinatoire importante et solution constituée d'un faible sous-ensemble de colonnes

Principe

1. Résolution du problème maître réduit (PMR) $\rightarrow$ solutions optimales primale et duale
2. Résolution du **sous-problème** : ajout de colonnes valides dans le PMR
3. Réitérer tant que de nouvelles colonnes sont trouvées

Génération de colonnes : sous-problème

Critères de génération d'une colonne

Couple (t, r) valide ; **coût réduit** $k_{t,r}$ strictement positif

Stratégies d'ajout de colonnes

- Meilleures colonnes au sens du coût réduit
- La meilleure colonne pour chaque train
- ...

Calcul des coûts réduits

- Coût réduit $k_{t,r}$ donné par les valeurs duales associées aux contraintes concernées par le train t utilisant la route r
- Processus itératif $\Rightarrow$ réduction de la complexité

Génération de colonnes : exemple

Variables duales et coût réduit

- $v_t \Leftrightarrow$ contraintes d'unicité
- $w_m,\ m = (z, p) \Leftrightarrow$ contraintes de ressources
- Coût réduit $k_{t,r} \ = \ 1 - v_t^* - \sum_{m \in r} w_m^*$

Exemple de solution au dual :

v^*

t_1	t_2
$0,1$	$0,8$

w^*

	p_1	p_2	p_3	p_4	p_5	p_6
z_1	$0,2$	$0,8$	$0,1$	0	0	0
z_2	$0,9$	$0,7$	$0,1$	$0,1$	0	0
z_3	$0,2$	$0,2$	0	0	0	0

$\rightarrow$ **Trouver** (t, r) **valide tel que** $k_{t,r} > 0$

Génération de colonnes : exemple

Train t_1 :

- peut emprunter l'itinéraire (z_1, z_2, z_3)
- entrée nominale sur z_1 et z_2 à p_1, sur z_3 une période plus tard
- z_1 utilisée 2 périodes, z_2 et z_3 3 périodes

v^*

t_1	t_2
$0,1$	$0,8$

w^*

	p_1	p_2	p_3	p_4	p_5	p_6
z_1	$0,2$	$0,8$	$0,1$	0	0	0
z_2	$0,9$	$0,7$	$0,1$	$0,1$	0	0
z_3	$0,2$	$0,2$	0	0	0	0

$$1 - v_t^* - \sum_{m \in r} w_m^*$$

Génération de colonnes : exemple

Train t_1 :

- peut emprunter l'itinéraire (z_1, z_2, z_3)
- entrée nominale sur z_1 et z_2 à p_1, sur z_3 une période plus tard
- z_1 utilisée 2 périodes, z_2 et z_3 3 périodes

v^*

t_1	t_2
$0,1$	$0,8$

w^*

	p_1	p_2	p_3	p_4	p_5	p_6
z_1	$0,2$	$0,8$	$0,1$	0	0	0
z_2	$0,9$	$0,7$	$0,1$	$0,1$	0	0
z_3	$0,2$	$0,2$	0	0	0	0

$1 - v_t^* - \sum_{m \in r} w_m^* = \quad -2 \quad < 0 \quad \rightarrow$ itérer sur le retard

Génération de colonnes : exemple

Train t_1 : (**retard** : 1 période)

- peut emprunter l'itinéraire (z_1, z_2, z_3)
- entrée nominale sur z_1 et z_2 à p_1, sur z_3 une période plus tard
- z_1 utilisée 2 périodes, z_2 et z_3 3 périodes

v^*

t_1	t_2
$0,1$	$0,8$

w^*

	p_1	p_2	p_3	p_4	p_5	p_6
z_1	$0,2$	$0,8$	$0,1$	0	0	0
z_2	$0,9$	$0,7$	$0,1$	$0,1$	0	0
z_3	$0,2$	$0,2$	0	0	0	0

$1 - v_t^* - \sum_{m \in r} w_m^* = \quad -0,9 \quad < 0 \quad \rightarrow$ itérer sur le retard

Génération de colonnes : exemple

Train t_1 : (**retard** : 2 périodes)

- peut emprunter l'itinéraire (z_1, z_2, z_3)
- entrée nominale sur z_1 et z_2 à p_1, sur z_3 une période plus tard
- z_1 utilisée 2 périodes, z_2 et z_3 3 périodes

v^*

t_1	t_2
$0,1$	$0,8$

w^*

	p_1	p_2	p_3	p_4	p_5	p_6
z_1	$0,2$	$0,8$	$0,1$	0	0	0
z_2	$0,9$	$0,7$	$0,1$	$0,1$	0	0
z_3	$0,2$	$0,2$	0	0	0	0

$1 - v_t^* - \sum_{m \in r} w_m^* = \quad 0,6 \quad > 0 \quad \rightarrow t$ et r déterminés

Caractéristiques des tests

Instance

- 66 trains, 3-6 itinéraires par train, 30s de retard maximum
- 10943 colonnes, 350000 lignes

Stratégies d'ajout de colonnes

- 5 stratégies testées

Matériel et logiciel

- CPU Intel Core2 Duo T9500 2,60GHz, 2Go RAM
- Coin-OR Clp 1.09

Résultats

Stratégies d'ajout

	1	2	3	4	**5**
Itérations	50	533	43	50	**39**
Temps (s)	16220	44766	11761	15152	**10188**
Colonnes	2196	2176	2291	2161	**2118**
Lignes	116529	116755	117236	116470	116839

Résultats

Stratégies d'ajout

	1	2	3	4	**5**
Itérations	50	533	43	50	**39**
Temps (s)	16220	44766	11761	15152	**10188**
Colonnes	2196	2176	2291	2161	**2118**
Lignes	116529	116755	117236	116470	116839

- Entre 19% et 21% du nombre maximum de colonnes
- Impact significatif de la stratégie
- Corrélation nombre d'itérations $\leftrightarrow$ temps
- Meilleur temps moyen d'une itération : 261 secondes
  $\rightarrow$ Comment réduire ce temps ?

Filtrage des contraintes (Merel et al. 2009)

Motivations

- Réduire le temps de résolution du PMR
- Redondances dans l'utilisation des ressources

Principe

- Suppression des contraintes redondantes/dominées
- Formulation linéaire équivalente
  $\Rightarrow$ solution optimale pour la formulation complète

Intégration à la génération de colonnes

- Exécution à chaque itération
- Contrainte supprimée $\Rightarrow$ variable duale nulle

Exemples de redondance et dominance

		(t_1, r_1)	(t_1, r_2)	(t_2, r_3)	(t_2, r_4)	
unicité	t_1	1	1			
	t_2			1	1	
	(z_1, p_1)	1				
	(z_1, p_2)	1	1			$\cdot x \leq \mathbf{1}$
ressources	$\vdots$		1			
	(z_2, p_1)			1	1	
	(z_2, p_2)			1	1	
	$\vdots$				1	

Filtrage : résultats

	Sans	Avec
Itérations	**39**	49
Temps (s)	10188	**5199**
Colonnes	**2118**	2584
Lignes	116839	**14008**

- Temps de filtrage : 516s (10% du total)
- Efficacité décroissante (de 99% à 12%)

Filtrage : résultats

	Sans	Avec
Itérations	**39**	49
Temps (s)	10188	**5199**
Colonnes	**2118**	2584
Lignes	116839	**14008**

- Temps de filtrage : 516s (10% du total)
- Efficacité décroissante (de 99% à 12%)

- 88% de contraintes en moins à la dernière itération
- Temps réduit de 49%
- 106s par itération (-59%)
- Détérioration de la qualité des colonnes générées
  $\rightarrow$ Comment améliorer cette qualité ?

Rééquilibrage des variables duales

Motivations

- Améliorer la qualité des colonnes générées pour réduire le nombre d'itérations

Principe

- Modification de la solution duale en conservant validité et optimalité
- Amélioration de la corrélation entre les w_m et la saturation des ressources après résolution du PMR

Rééquilibrage des variables duales

Motivations

- Améliorer la qualité des colonnes générées pour réduire le nombre d'itérations

Principe

- Modification de la solution duale en conservant validité et optimalité
- Amélioration de la corrélation entre les w_m et la saturation des ressources après résolution du PMR

$$1 - v_t^* - \sum_{m \in r} w_m^*$$

Exemple de rééquilibrage

1. $m_1 = (z_1, p_1),\ m_2 = (z_1, p_2),\ m_3 = (z_1, p_3)$ redondantes
   $\Rightarrow$ Contraintes m_2 et m_3 supprimées par filtrage

Exemple de rééquilibrage

1. $m_1 = (z_1, p_1),\ m_2 = (z_1, p_2),\ m_3 = (z_1, p_3)$ redondantes
   $\Rightarrow$ Contraintes m_2 et m_3 supprimées par filtrage
2. Résolution du PMR, solution duale :

		p_1	p_2	p_3
w^*	z_1	$0,9$	$\mathbf{0}$	$\mathbf{0}$

Exemple de rééquilibrage

1. $m_1 = (z_1, p_1)$, $m_2 = (z_1, p_2)$, $m_3 = (z_1, p_3)$ redondantes
   $\Rightarrow$ Contraintes m_2 et m_3 supprimées par filtrage
2. Résolution du PMR, solution duale :

		p_1	p_2	p_3
w^*	z_1	$0,9$	0	0

3. Rééquilibrage : **répartition** de la valeur duale associée à des contraintes redondantes :

		p_1	p_2	p_3
w^*	z_1	$0,3$	$0,3$	$0,3$

Exemple de rééquilibrage

1. $m_1 = (z_1, p_1),\ m_2 = (z_1, p_2),\ m_3 = (z_1, p_3)$ redondantes
   $\Rightarrow$ Contraintes m_2 et m_3 supprimées par filtrage
2. Résolution du PMR, solution duale :

		p_1	p_2	p_3
w^*	z_1	$0,9$	$\mathbf{0}$	$\mathbf{0}$

3. Rééquilibrage : **répartition** de la valeur duale associée à des contraintes redondantes :

		p_1	p_2	p_3
w^*	z_1	$0,3$	$0,3$	$0,3$

4. Génération de nouvelles colonnes en utilisant la nouvelle solution duale

Résultats du rééquilibrage

	Basique	Filtré	Filtré, rééquilibré	Amélioration	
Itérations	39	49	**32**	18%	35%
Temps (s)	10188	5199	**996**	90%	81%
Colonnes	2118	2584	**1793**	15%	31%
Lignes	116839	14008	10216	91%	27%

Temps de filtrage cumulé : 205s

Résultats du rééquilibrage

	Basique	Filtré	Filtré, rééquilibré	Amélioration	
Itérations	39	49	**32**	18%	35%
Temps (s)	10188	5199	**996**	90%	81%
Colonnes	2118	2584	**1793**	15%	31%
Lignes	116839	14008	10216	91%	27%

Temps de filtrage cumulé : 205s

- 31s par itération (-88% et -71%)
- Temps de filtrage significatif (21%)
- Colonnes de meilleur qualité
  $\Rightarrow$ Le rééquilibrage guide la génération des colonnes en créant une solution duale plus adaptée

Bilan

- Assemblage de plusieurs techniques
  - Génération de colonnes
  - Diverses stratégies d'ajout
  - Suppression de contraintes redondantes
  - Amélioration des colonnes générées par modification de la solution duale
- Résultats significatifs
  - Division par 2 puis par 10 du temps de résolution
  - Meilleure efficacité dans la génération des colonnes
- Verrous restants
  - Taille encore rédhibitoire sur d'autres instances
  - Solution fractionnaire

Perspectives

- Amélioration de l'existant
  - Filtrage : n'effectuer les comparaisons que sur les colonnes ajoutées
- À ajouter
  - Génération de coupes (réduction de la taille en nombre de lignes)
  - Solution entière