

Ordonnement de véhicules dans les usines RENAULT : une approche par recherche locale à voisinage large

B. Estellon¹, F. Gardi^{1,2}, et K. Nouioua¹

¹ Laboratoire d'Informatique Fondamentale,
Parc Scientifique et Technologique de Luminy, Marseille, France

² PROLOGIA - Groupe Air Liquide,
Parc Scientifique et Technologique de Luminy, Marseille, France

{Bertrand.Estellon,Frederic.Gardi,Karim.Nouioua}@lif.univ-mrs.fr

Le Challenge ROADEF 2005 a pour objet l'ordonnement des véhicules dans les chaînes de montage des usines RENAULT (problème plus connu sous le nom de *car sequencing* en anglais). Voici une heuristique naturelle pour résoudre ce problème :

Descente par recherche locale :

début ;

calculer une solution initiale ;

tant que le temps imparti n'est pas écoulé **faire**

 sélectionner k véhicules de la séquence ;

 calculer une nouvelle solution par permutation de ces k véhicules ;

retourner la solution ;

fin ;

Lorsque $k = 2$, la permutation en question correspond à un simple échange. En dépit de son voisinage réduit, ce type de recherche locale permet d'obtenir d'excellents résultats [1]. Toutefois, le choix d'une valeur de k plus grande élargit considérablement l'espace de recherche et donc les chances de trouver une permutation améliorant le coût de la solution courante. Dans cette note, nous indiquons comment déterminer efficacement une permutation optimale (ou presque) de ces k véhicules lorsque k est très grand, c'est-à-dire de l'ordre du tiers voire de la moitié du nombre de véhicules à ordonner.

Le problème du *car sequencing* peut être représenté sous la forme d'un programme linéaire en nombres entiers. En associant à chaque couple véhicule/position une variable binaire, nous obtenons un programme de type *set partitioning* auquel il faut ajouter les variables associées aux pénalités provoquées par les violations de ratios ainsi que les contraintes permettant d'activer celles-ci. Les contraintes portant sur la couleur des véhicules peuvent aussi être traduites, mais de façon moins naturelle. Bien entendu, la résolution exacte d'un tel programme est impossible puisque le nombre de variables binaires est quadratique en le nombre de véhicules à ordonner. De plus, la relaxation linéaire, elle aussi difficile à calculer, est de très mauvaise qualité (solution très fractionnaire de coût quasi nul). Cependant, nous avons pu observer qu'en fixant des véhicules à des positions données (c'est-à-dire en fixant des variables binaires à un), la qualité de la relaxation s'améliore nettement, rendant ainsi abordable la résolution en nombres entiers. En fait, moins il y a de contiguïté entre les véhicules libres de permuter, meilleure est la relaxation linéaire. En choisissant de façon aléatoire les k véhicules à permuter puis en ajoutant à ceux-ci les véhicules provoquant des violations, nous obtenons un programme linéaire en nombres entiers dont la résolution ne prend que quelques secondes à l'aide d'un *branch and bound* classique [2] sur un PC 1.6 MHz et 256 Mo (la résolution de la relaxation linéaire du programme se fait par la méthode du simplexe, accélérée par le passage d'une base réalisable calculée à partir de la solution courante).

Notons enfin qu'une solution initiale de qualité peut être calculée à l'aide de cette même méthode en résolvant (exactement ou presque) le problème pour un échantillon représentatif de l'ensemble des véhicules : les violations provoquées par le raccordement des échantillons optimisés sont facilement réparables). Malgré cela, les résultats obtenus par la recherche locale à voisinage large restent largement inférieurs à ceux obtenus par recherche locale à voisinage réduit [1]. À ce propos, nous

pensons que l'écriture d'un algorithme du simplexe dédié au problème peut permettre une nette amélioration des performances de la méthode (l'opération la plus coûteuse à chaque itération est la résolution de la relaxation linéaire du programme).

Références

1. B. ESTELLON, F. GARDI et K. NOUIOUA (2005). Ordonnancement de véhicules dans les usines RENAULT : une approche par recherche locale à voisinage réduit. In *Actes du 6ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*.
2. A. MAKHORIN (2004). Librairie GLPK (GNU Linear Programming Kit, version 4.4). <http://www.gnu.org/software/glpk/glpk.html>.

Ordonnement de véhicules dans les usines RENAULT : une approche par recherche locale à voisinage réduit

B. Estellon¹, F. Gardi^{1,2}, et K. Nouioua¹

¹ Laboratoire d'Informatique Fondamentale,
Parc Scientifique et Technologique de Luminy, Marseille, France

² PROLOGIA - Groupe Air Liquide,
Parc Scientifique et Technologique de Luminy, Marseille, France

{Bertrand.Estellon,Frederic.Gardi,Karim.Nouioua}@lif.univ-mrs.fr

Le Challenge ROADEF 2005 a pour objet l'ordonnement des véhicules dans les chaînes de montage des usines RENAULT (problème plus connu sous le nom de *car sequencing* en anglais). Par opposition à [1], nous proposons une *méthode de descente par recherche locale à voisinage réduit* pour résoudre le problème. À chaque itération, une transformation de la séquence de véhicules (appelée film) est effectuée : si cette transformation n'augmente pas le coût de la solution, alors celle-ci est validée. Les transformations en question ne portent que sur de petites portions du film et un très petit nombre de véhicules. Par exemple, une des transformations les plus utilisées consiste simplement en l'échange de deux véhicules dans le film. Bien que le voisinage exploré à chaque itération soit très restreint, la répétition de ces transformations en très grand nombre (en l'occurrence, plus d'une dizaine de millions par minute) s'avère très efficace : sur l'ensemble des instances fournies par RENAULT, aucun minimum local n'a jamais été rencontré et la descente semble converger vers un optimum global (les tests ont été effectués sur PC 1.6 MHz et 256 Mo, avec une limite de temps fixée à 10 mn).

Descente par recherche locale :

début ;

calculer une solution initiale ;

tant que le temps imparti n'est pas écoulé **faire**

choisir une transformation à appliquer au film ;

si la transformation respecte les contraintes et n'augmente pas le coût **alors**
effectuer la transformation ;

retourner la solution ;

fin ;

Une bonne solution initiale peut être rapidement obtenue à l'aide d'un algorithme de type glouton [4], lorsque la solution fournie par RENAULT est de mauvaise qualité. Les transformations, dérivées des travaux de Putcha et Gottlieb [2] et Putcha *et al.* [3] sur le sujet, sont au nombre de cinq : échange, insertion avant, insertion arrière, miroir et mélange aléatoire (voir la figure 1). La fréquence de l'emploi de chacune de ces transformations est plus ou moins élevée selon la priorité des objectifs et le temps imparti pour la résolution. De manière générale, le nombre d'échanges doit rester prépondérant (environ 60 % du nombre total de transformations) ; le nombre d'insertions avant, d'insertions arrière et de miroirs est partagé (environ 13 % de chaque) et le nombre de mélanges aléatoires, coûteux en temps mais nécessaire à la convergence, est inférieur à 1 %.

L'efficacité de la méthode repose ensuite sur deux critères : le choix des positions et des véhicules sur lesquels sont appliquées les transformations et la qualité de l'implantation de la procédure d'évaluation de ces dernières. La mise en œuvre de structures de données spécifiques a notamment permis de multiplier par 100 le nombre d'itérations effectuées par minute, passant ainsi de quelques centaines de milliers d'itérations par minute à quelques dizaines de millions. Ce type de recherche locale permet ainsi d'atteindre en quelques secondes les résultats obtenues en plusieurs minutes à l'aide d'une recherche locale au voisinage plus large, comme celle que nous présentons dans [1].

Références

1. B. ESTELLON, F. GARDI et K. NOUIOUA (2005). Ordonnement de véhicules dans les usines RENAULT : une approche par recherche locale à voisinage large. In *Actes du 6ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*.
2. M. PUTCHA et J. GOTTLIEB (2002). Solving car sequencing problems by local optimization. In *Applications of Evolutionary Computing, LNCS 2279*, p. 132–142. Springer-Verlag, Berlin, Allemagne.
3. M. PUTCHA, J. GOTTLIEB et C. SOLNON (2003). A study of greedy, local search and ant colony optimization approaches for car sequencing problems. In *Applications of Evolutionary Computing, LNCS 2611*, p. 246–257. Springer-Verlag, Berlin, Allemagne.
4. B. SMITH (1996). Succeed-first or fail-first : a case study in variable and value ordering heuristics. In *Proceedings of the Third Conference on Practical Applications of Constraint Technology*, p. 321–330.

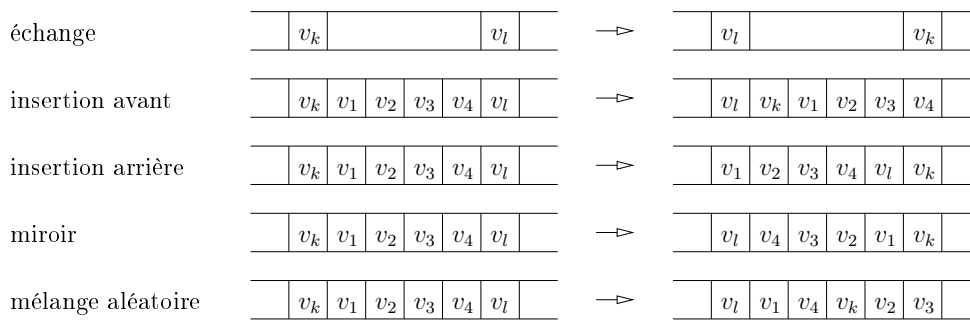


FIG. 1. Les cinq transformations.