IPPLAN: Planning as Integer Programming

Menkes van den Briel
Department of Industrial Engineering
Arizona State University
Tempe AZ, 85287-8809
menkes@asu.edu

Subbarao Kambhampati
Department of Computer Science
Arizona State University
Tempe AZ, 85287-8809
rao@asu.edu

Thomas Vossen
Leeds School of Business
University of Colorado at Boulder
Boulder CO, 80309-0419
vossen@colorado.edu

Overview

IPPLAN is an integer programming based planning system. It builds on the previous work of planning as integer programming, including that of: ILP-PLAN by Kautz and Walser (1999), the state change encoding by Vossen *et al.* (1999), Optiplan by van den Briel and Kambhampati (2005), and most significantly the state change flow encodings by van den Briel, Vossen, and Kambhampati (2005). Moreover, it adds on to the existing planning compilation approaches, including that of: SATPLAN by Kautz and Walser (1992), and GP-CSP by Do and Kambhampati (2000).

The current version of IPPLAN consists of two separate modules: (1) a translator written in Python, and (2) an integer programming modeler written in C++.

In order to solve a planning problem, the two modules are run consecutively. The translator is run first, and transforms a PDDL input into a state variable representation based on the SAS+ formalism. The integer programming modeler is run second, and generates the needed data structures and formulates the planning problem as an integer programming problem. The resulting integer programming problem is then solved using CPLEX (ILOG 2002).

The translator is an extension to the preprocessing algorithm of MIPS (Edelkamp & Helmert 1999). It was designed and developed by Helmert (2006) as one of the components for the Fast Downward planner. The translator is a stand alone component and therefore can easily be incorporated into other applications. The purpose of the translator is to ground all operators and axioms, convert the propositional (binary) representation to a state variable (multi-valued) representation of the planning problem, and to compile away most of the ADL features. A detailed description of the translator and its translation algorithm is described by Helmert (2006).

IPPLAN can support a collection of integer programming formulations. Currently, IPPLAN supports the One State Change (1SC) and the Generalized One State Change (G1SC) formulations as described by van den Briel, Vossen, and Kambhampati (2005). Both these formulations are restricted to solve propositional planning problems only, so currently IPPLAN is a propositional planning system. In the future, however, we would like to add more formulations to IPPLAN and broaden the scope of planning problems that it can handle.

When the 1SC formulation is used IPPLAN will find optimal makespan plans. With the G1SC formulation IPPLAN will not guarantee optimality, but generally find plans with few number of actions. In both these formulations state changes in the state variables are modeled as flows in an appropriately defined network. As a consequence, the integer programming formulations can be interpreted as a network flow problems with additional side constraints.

IPPLAN uses CPLEX (ILOG 2002) for solving the integer programming problems. CPLEX is a commercial software package that solves linear programming, mixed integer programming, network flow, and convex quadratic programming problems.

References

Do, M., and Kambhampati, S. 2000. Solving planning graph by compiling it into a CSP. In *Proceedings of the 5th International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000)*, 82–91.

Edelkamp, S., and Helmert, M. 1999. Exhibiting knowledge in planning problems to minimize state encoding length. In *Proceedings of the European Conference on Planning (ECP-99)*, 135–147. Springer-Verlag.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 25:(Accepted for publication).

ILOG Inc., Mountain View, CA. 2002. *ILOG CPLEX 8.0 user's manual*.

Kautz, H., and Selman, B. 1992. Planning as satisfiability. In *Proceedings of the European Conference on Artificial Intelligence (ECAI-1992)*.

Kautz, H., and Walser, J. 1999. State-space planning by integer optimization. In *AAAI-99/IAAI-99 Proceedings*, 526–533.

van den Briel, M., and Kambhampati, S. 2005. Optiplan: Unifying IP-based and graph-based planning. *Journal of Artificial Intelligence Research* 24:623–635.

van den Briel, M.; Vossen, T.; and Kambhampati, S. 2005. Reviving integer programming approaches for ai planning: A branch-and-cut framework. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-2005)*, 161–170.

Vossen, T.; Ball, M.; Lotem, A.; and Nau, D. 1999. On the use of integer programming models in AI planning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 304–309.