Construction des séries d'états dans l'algorithme Divide-and-Evolve

Jacques Bibai, Marc Schoenauer, Pierre Savéant

Construction des séries d'états dans l'algorithme *Divide-and-Evolve*

Jacques BIBAÏ[1,2], Marc SCHOENAUER[1], and Pierre SAVÉANT[2]

[1] Projet TAO, INRIA Saclay & LRI, Université Paris Sud, Orsay, France.
prenom.nom@inria.fr
[2] Thales Research & Technology, Palaiseau, France.
prenom.nom@thalesgroup.com

Mots-Clefs. Divide-and-Evolve ; planification temporelle ; décomposition en série d'états ; algorithme évolutionnaire ; optimisation combinatoire.

1 L'approche *Divide-and-Evolve*

Nous nous intéressons à la résolution générique des problèmes de planification temporelle de type STRIPS, telle qu'elle est définie par le formalisme PDDL2.1 [5]. Plusieurs approches ont été proposées pour la résolution de problèmes de planification temporelle, combinant des techniques de recherche heuristique de résolution des problèmes de planification classique (considérant que toutes les actions ont la même durée) aux techniques issues de la recherche opérationnelle [1,2]. Mais ces techniques de résolution, certes efficaces, ne passent pas à l'échelle lorsque la taille de l'espace des états et/ou le nombre d'actions possibles augmentent. Afin de surmonter cette difficulté, plusieurs méthodes sub-optimales ont été proposées. Certaines d'entre elles permettent de produire rapidement des solutions, mais souvent au prix d'une mauvaise qualité du plan généré [9].

L'approche *Divide-and-Evolve* (DAE) [3] est basée sur l'hypothèse de l'existence de séquences d'états intermédiaires $E_0 = I, E_1, \ldots, E_n = G$ "faciles à atteindre". Dans un premier temps, le problème initial de planification temporelle $\mathcal{P}_D(I, G)$ est décomposé par évolution artificielle [4] en une séquence de sous-problèmes $\mathcal{P}_D(E_i, E_{i+1})$ qui sont sous-traités à un planificateur temporel existant. Si tous les sous-problèmes sont résolus, les solutions partielles ainsi trouvées sont ensuite recomposées pour obtenir une solution (sub-optimale) du problème initial. Dans son implantation actuelle, chaque sous-problème $\mathcal{P}_D(E_i, E_{i+1})$ est résolu par le planificateur exact CPT (Constraint Programming Temporal) [2].

Nous nous intéresserons ici, au sein de l'approche DAE, à l'étape particulière que constitue la construction des séquences d'états que l'on espère faciles à résoudre. Le lecteur intéressé par le détail de l'algorithme *Divide-and-Evolve* pourra se reporter à [7] et à [6].

2 Construction des états

La construction d'états est un des points critiques de l'algorithme DAE qui intervient dans les phases d'initialisation et génération de nouvelles solutions candidates à l'aide des opérateurs de variations. La phase d'initialisation consiste à une construction stochastique d'un ensemble de séries d'états potentiel, qui est le point de départ de l'optimisation. A

chaque génération, les séries d'états possédant les meilleures évaluations seront modifiées par des opérateurs de variation pour la production de nouvelles solutions.

Deux aspects principaux guideront la conception de l'algorithme : essayer de respecter la cohérence des états E_i, et restreindre au maximum la taille de l'espace de recherche en utilisant des connaissances a priori que l'on peut déduire du problème. Deux points de vue vont être envisagés, différant par le degré de prise en compte des relations entre atomes constituant un état donné. Ainsi, dans la **construction aveugle**, nous nous intéresserons à la construction des séries d'états $E_0 = I, E_1, \ldots, E_n = G$ grâce à des choix uniformes des atomes dans l'ensemble des atomes tout en garantissant la cohérence 2 à 2 des atomes dans chaque E_i. La **construction orientée** quant à elle utilisera en plus une estimation du temps au plus tôt d'apparition d'un atome dans toute solution du problème lors de la construction de la série d'états $E_0 = I, E_1, \ldots, E_n = G$.

3 Expérimentations

Le choix des atomes permettant la description des états étant un problème ouvert [7], nous avons implémenté en deux versions la construction aveugle des séries d'états. La première version (1) utilise les atomes de l'état final et leurs voisins pour la construction des séries d'états et la seconde construit ces séries d'états grâce aux atomes issus de la règle définie dans [6]. Cette règle permet de restreindre l'ensemble des atomes de base utilisé pour la construction des séries d'états aux atomes construits à partir d'un nombre de prédicats fixé. L'implémentation de la construction orientée des séries d'états est obtenue à partir de (1) en remplaçant ces opérateurs de variation et son initialisation par ceux implémentant la méthode orientée. L'heuristique $h2$ définie dans [10] a été utilisée pour l'estimation des dates.

Nos expériences sur 4 domaines des benchmarks l'*Internationnal Planning Competition* (IPC) ont montré qu'aucune méthode n'est meilleure (en proportion d'optima trouvée) que les autres sur les 4 domaines. Lors de notre présentation, nous exposerons ces différentes méthodes de construction des séries d'états ainsi que des résultats de comparaisons.

Références

1. Steven A. Wolfman and Daniel S. Weld. Combining linear programming and satisfiability solving for resource planning. The Knowledge Engineering Review, 16, Cambridge University Press, New York (2001)
2. V. Vidal and H. Geffner. Branching and Pruning : An Optimal Temporal POCL Planner based on Constraint Programming. Artificial Intelligence, 170,pages 298-335, (2006)
3. Marc Schoenauer and Pierre Savéant and Vincent Vidal. Divide-and-Evolve : a Sequential Hybridization Strategy using Evolutionary Algorithms. Advances in Metaheuristics for Hard Optimization, Z. Michalewicz and P. Siarry, Springer, pages 179-198 (2007)
4. A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*, Springer Verlag (2003).
5. M. Fox and D. Long. PDDL2.1 : An extension to PDDL for expressing temporal planning domains. JAIR, 20 :61-124, 2003.
6. J. Bibai and M. Schoenauer and P. Savéant and V. Vidal. DAE : Planning as Artificial Evolution (Deterministic part). In 6th International Planning Competition Booklet (IPC-2008).
7. J. Bibai and M. Schoenauer and P. Savéant and V. Vidal. Planification Evolutionnaire par Décomposition. INRIA research report NRT-0355 (2008). URL : http ://hal.inria.fr/inria-00322880/en/
8. T. Bylander. The Computational Complexity of Propositional STRIPS planning. Artificial Intelligence, 69(1-2) : 165-204 (1994)
9. S. Edelkamp and M. Helmert. Mips - The Model Checking Integrated Planning System. AI Magazine Volume 22 Number 3 (2001).
10. P. Haslum and H. Geffner. Heuristic planning with time and resources. European Conference of Planning (ECP-01), 121-132 (2001)