Introduction to the Special Issue on Planning: Research Issues at the Intersection of Planning and Constraint Programming

Alexander Nareyek (alex@ai-center.com)
Computer Science Department, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891, USA

Subbarao Kambhampati (rao@asu.edu)
Department of Computer Science, College of Engineering and Applied Sciences,
Arizona State University, Tempe, AZ 85287-5406, USA

Members of the constraint programming community are usually more familiar with applications like scheduling than with *action planning* — the topic of this special issue. However, action planning tackles problems belonging to a similar domain and can be interpreted as an extension of the scheduling problem by the problem of selecting which actions/tasks a plan should include so that a set of given goals is satisfied/optimized.

The connections between planning and constraint satisfaction have been recognized, at least since the development of the MOLGEN planner (Stefik, 1981). Much of the subsequent work at the intersection of planning and CSP involved tackling subproblems of plan synthesis that can be posed as constraint satisfaction problems, such as the management of resources, temporal constraints, and the domains of planning objects. Well-known planning systems like SIPE (Wilkins, 1988), O-Plan (Tate et al., 1994) and IxTeT (Laborie and Ghallab, 1995) have posed such tasks as constraint satisfaction problems. The planning field often benefits here from techniques developed in the context of scheduling.

The advent of Graphplan (Blum and Furst, 1997) further strengthened the connection between plan synthesis and constraint programming. Many of the explanations of Graphplan's and its extensions' impressive performance pointed to the connections between its graph search and CSP techniques (Kambhampati, 2000; Rintanen, 1998). Work on variants of Graphplan has shown that bounded-length plan finding (i.e., finding if a plan of length k exists for a given problem) can be usefully posed as a model finding problem, and solved using CSP, SAT or IP techniques (Kautz and Selman, 1996; Do and Kambhampati, 2001; Kautz and Selman, 1999; Bockmayr and Dimopoulos, 1999; Vossen et. al., 1999; Kautz and Walser, 1999).

These developments led to a significant new interest in the approaches to planning based on constraint programming. The three articles contributed to this special issue are a good example of such efforts. Together, they highlight several important research issues at the intersection of planning and CP.

Planning by Constraint Programming

Planning applications confront the CP community with several important modeling issues and trade-offs. An important characteristic of planning problems is that the length of the plan achieving a certain set of goals is not known *a priori*. This means that plan finding cannot be directly converted into a CSP instance. The usual way to finesse this is to look instead at the problem of "bounded-length" plan construction, which asks the question: "Is there a plan of length k for this problem?" A negative answer will necessitate construction of a new CSP encoding to find a $k+1$ length plan, and so on.

Not surprisingly, this has led to interest in conditional (dynamic) constraint satisfaction problems of the sort first introduced by Mittal and Falkenhainer for configuration problems (Mittal and Falkenhainer, 1990), and planning systems like CPlan (Van Beek and Chen, 1999). Several slightly more "hybrid" approaches have also been developed, e.g., modeling the planning problem in terms of structures such as planning graphs that are more natural to planning, and then compiling these structures into a SAT/CSP substrate (cf. (Kautz and Selman, 1999; Do and Kambhampati, 2001)).

An advanced approach will involve looking at CSP representations that automatically allow the introduction of new variables and new constraints. Work that completely captures planning within constraint programming thereby has been done in the context of the EXCALIBUR planning system (Nareyek, 2001), which makes use of so-called *structural constraints* to relax the restriction that a given constraint graph must be fixed a priori, and includes the search for a correct constraint graph as part of the optimization process. Frank and Jonsson's paper in the current issue proceeds in the same direction, but is more focused on specific interval elements. They introduce so-called *compatibilities* to define implicative relations among interval elements, which enables the existence of further intervals and constraints in specific situations to be required. During the search process, the CSP graph is dynamically changed to satisfy these compatibilities.

Using Constraint-based Solving Techniques within Planning

Many planning problems require the ability to handle complex logical, temporal and resource-oriented constraints. Constraint programming techniques can be used within a planning system to satisfy these constraints.

One important problem of planning in metric temporal domains is, for example, modeling and managing temporal constraints. This problem has received considerable attention from researchers in planning as well as CP, and has become a vigorous subfield in its own right (Dechter et al., 1991). The paper by Tsamardinos et al. in the current issue is an example of this trend. They discuss an extension of a temporal CSP called "conditional temporal CSP" that is relevant for representing conditional plans in metric temporal domains. In conditional planning, the interest is not in *one* possible plan but in a plan branching on different observation outcomes, which can only be perceived in the future.

Furthermore, given that modeling planning problems often requires mixed numeric, logical and temporal constraints, advances in the development of effective solution techniques to problems with mixed constraint types will be quite relevant to the planning applications. There is some promising work on solving coupled constraint systems (cf. LP-SAT (Wolfman and Weld, 1999), RealPlan (Srivastava et. al., 2001)), but many unresolved issues – including the handling of constraint propagation and optimization in coupled approaches – still remain.

Porting Solution Techniques

An interesting topic in recent research at the intersection of planning and CP has been whether to adapt ideas from the CP community to planning models (cf. (Kambhampati, 2000; Rintanen, 1998)) or to model the planning problem itself into a CP substrate (cf. (Van Beek and Chen, 1999; Bockmayr and Dimopoulos, 1999; Kautz and Walser, 1999; Kautz and Selman, 1996)) and solve it using standard CP techniques. The second approach has the apparent advantage that planning researchers can, to some extent, exploit the advances being made in CP solution techniques and benefit from the high expressiveness. The first approach, on the other hand, allows a much tighter coupling with the planning system, thus providing performance advantages.

For example, the mutex propagation used by the Graphplan algorithm, when viewed as a constraint propagation technique, corresponds to a specific form of partial and directional 3-consistency enforcement[1]. The power of the propagation procedure, however, seems to come from its directionality as well as its partial extent. In particular, mutex propagation (as well as memo discovery (Kambhampati, 2000)) is applied in a staged form and is limited to pairs of actions or pairs of propositions appearing at the same level in the planning graph structure. By contrast, when the problem is converted to a CP encoding, the distinction between literals (actions) belonging to different levels is lost.

A similar situation occurs when applying local search techniques to planning problems. Although local search has been applied both at the planning-graph level as well as the SAT/CSP-encoding level, the natural local moves differ substantially in both cases. The LPG planner, described in the article by Gerevini and Serina, carries out local search directly on the planning-graph representation. It significantly outperforms SATPLAN using GSAT, a state-of-the-art local-search technique for SAT problems. Indeed, LPG was the best performer in the AIPS 2002 planning competition.

We hope that readers will enjoy this issue and that it helps increase awareness of action planning domains among members of the constraint programming community. The field has great potential, both scientifically and from the application point of view, and we would be glad if this issue helped increase interaction between the constraint programming and action planning communities.

Finally, we wish to express our gratitude to the many reviewers who helped us to select the best papers and provided substantial feedback to the authors.

November 2002

Alexander Nareyek
Subbarao Kambhampati
(Guest Editors)

References

Blum, A. L. and Furst, M. L. Fast Planning Through Planning Graph Analysis. *Artificial Intelligence* 90: 281–300, 1997.

[1] Or a form of partial binary negative propagation in SAT terms (Kautz and Selman, 1999)

Bockmayr, A., and Dimopoulos, Y. Integer Programs and Valid Inequalities for Planning Problems. In Proceedings of the Fifth European Conference on Planning (ECP'99), 239–251, 1999.

Dechter, R., Meiri, I., and Pearl, J. Temporal Constraint Networks. *Artificial Intelligence* 49: 61–95, 1991.

Do, M., and Kambhampati, S. Planning as Constraint Satisfaction: Solving the Planning Graph by Compiling it into CSP. *Artificial Intelligence* 132(2): 151–182, 2001.

Kambhampati, S. Planning Graph as a (Dynamic) CSP: Exploiting EBL, DDB and other CSP Search Techniques in Graphplan. *Journal of Artificial Intelligence Research* 12: 1–34, 2000.

Kautz, H. A., and Walser, J. P. State-space Planning by Integer Optimization. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), 526–533, 1999.

Kautz, H. A., and Selman, B. Unifying SAT-based and Graph-based Planning. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), 318–325, 1999.

Kautz, H. A., and Selman, B. Pushing the Envelope: Planning, Propositional Logic and Stochastic Search. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), 1194–1201, 1996.

Laborie, P., and Ghallab, M. Planning with Sharable Resource Constraints. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 1643–1649, 1995.

Mittal, S., and Falkenhainer, B. Dynamic Constraint Satisfaction Problems. In Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), 25–32, 1990.

Nareyek, A. *Constraint-Based Agents – An Architecture for Constraint-Based Modeling and Local-Search-Based Reasoning for Planning and Scheduling in Open and Dynamic Worlds*. Reading, Springer LNAI 2062, 2001.

Rintanen, J. A Planning Algorithm not based on Directional Search. In Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), 617–625, 1998.

Stefik, M. J. Planning with Constraints (MOLGEN: Part 1). *Artificial Intelligence* 16(2): 111–140, 1981.

Srivastava, B.; Kambhampati, S.; and Do, M. B. Planning the project management way: Efficient planning by effective integration of causal and resource reasoning in RealPlan. *Artificial Intelligence* 131(1-2): 73–134, 2001.

Tate, A.; Drabble, B.; and Dalton, J. Reasoning with Constraints within O-Plan2. In Proceedings of ARPI Workshop, Morgan Kaufmann Publishers, 1994.

Van Beek, P. and Chen, X. CPlan: A Constraint Programming Approach to Planning. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), 585–590, 1999.

Vossen, T.; Ball, M.; Lotem, A.; and Nau, D. On the Use of Integer Programming Models in AI Planning. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), 304–309, 1999.

Wilkins, D. E. *Practical Planning — Extending the classical AI Planning Paradigm.* Reading, Morgan Kaufmann Publishers, 1988.

Wolfman, S. A., and Weld, D. S. The LPSAT Engine & its Application to Resource Planning. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), 310–316, 1999.