Discussion et Traitements des DCSP pour la Configuration

Mathieu Veron
Access Commerce,
rue Ampère - BP 555
31674 Labège Cedex
email: mv@access-commerce.com

Résumé

La recherche d'un formalisme convenant à la fois pour la représentation des connaissances liées aux produits configurables et à leur configuration a donné naissance à de nombreux modèles. Nous prenons ici le contre-pied de l'approche traditionnelle à deux niveaux (contraintes structurelles d'une part et contraintes techniques d'autre part) en mêlant au sein d'un même problème de satisfaction de contraintes l'ensemble des règles et contraintes.

1 Introduction

Un produit configurable, est un produit contenant des options et des variantes. Il n'existe pas dans un catalogue, mais sera (re-)défini pour chaque client. Exemple : un ordinateur, une menuiserie, un service bancaire. Afin de permettre la configuration de tel produit, il est nécessaire de les modéliser dans un formalisme à même de représenter l'ensemble des produits faisables sous une forme compacte. Puis, il faut pouvoir, de manière interactive, capturer les choix du client et s'assurer à tout moment de la faisabilité du produit, pour aboutir à un produit valide entièrement défini.

Actuellement les directions de recherches et les approches industrielles retenues pour les logiciels de configuration misent sur le formalisme des contraintes (CSP). Mais le formalisme des CSPs nécessite quelques aménagements afin de répondre à deux besoins considérés comme élémentaires en configuration. Premièrement, l'ensemble des variables ne participent pas obligatoirement toutes simultanément à une solution. Deuxièmement, les modèles doivent pouvoir être structurés en une hiérarchie de sous-problèmes.

2 Travaux précédents

Le modèle des Dynamics CSP [MF90] fut le premier à être défini afin de répondre à la première interrogation. Mêlant règles d'activation des variables (ou Activity Constraints, AC) et contraintes "classiques" permettant d'encoder les affectations de variables compatibles (ou Compatibility Constraints, CC). A chaque étape, en se basant sur les ACs, l'algorithme déduit le problème formé des variables et des contraintes actives , sur lequel il assure un niveau de cohérence donné. Afin de répondre au besoin de structuration du modèle, d'autres cadres furent proposés, dont, les composites CSP ([SF96]). Dans ce cadre, la structure du problème est représentée par l'inclusion de sous-problèmes, dans les domaines des variables de méta-problèmes. Une affectation valide pour une variable est donc un CSP, dont les composants (variables et contraintes) se substituent à la variable affectée.

3 Représentation par variables d'état et contraintes

Nous désirions conserver l'expressivité des DCSPs (au niveau des ACs), mais ne pas faire appel à deux algorithmes séparés. La structure bien établie des CCSPs est également séduisante, mais trop rigide. Pour concilier tout cela, nous proposons d'encoder le modèle d'un produit configurable au sein d'une structure de données récursive, et de représenter l'état des variables par une variable d'état. Les ACd sont transformées en contraintes impliquant ces variables d'état. En ajoutant explicitement au modèle des contraintes exprimant le fait que lorsqu'un élément est inactif, tous les éléments qui le composent sont eux-mêmes inactifs, on obtient un modèle uniquement basé sur le formalisme CSP. Un même algorithme de maintien de cohérence assurera une propagation sur les domaines des variables d'état et établira un niveau de cohérence pré-défini sur l'ensemble du problème actif (variables actives et contraintes ayant toutes leurs variables actives). Nous avons expérimenté les gains de performances résultant de l'application d'un niveau de cohérence plus élevé sur le sous-problème composé des contraintes actives contenant des variables d'état. Les premiers résultats sont encourageants même s'ils dépendent fortement de la structure du problème, structure qu'il est difficile de caractériser.

4 Conclusion

Le choix de représenter l'état d'un élément du modèle par une variable permet d'élargir l'expressivité des modèles précédents (gestion d'un état "optionnel" par exemple). De plus tout encoder par des contraintes permet d'uniformiser le traitement de l'activité des variables et de la cohérence des domaines de définition.

Références

[MF90] Sanjay Mittal and Brian Falkenhainer. Dynamic constraint satisfaction problems. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 25–32, 1990.

[SF96] Daniel Sabin and Eugene Freuder. Configuration as composite constraint satisfaction. In *Proceedings of the American Association of Artificial Intelligence Fall Symposium*, pages 28–36, 1996.