Programmation par contraintes pour l'optimisation multicritère

Diego Olivier Fernandez Pons

Programmation par contraintes pour l'optimisation multicritère

Diego Olivier Fernandez Pons [12]

[1] Université Pierre et Marie Curie - Paris VI, 5 place Jussieu 75005 Paris
[2] ILOG S.A, 9 rue de Verdun 94253 Gentilly Cedex
diego-olivier.fernandez-pons@cicrp.jussieu.fr dofpons@ilog.fr

Résumé

Les problèmes d'optimisation combinatoire multicritère sont des problèmes ardus car ils combinent les difficultés des problèmes combinatoires classiques avec des préférences complexes sur leur solutions. C'est cependant sous cette forme que se présentent la plupart des problèmes industriels réels.

Après avoir examiné des travaux utilisant des procédures de programmation linéaire, relaxation lagrangienne ou énumération croissante pour les résoudre, nous en extrairons les idées fondamentales que nous réintroduirons dans une approche de programmation par contraintes.

Nous pourrons alors combiner ces méthodes avec des contraintes globales d'optimisation (chemin, arbres, couplages) pour résoudre certains problèmes multicritère.

Abstract

Multicriteria optimization problems are specially hard because they combine difficult combinatorial problems with complex preferences on their solutions. However this is how most real problems arise in practice.

We will see some techniques used in linear programming, langrangian relaxation and cost-increasing enumeration to solve multicriteria optimization problems.

We will reintroduce these techniques in constraint programming and combine them with optimization global constraints allowing us to solve some multicriteria optimization problems.

1 Introduction

La résolution de problèmes complexes en programmation par contraintes se fait par une approche de décomposition et coopération : D'abord le problème est ramené à une conjonction de contraintes $P = C_1 \wedge C_2 \wedge \ldots \wedge C_n$, où chacune des contraintes admet un algorithme de résolution efficace, souvent issu de la recherche opérationnelle. Ensuite est effectuée une recherche arborescente au cours de laquelle chaque contrainte fait des déductions locales en fonction des données connues (variables fixées, bornes inférieures ou supérieures), puis les propage aux autres contraintes afin que ces dernières puissent également en bénéficier.

Nous proposons de suivre une démarche similaire pour aborder certains problèmes d'optimisation combinatoire multicritère : optimisation sous contraintes de ressources, optimisation min-max, optimisation sous incertitude avec différents scénarios de données, optimisation sous incertitude représentées par des intervalles.

Ainsi nous partirons de contraintes globales d'optimisation pour des problèmes monocritère (arbre, chemin, couplage). Nous examinerons au fur et à mesure des méthodes de recherche opérationnelle déjà mises en oeuvre par d'autres auteurs pour ces problèmes. Puis nous les transposerons en programmation par contraintes en les combinant aux contraintes globales.

2 Contraintes globales d'optimisation

L'élément atomique de notre décomposition est la contrainte globale d'optimisation. Il existe des contraintes globales d'optimisation pour les problèmes d'arbres [1], couplages [10] et plus court chemin [11] et une méthode générale pour en créer à partir d'une formulation linéaire du problème considéré et des informations fournies par un simplexe [4][5].

Il nous faut cependant imposer que les contraintes globales d'optimisation mettent à disposition des autres éléments du solveur de contraintes les informations suivantes :

1. une solution $\underline{x}$ de coût minimal parmi l'ensemble des solutions possibles
2. pour tout arc a appartenant à $\underline{x}$, le coût de la plus petite solution ne contenant pas a
3. pour tout arc a n'appartenant pas à $\underline{x}$, le coût de la plus petite solution contenant a

La plupart des contraintes globales d'optimisation manipulent internement ces informations ou des valeurs proches.

2.1 Lien avec les problèmes d'énumération

Une des méthodes classiques pour résoudre des problèmes multicritères est l'énumération en ordre croissant des solutions (ranking methods). Ehrgott [3] propose un algorithme générique en deux phases qui s'appuie sur des algorithmes d'énumération croissante des solutions. Afin de montrer l'applicabilité de cette procédure, il recense 4 familles différentes d'algorithmes d'énumération parus dans la littérature :

1. Les plus courts chemins (Eppstein, Martins)
2. Les arbres couvrants (Gabow, Katoh)
3. Les affectations (Camerini et Hamacher)
4. Les flots (Chegireddy et Hamacher)

Il signale également l'existence d'algorithmes génériques dus à Lawler, et Hamacher et Queyranne. Il existe aussi des généralisations dans certaines classes de matroïdes.

De fait, les contraintes globales précédentes entretiennent des rapports étroits avec ces algorithmes [9]. En particulier, les informations fournies par les contraintes suffisent pour imposer au solveur de contraintes de parcourir les solutions en ordre des coûts croissants.

3 Interaction avec le solveur de contraintes

Le propos des contraintes globales est de réduire l'espace de recherche en éliminant des zones où il n'existe aucune solution meilleure que la meilleure solution disponible. Elles réagissent de ce fait aux évènements d'interdiction ou imposition d'un arc (par l'arbre de branchement ou par une autre contrainte) et de réduction de la borne supérieure.

Nous considérons les problèmes multicritères où :

- chaque critère est matérialisé par un scénario (une valuation c^k_{ij}, $(i,j) \in G$ d'un graphe G commun)
- la performance d'une solution sur un critère est la somme des coûts de chacun de ses éléments $c^k(s) = \sum_{ij \in s} c^k_{ij}$
- l'objectif est de trouver une solution préférée au sens d'une relation de préférence que nous allons définir

Etant donnée une borne B^k sur chaque critère, le solveur de contraintes s'occupe de trouver une solution réalisable s au problème de satisfaction :

$$\left[\sum_{ij \in s} c^1_{ij} < B^1\right] \wedge \left[\sum_{ij \in s} c^2_{ij} \leq B^2\right]$$

Notre moyen d'interaction avec le solveur de contraintes est de résoudre des suites de problèmes de satisfaction cette forme en modifiant les bornes B^k.

4 Optimisation sous contraintes de ressources

La modification la plus simple est de transformer l'un des critères en objectif : on recherche l'arbre couvrant de coût minimal, ne dépassant pas les bornes de ressource B_k allouées aux autres critères (ici r^k).

$$\left[\sum_{ij \in s} c_{ij} < \overline{C}_t\right] \wedge \left[\forall k,\ \sum_{ij \in s} r^k_{ij} \leq B^k\right]$$

On résoud donc une succession de problèmes P_t où la borne $\overline{C}_t$ prend la valeur de la solution du problème précédent, donc $\overline{C}_0 > \overline{C}_1 > \ldots > \overline{C}_m = opt$. Intuitivement, le solveur de contraintes va énumérer les solutions suivant le critère de coût jusqu'à ce que toutes les contraintes de ressource soient satisfaites.

Sellmann [12] à la suite des travaux de Dumitrescu [2] a récemment proposé de renforcer la propagation par des contraintes redondantes (autrement dit qui ne changent pas les solutions du problème mais en réduisent le temps de calcul) sous forme de combinaisons linéaire de ressources.

Comme ces deux auteurs utilisent des méthodes de relaxation lagrangienne qui calculent déjà des suites de graphes de la forme $(c_{ij} + \sum_k \lambda^k r^k_{ij})$ ils se servent des combinaisons linéaires déjà générées ainsi.

5 Optimisation min-max

Pour résoudre un problème d'optimisation min-max, il faut faire décroître toutes les bornes simultanément. On pose $z_t = \max_k \overline{C}^k_t$ et à chaque résolution, $\overline{C}^k_{t+1} = z_t$, si bien que l'on obtient une suite décroissante $z_0 > z_1 > \ldots > z_m = opt$

$$\left[\forall k,\ \sum_{ij \in s} c^k_{ij} < \overline{C}^k_t\right]$$

Hamacher et Ruhe [6] montrent qu'il existe un critère d'arrêt pour l'énumération croissante des solutions suivant toute combinaison linéaire convexe des critères $c_{ij}^{\lambda} = \sum_k \lambda_k c_{ij}^k$ avec $\sum_k \lambda^k = 1$. Ils montrent également que le choix des coefficients peut affecter grandement les performances et proposent diverses heuristiques pour leur calcul.

Les modifications que nous avons introduites suffisent à garantir la résolution du problème. Afin de retrouver l'approche d'Hamacher et Ruhe, il suffit d'introduire la combinaison linéaire convexe sous forme de contrainte supplémentaire et de demander au solveur d'énumérer suivant celle-ci.

$$\left[\sum_{ij \in s} \left(\sum_k \lambda^k c_{ij}^k \right) < \sum_k \lambda^k \overline{C}_t^k \right]$$

Il est aussi possible de résoudre des problèmes min-max regret (dans lequel la performance d'une solution pour chaque critère est la différence entre son coût et le meilleur coût qu'elle aurait pu avoir). En effet, les contraintes globales d'optimisation calculent des informations différentielles (augmentation du coût de la solution minimale quand on impose ou interdit un arc donné). Elles sont donc invariantes par translation de l'origine. Il est enfin possible de résoudre des problèmes d'optimisation Tchebychev en introduisant les coefficients multiplicatifs adaptés.

6 Optimisation robuste

Supposons désormais que nous ayons un problème min-max avec un nombre exponentiel de critères. La méthode classique pour aborder ce genre de problèmes en programmation linéaire est la génération de contraintes (décomposition de Benders, branch-and-cut) : on résoud une relaxation $P_k = C_1 \wedge C_2 \wedge \ldots \wedge C_k$ du problème où seul un petit nombre de contraintes figure. Si la solution s trouvée vérifie toutes les contraintes, elle est la solution globale du problème. Autrement on trouve des contraintes violées et on les ajoute au problème $P_{k+1} = P_k \wedge C_{k+1}$.

La justification théorique de la génération de contraintes en programmation linéaire est que, pour des raisons de dimension d'espaces vectoriels, la solution optimale est déterminée par un nombre restreint de contraintes. Autrement dit, si l'on dispose d'une bonne heuristique pour ajouter les contraintes au problème P_k la convergence sera rapide. Pour les problèmes de min-max en programmation par contraintes, une seule contrainte suffit à déterminer la solution optimale.

Ainsi, Aron et Van Hentenryck [1] ont pu mettre en oeuvre une méthode de génération de contraintes (de type branch and cut) pour un problème d'optimisation sous incertitude où l'incertitude est matérialisée par des intervalles dans les valeurs des arcs d'un graphe $c_{ij} \in [inf_{ij}, sup_{ij}]$. Il s'agit de trouver une solution (un arbre couvrant) au problème min-max regret avec autant de critères que de combinaisons différentes de valeurs pour c_{ij}

7 OWA et intégrales de Choquet

Les préférences rencontrées dans les parties précédentes sont toutes des cas particuliers d'OWA (ordered weighted average). Il s'agit de moyennes pondérées dans lesquelles les coefficients de pondération ne sont pas appliqués aux critères mais aux rangs :

$$z(x) = \sum_k \omega_{\sigma_x(k)} \left(\sum_{ij \in x} c_{ij}^k(x) \right)$$

où σ_x est une permutation qui ordonne les critères en ordre décroissant - en d'autres termes on commence par trier les performances $c^k(x)$ de x puis on applique la somme pondérée.

Or pour résoudre un problème de job-shop en programmation par contraintes, Zhou [13] a introduit une variante de la contrainte d'affectation, permettant de maintenir les bornes inférieures et supérieures d'une famille y_k, $y_1 \leq y_2 \leq \ldots \leq y_n$ sachant qu'elle doit être une permutation d'une autre famille x_k.

Enfin, les OWA sont elle mêmes des cas particuliers d'intégrales de Choquet, pour lesquelles Le Huede et al. ont introduit une contrainte globale [8][7]. Ce sont il nous semble autant de pistes intéressantes pour poursuivre ces travaux.

8 Conclusion

Nous nous sommes efforcés de montrer que les problèmes d'optimisation multicritère parce qu'ils sont à la frontière entre les problématiques combinatoires et décisionnelles sont certes particulièrement difficiles mais aussi particulièrement intéressants. Ainsi donnent-ils l'occasion d'appliquer des techniques issues de la recherche opérationnelle, la programmation par contraintes, la programmation linéaire ou encore de l'aide à la décision. Nous avons en particulier mis l'accent sur le rôle que pouvaient y jouer les contraintes globales et comment les combiner avec d'autres informations notamment celles issues des propriétés des préférences utilisées.

Références

[1] Ionut D. Aron and Pascal Van Hentenryck. A constraint satisfaction approach to the robust spanning tree problem with interval data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 18–25, 2002.

[2] Irina Dumitrescu and Natashia Boland. Improved preprocessing, labeling and scaling algorithms for the weight constrained shortest path problem. *Networks*, 42(3) :135–153, 2003.

[3] Matthias Ehrgott and Anders J. V. Skriver. Solving biobjective combinatorial max-ordering problems by ranking methods and a two-phases approach. *European Journal of Operational Research*, 147 :657–664, 2003.

[4] Filippo Focacci, Andrea Lodi, and Michela Milano. Cost-based domain filtering. In *Proceedings of the 5th International conference on principles and practice of constraint programming (CP)*, pages 189–203, 1999.

[5] Filippo Focacci, Andrea Lodi, and Michela Milano. Cutting planes in constraint programming : A hybrid approach. In *Proceedings of the 6th International conference on principles and practice of constraint programming (CP)*, pages 187–201, 2000.

[6] Horst W. Hamacher and Günter Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52 :209–230, 1997.

[7] F. Le Huede, M. Grabisch, C. Labreuche, and P. Savéant. Multicriteria search in constraint programming. In *Proceedings of the 5th International Workshop on integration of AI and OR Techniques in constraint programming for combinatorial optimization problems (CPAIOR)*, 2003.

[8] F. Le Huede, P. Gérard, M. Grabisch, C. Labreuche, and P. Savéant. Integration of a multicriteria decision model in constraint programming. In *Proceedings of the AIPS'02 Workshop on Planning and Scheduling with Multiple Criteria*, pages 15–20, 2002.

[9] Diego Olivier Fernandez Pons. Au sujet de certaines contraintes globales. In *Journées Francophones de Programmation par Contraintes (JFPC)*, 2005.

[10] Jean-Charles Régin. Arc consistency for global cardinality constraints with costs. In *Proceedings of the 5th International conference on principles and practice of constraint programming (CP)*, pages 390–404, 1999.

[11] Meinolf Sellmann. Cost-based filtering for shorter path constraints. In *Proceedings of the 9th International conference on principles and practice of constraint programming (CP)*, pages 694–708, 2003.

[12] Meinolf Sellmann. Theoretical foundations of CP-based lagrangian relaxation. In *Proceedings of the 10th international Conference on the Principles and Practice of Constraint Programming (CP)*, pages 634–647, 2004.

[13] Jianyang Zhou. A constraint program for solving the job-shop problem. In *Proceedings of the 2nd International conference on principles and practice of constraint programming (CP)*, pages 510–542, 1996.