CO-PLAN: Combining SAT-Based Planning with Forward-Search

Nathan Robinson, Charles Gretton and **Duc-Nghia Pham**
SAFE Program, NICTA Ltd., Australia and
Institute for Integrated and Intelligent Systems, Griffith University, Australia

Abstract

This extended-abstract introduces CO-PLAN, a two-phase propositional planning system for the cost-optimal case. During the first phase, CO-PLAN constructs the n-step plangraph for increasing values of n, passing the corresponding decision problem at each stage to a modified Boolean satisfiability procedure. This procedure determines whether an n-step plan exists, and if so identifies a plan that has the minimal action cost given the plan length bound n. The second phase proceeds as soon as the first phase yields a plan. This consists of a forward-search in the problem state space, bounded by the action-cost of the best plan found during the first phase.

Background

The planning-as-SAT(isfiability) paradigm has dominated recent optimal tracks of International Planning Competitions. The winner of the optimal track in the 2004 International Planning Competition (IPC-4) was SATPLAN-04, and in the 2006 competition (IPC-5) SATPLAN-06 (Kautz, Selman, & Hoffmann 2006) and MAXPLAN (Chen, Xing, & Zhang 2007) tied for first place. These solvers, all descended from BLACKBOX (Kautz & Selman 1999), compile the problem posed by an n-step *plangraph* (Blum & Furst 1997) into a *conjunctive normal form* (CNF) formula. A plan is then computed by a dedicated SAT solver, such as Lawrence Ryan's SIEGE.

The above planning systems seek parallel step-optimality, meaning that they find a parallel plan that uses the minimal number of plan-steps. To efficiently find a step-optimal plan, SAT-based techniques employ *query strategies* such as the *binary-search* used by an early version of SATPLAN (Kautz & Selman 1996), and the *ramp-up* and *ramp-down* (*backward-level reduction*) strategies employed by SATPLAN-04/06 (Kautz 2006) and MAXPLAN (Chen, Xing, & Zhang 2007) respectively. *Ramp-up* starts with $n = 1$, and then incrementally increases this by one step until a solution is found. *Ramp-down* starts with $n = k$ for some upper bound k (obtained in practice using the fast satisficing planner FASTFORWARD (Hoffmann & Nebel 2001)) and proceeds towards the optimal.

We should note that alternate query strategies have been proposed for SAT-based planning, and more generally for solving discrete optimisation problems using decision procedures. In particular, Rintanen (2004) observed that interleaved (or parallel) evaluation of decision problems usually finds a plan faster than sequential evaluation. They obtained further runtime efficiency for the interleaved case by allocating a fraction of the SAT solvers time to the problem at horizon $n+1$ than is allocated at n. In the same vein, Streeter & Smith (2007) describe a binary search strategy that quickly finds an approximately optimal plan.

Not all SAT-based planning systems are designed for parallel step-optimality in a propositional planning setting. For example, one mode of the Medic (Ernst, Millstein, & Weld 1997) system is an efficient planner for the case of linear (resp. parallel) step-optimal planning. More recently, Rintanen, Heljanko, & Niemelä (2006) and Wehrle & Rintanen (2007) propose reducing the number of queries required to achieve a plan by relaxing constraints on action parallelism. In this case, parallel step-optimality is sacrificed for planning efficiency. Making them even more distinct from SAT-based entries in IPC-4 and IPC-5, all of the above systems propose direct encodings of planning-as-SAT, thus do not exploit plangraph analysis. Finally, SAT-based planners have been proposed for solving resource-constrained domains where numerical constraints play a key role in determining the validity of plans (Hoffmann & Nebel 2001).

For the optimal propositional track of IPC-6, competing planners must minimise the *total action cost* of a plan. In this respect IPC-6 differs significantly from past competitions. Existing SAT-based planning systems are unable to compete in IPC-6 directly, because even in the case where they guarantee step-optimality, the resultant plan may have a higher cost than a cost-optimal counterpart. We are not aware of any existing SAT-based planning system that guarantees cost-optimality.

Our Approach

We have developed the cost-optimal planner CO-PLAN, that proceeds in two phases. A Flow-Chart for CO-PLAN is given in Figure 1. During the initial phase of processing, CO-PLAN iteratively constructs propositional plangraphs for successively larger horizons n, while ignoring action costs, and performing reachability and neededness analysis (Blum & Furst 1997). The problem posed by the n^{th} plangraph is compiled into a CNF formula whose solutions, if any, correspond to parallel n-step plans. During the first phase of processing, CO-PLAN exploits a modified version

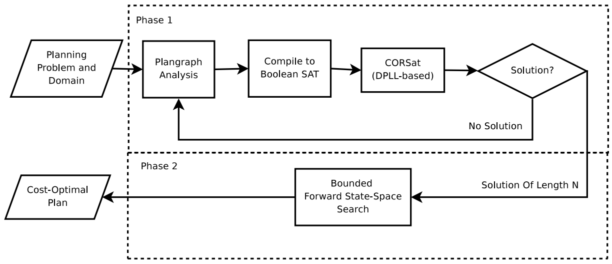

Figure 1: Flow-chart for CO-PLAN.

of RSAT (Pipatsrisawat & Darwiche 2007) called CORSAT to process the CNF formulae. RSAT won gold medals for the SAT+UNSAT and UNSAT problems of the Industrial Category of the 2007 International SAT Competition. The modifications we make for CORSAT enable it to identify: (1) Whether a solution exists for the given decision problem, and (2) if a solution exists, CORSAT identifies that with minimal action cost.[1] In summary, given a cost schedule for variables occurring in a satisfiable query formula, CORSAT returns a satisfying assignment with minimal cost. Consequently, the first phase of CO-PLAN produces a step-optimal plan with minimised action cost.

In a direction somewhat opposite to that taken by Chen, Xing, & Zhang for MAXPLAN, CO-PLAN exploits the speed and efficiency of SAT-based planners to obtain a good admissible initial bound on the cost of an optimal plan. The second phase of CO-PLAN then performs a bounded forward-search in the problem state space. This search makes use of a number of computationally cheap static problem analysis procedures during a preprocessing step. Along the lines of (Haslum & Jonsson 2000), as we ground domain operators we ignore actions whose preconditions are statically false. We also perform *static relevance* testing on the remaining actions as described by Bacchus & Teh (1998) for their forward chaining planner. Finally, during search, states which are closest to the goal, in the sense that they satisfy the most goal conditions, are explored with the highest priority. Ties are broken arbitrarily.

Summary

We presented CO-PLAN, a two phase planning system that uses a SAT-based planning procedure to bound the search space of a cost-optimal state-based forward-search planner. Along the lines of SATPLAN-06 and MAXPLAN, we leverage the latest technology from the SAT community to address planning by solving the corresponding decision problem. Moreover, we do this in a cost-optimal setting. Our approach exploits plangraph analysis during its first phase to ensure a compact and efficient encoding of the planning problem as SAT. During the second phase, we employ cheap static problem analysis to reduce consideration of redundant states and actions during the forward-search. CO-PLAN shall participate in the optimal track at IPC-6.

[1] It should be noted that DPLL procedures have been modified previously for finding a collection of satisfying models (Huang & Darwiche 2004).

Acknowledgements

Thanks to the authors Knot Pipatsrisawat and Adnan Darwiche of RSAT and authors Henry Kautz, Joerg Hoffmann, and Bart Selman of SATPLAN-06 for permission to incorporate parts of their respective systems in CO-PLAN.

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Bacchus, F., and Teh, Y. W. 1998. Making forward chaining relevant. In *In Proc. AIPS*, volume 4, 54–61. AAAI Press.

Blum, A., and Furst, M. 1997. Fast planning through planning graph analysis. *Artificial Intelligence* (90):281–300.

Chen, Y.; Xing, Z.; and Zhang, W. 2007. Long-distance mutual exclusion for propositional planning. In Proc. *IJCAI*.

Ernst, M.; Millstein, T.; and Weld, D. S. 1997. Automatic sat-compilation of planning problems. In Proc. *IJCAI*.

Haslum, P., and Jonsson, P. 2000. Planning with reduced operator sets. In *Artificial Intelligence Planning Systems*, 150–158.

Hoffmann, J., and Nebel, B. 2001. The ff planning system: Fast plan generation through heuristic search. *J. Artif. Intell. Res. (JAIR)* 14:253–302.

Huang, J., and Darwiche, A. 2004. Using dpll for efficient obdd construction. In *SAT*.

Kautz, H., and Selman, B. 1996. Pushing the envelope: Planning, propositional logic, and stochastic search. In Proc. *AAAI*.

Kautz, H. A., and Selman, B. 1999. Unifying SAT-based and graph-based planning. In Proc. *IJCAI*.

Kautz, H. A.; Selman, B.; and Hoffmann, J. 2006. SatPlan: Planning as satisfiability. In Abstracts *IPC-05*.

Kautz, H. A. 2006. Deconstructing planning as satisfiability. In Proc. *AAAI*.

Pipatsrisawat, K., and Darwiche, A. 2007. Rsat 2.0: Sat solver description. Technical Report D–153, Automated Reasoning Group, Computer Science Department, UCLA.

Rintanen, J.; Heljanko, K.; and Niemelä, I. 2006. Planning as satisfiability: parallel plans and algorithms for plan search. *Artificial Intelligence* 170(12-13):1031–1080.

Rintanen, J. 2004. Evaluation strategies for planning as satisfiability. In *ECAI*.

Streeter, M., and Smith, S. 2007. Using decision procedures efficiently for optimization. In *ICAPS*.

Wehrle, M., and Rintanen, J. 2007. Planning as satisfiability with relaxed e-step plans. In Proc. *AAI*.