
Portfolio Adaptatif pour la Parallélisation d'un solveur de Programmation Par Contraintes

Tarek Menouer*¹ and Bertrand Le Cun*¹

¹Parallélisme, Réseaux, Systèmes d'information, Modélisation (PRISM) – CNRS : UMR8144, Université de Versailles Saint-Quentin-en-Yvelines (UVSQ) – 45 avenue des Etats-Unis Bâtiment Descartes 78035 Versailles CEDEX, France

Résumé

L'objectif de notre travail est de proposer un solveur parallèle en se basant sur le principe de Portfolio dans le but de réduire le temps de résolution des problèmes de Programmation Par Contraintes (PPC). Le principe de Portfolio est largement utilisé dans la parallélisation des solveurs SAT (boolean SATisfiability), il consiste à lancer N stratégies de recherche en utilisant N coeurs de calcul. Chaque coeur prend une stratégie et il effectue une exploration locale de l'espace de recherche sans communiquer avec les autres coeurs de calcul. La première stratégie qui répond au besoin de l'utilisateur arrête toutes les autres stratégies. Le problème est que le nombre de stratégies de recherche est limité comparé au nombre actuel de coeurs de calcul utilisables dans les machines parallèles.

L'idée de notre travail consiste à lancer N stratégies de recherche pour la même modélisation d'un problème de PPC et ordonnancer ces N stratégies sur P coeurs de calcul ($P > N$).

La nouveauté est que l'ordonnancement de ces N stratégies est effectué d'une façon totalement dynamique entre les différents coeurs de calcul. Le but est d'adapter l'ordonnancement de la recherche de manière à privilégier la/les stratégie(s) qui donne(nt) le résultat le plus rapidement.

Les performances obtenues avec notre portfolio solveur sont illustrées par la résolution des problèmes de satisfaction et d'optimisation de contraintes. Ces problèmes sont modélisés avec le format FlatZinc et résolus avec un solveur de PPC appelé Google OR-Tools porté au dessus de notre framework parallèle Bobpp.

Mots-Clés: Parallélisme, Équilibrage de charge dynamique, Optimisation Combinatoire, Ordonnancement

*Intervenant