CPT: An Optimal Temporal POCL Planner based on Constraint Programming

Vincent Vidal
CRIL - Université d'Artois
rue de l'université - SP16
62307 Lens Cedex, FRANCE
vidal@cril.univ-artois.fr

Héctor Geffner
ICREA & Universitat Pompeu Fabra
Paseo de Circunvalacion 8
08003 Barcelona, SPAIN
hector.geffner@upf.edu

CPT is a new domain-independent temporal planner that combines a branching scheme based on Partial Order Causal Link (POCL) Planning with powerful and sound pruning rules implemented as constraints. Unlike other recent approaches that build on POCL planning (Nguyen & Kambhampati 2001; Younes & Simmons 2003), CPT is an optimal planner that minimizes makespan. The details of the planner and its underlying formulation are described in (Vidal & Geffner 2004) that is focused on the computation of 'canonical plans' where ground actions are not done more than once in the plan. The version used in the competition, removes this restriction and computes optimal temporal plans, whether canonical or not.

The development of CPT is motivated by the limitation of heuristic state approaches to parallel and temporal planning that suffer from a high branching factor (Haslum & Geffner 2001) and thus have difficulties matching the performance of planners built on SAT techniques such as Blackbox (Kautz & Selman 1999). In CPT, all branching decisions (resolution of open supports, support threats, and mutex threats), generate binary splits, and nodes σ in the search correspond to 'partial plans' very much as in POCL planning.

While ideally, one would like to have informative lower bounds $f(\sigma)$ on the makespan $f^*(\sigma)$ of the best complete plans that expand σ, so that the partial plan σ can be pruned if $f(\sigma) \not\leq B$ for a given bound B, such lower bounds are not easy to come by in the POCL setting. CPT thus models the planning domain as a temporal constraint satisfaction problem, adds the constraint $f^*(\sigma) \leq B$ for a suitable bound B on the makespan, and performs limited form of constraint propagation in every node σ of the search tree. The novelty of CPT in relation to other temporal POCL planners such as IxTET (Laborie & Ghallab 1995) and RAX (Jonsson *et al.* 2000), that also rely on constraint propagation (and Dynamic CSP approaches such as (Joslin & Pollack 1996)), is the formulation that enables CPT to reason about actions a that are not yet in the plan. Often a lot can be inferred about such actions including restrictions about their possible starting times and supports. Some of this information can actually be inferred before any commitments are made; the lower bounds on the starting times of *all* actions as computed in Graphplan being one example (Blum & Furst 1995). CPT thus reasons with CSP variables that involve *all* the actions a in the domain and not only those present in the current plan, and for each such action, it deals with two variables $S(p, a)$ and $T(p, a)$ that stand for the possibly undetermined action supporting precondition p of a, and the possibly undetermined starting time of such an action. A causal link $a'[p]a$ thus becomes a constraint $S(p, a) = a'$, which in turn implies that the supporter a' of precondition p of a starts at time $T(p, a) = T(a')$. A number of constraints enforce the correspondences among these variables. At the same time, the heuristic functions for estimating costs in a temporal setting, as introduced in (Haslum & Geffner 2001), are used to initialize variables domains and some 'distances' between actions (Van Beek & Chen 1999).

The CPT planner is implemented using the Choco CP library (Laburthe 2000) that operates on top of Claire, (Caseau, Josset, & Laburthe 1999), a high-level programming language that compiles into C++. Further details can be found in (Vidal & Geffner 2004) that is concerned mostly with the computation of optimal canonical plans; plans where no ground action is done more than once. The version of CPT used in the competition removes this restriction, and computes optimal temporal plans, whether canonical or not. Currently, the semantics of these plans follows the one in (Smith & Weld 1999) where interfering actions are not allowed to overlap in time. This condition has been relaxed in PDDL 2.1 where interfering actions may overlap sometimes (e.g., when preconditions do not have to be preserved throughout the execution of the action). We are currently trying to accommodate that semantics as well.

References

Blum, A., and Furst, M. 1995. Fast planning through planning graph analysis. In *Proceedings of IJCAI-95*, 1636–1642. Morgan Kaufmann.

Caseau, Y.; Josset, F. X.; and Laburthe, F. 1999. Claire: Combining sets, search and rules to better express algorithms. In *Proceedings of the Int. Conf. on Logic Programming*.

Haslum, P., and Geffner, H. 2001. Heuristic planning with time and resources. In *Proc. European Conference of Planning (ECP-01)*, 121–132.

Jonsson, A.; Morris, P.; Muscettola, N.; and Rajan, K. 2000. Planning in interplanetary space: Theory and practice. In *Proc. AIPS-2000*, 177–186.

Joslin, D., and Pollack, M. E. 1996. Is "early commitment" in plan generation ever a good idea? In *Proceedings AAAI-96*, 1188–1193.

Kautz, H., and Selman, B. 1999. Unifying SAT-based and Graph-based planning. In Dean, T., ed., *Proceedings IJCAI-99*, 318–327. Morgan Kaufmann.

Laborie, P., and Ghallab, M. 1995. Planning with sharable resources constraints. In Mellish, C., ed., *Proc. IJCAI-95*, 1643–1649. Morgan Kaufmann.

Laburthe, F. 2000. Choco: implementing a cp kernel. In *Proceedings CP-00, Lecture Notes in CS, Vol 1894*. Springer.

Nguyen, X. L., and Kambhampati, S. 2001. Reviving partial order planning. In *Proc. IJCAI-01*.

Smith, D., and Weld, D. 1999. Temporal planning with mutual exclusion reasoning. In *Proc. IJCAI-99*, 326–337.

Van Beek, P., and Chen, X. 1999. CPlan: a constraint programming approach to planning. In *Proc. National Conference on Artificial Intelligence (AAAI-99)*, 585–590. AAAI Press/MIT Press.

Vidal, V., and Geffner, H. 2004. Branching and pruning: An optimal temporal POCL planner based on constraint programming. In *Proceedings AAAI-04*. To appear.

Younes, B. L. S., and Simmons, R. G. 2003. VHPOP: Versatile heuristic partial order planner. *Journal of AI Research* 20:405–430.