The Knowledge Engineering Review, Vol. 16:1, 2001, 1–4
Printed in the United Kingdom

On the intersection of AI and OR

CARLA P. GOMES

Computer Science Department, Cornell University, Ithaca, NY, USA
E-mail: gomes@cs.cornell.edu

This is the second of two special issues focusing on the integration of artificial intelligence (AI) and operations research (OR) techniques for solving hard computational problems, with an emphasis on planning and scheduling. Both the AI and the OR community have developed sophisticated techniques to tackle such challenging problems. OR has relied heavily on mathematical programming formulations such as integer and linear programming, while AI has developed constraint-based search techniques and inference methods. Recently, we have seen a convergence of ideas, drawing on the individual strengths of these paradigms. Furthermore, there is a great deal of overlap in research on local search and meta-heuristics by both communities.

In these two special issues, we compare and contrast AI and OR techniques. In the editorial of the first issue I discussed the main themes in OR and identified opportunities for the integration of AI and OR techniques focusing on three overarching themes: problem structure, duality and randomisation (Gomes 2000). In order to provide the readers with an overview of the content of the two special issues, I provide below a brief synopsis and discussion of each article.

Hooker *et al.* (2000) propose a framework for unifying OR-style optimisation and AI-style constraint-satisfaction methods. Their approach is based on exploiting the duality between search and inference, as well as the duality between the strengthening and relaxation of constraints. The paper provides detailed motivational examples illustrating how such dualities occur in the context of various branching schemes. The duality of search and inference is described in terms of a generalisation of the notion of duality for classic linear programming (LP), thereby providing a general method for a sensitivity analysis for constraint optimization. Hooker *et al.* also discuss an interesting connection between the notion of *nogoods* in constraint satisfaction and Benders decomposition in optimisation. (Intuitively speaking, a nogood identifies an inconsistent assignment to a subset of variables.) Finally, Hooker *et al.* propose as a particular promising research direction the combination of a constraint programming approach to inference with constraint-relaxation techniques from OR, such as LP relaxations.

Dixon and Ginsberg (2000) propose the integration of AI and OR techniques for solving propositional satisfiability problems, comparing and contrasting their relative strengths and weaknesses. The paper provides a good introduction to pseudo-Boolean representations and cutting plane proofs for the AI community, and to restricted learning methods, such as bounded learning, for the OR community. Dixon and Ginsberg also give a detailed description of the representation and inference methods adopted by both communities: in AI, conjunctive normal form and resolution and, in OR, pseudo-Boolean inequalities and cutting-planes techniques. They discuss how inference using cutting planes can be exponentially more efficient than inference using propositional logic. Dixon and Ginsberg present a cutting plane proof for the pigeonhole principle that is of size n^2. Nevertheless, to take advantage of the apparent efficiency of cutting-planes techniques for inference is a practical challenge because no effective general automatic techniques are known for generating cutting planes. However, certain heuristic approaches may be feasible. The paper also discusses the main search paradigms adopted by both communities: branch-and-bound with cuts is the method of choice in OR, while the Davis–Putnam–Loveland procedure, with unit-propagation and sophisticated branching, is the (complete) method of choice in AI. Nogood learning is analogous to the OR strategy of adding cuts to a branch-and-bound algorithm. Given the added power of the cutting-plane proofs and the effectiveness of nogood learning, Dixon and Ginsberg argue that the combination of cutting-plane proofs and relevance-bounded learning provides a promising research direction for further improvement of satisfiability procedures.

The remaining papers in the first issue focus on the integration of AI and OR techniques for planning and scheduling. The paper by Smith *et al.* (2000) compares and contrasts AI planning and scheduling methods, using as a motivational example a hypothetical spacecraft application. The example incorporates requirements from a wide range of actual NASA applications, ranging from space probes, such as the Deep Space One mission, to planetary rovers, such as the Mars Sojourner, to space-based observatories, such as the Hubble Space Telescope. Smith *et al.* provide an overview of AI classical planning and discuss the main plan search methods, such as forward state space search, goal-directed planning, Graphplan and SAT-based techniques. In addition to classical planning, the paper also describes hierarchical task network planning, and the more recent planning approaches based on Markov decision processes. Different representation formalisms for AI scheduling, based on constraint-satisfaction methods, are also discussed in the paper. One possible approach uses variables to represent start times, with constraints enforcing the order between tasks and the consumption of resources. Another representation scheme associates Boolean variables with ordered pairs of tasks, and constraints among variables enforcing additional ordering and resource requirements. Another approach discussed in the paper is scheduling as satisfiability. Finally, Smith *et al.* discuss the limitations of the current state of the art of planning and scheduling systems, using their spacecraft application as an example domain. They point out that such domains exhibit characteristics of both planning and scheduling problems: on the one hand, as in scheduling problems, the tasks involve complex temporal and resource constraints; on the other hand, as in planning problems, the applications involve choices of actions which affect the set of tasks to be performed and therefore the scheduling problem. Given the inherent hybrid nature of such problems, Smith *et al.* stress the need to bridge the gap between planning and scheduling techniques. In particular, they discuss the issues of how to handle resources, metric quantities and continuous time.

An interesting recent step towards integrating planning and scheduling techniques is based on the integration of linear and integer programming methods into AI planning systems. In the fourth article of the first issue, Vossen *et al.* (2000) provide different approaches for encoding AI planning problems as integer programming (IP) problems. They consider a SATPLAN-based IP formulation and an alternative formulation which they refer to as the "state change" formulation. The paper also discusses important issues that arise when using IP methods, such as the impact of the choice of IP formulation and the importance of preprocessing. They also discuss several research directions for exploiting IP formulations in combination with AI planning techniques.

In the final paper of the first issue, Kautz and Walser (2000) propose extending the planning as satisfiability framework to handle resource constraints, action costs and complex objective functions. They provide an IP-based formulation of the overall planning task. The resulting problems are very challenging and cannot be solved using pure branch-and-bound techniques. Kautz and Walser (2000) use an integer local search method to solve such instances with promising results.

The first article of the second special issue, by Puget and Lustig (2001), provides a good introduction to mathematical programming for the AI community, and an introduction to constraint programming for the OR community. The approaches are compared both from an algorithmic and from a modelling point of view. Three examples – a capacity warehouse problem, a resource allocation problem and a graph-colouring problem – are carefully worked out using both a constraint-programming and a mixed integer programming approach, as well as a hybrid approach, combining constraint propagation techniques with linear programming.

The paper by de Farias *et al.* (2001) focuses on problems that involve combinatorial constraints on continuous variables. Traditionally, in the mathematical programming community, such problems have been modelled as mixed-integer programs by introducing auxiliary binary variables and additional constraints. However, given the increase in number of variables and constraints, such an approach is only feasible for relatively small instances. Constraint-programming approaches, on the other hand, use the combinatorial structure of the problems but do not take advantage of the linear nature of the constraints to generate strong bounds. De Farias *et al.* propose a branch-and-cut approach *without* the introduction of binary variables. Instead, the combinatorial constraints are incorporated directly into the algorithm through the use of specialised branching and strong inequalities valid in the space of the continuous variables. They apply their approach to a generalised assignment problem that arises in the scheduling of fiber-optic cable manufacturing with promising results.

Caseau *et al.* (2001) exploit the combination of local search and global search techniques in a constraint-programming framework. The approach provides a novel perspective on the nature of constraint propagation as it results from the combination of several local and incremental processes associated with a particular constraint or a constraint class. They present several case studies – variants of job scheduling and real-world vehicle routing problems – that illustrate the differences between global search and local search and how one can use a constraint-programming framework, usually considered as a global search technique, to implement variants of local search. The key idea is that at each step, a given neighbourhood of a certain solution is searched using a global search approach. The various insights into the constraint-propagation process and the search techniques lead Caseau *et al.* to suggest a new direction for the development of a generic problem-independent approach for the automatic discovery of local search heuristics.

The focus of the paper by Westfold and Smith (2001) is on the generation of efficient constraint satisfaction programs. They use KIDS (Kestrel Interactive Development System), a framework that allows one to start with a high-level problem representation and obtain a highly optimised implementation through the application of a sequence of automatic refinement steps. The constraint-satisfaction algorithms adopted in KIDS are based on a global search strategy, combined with pruning mechanisms at each node of the search tree. The methodology proposed in the paper is illustrated in detail, in particular how constraints are reformulated, how to generate splitting code and how to specialise the constraint-propagation code.

In the final paper of this special issue, Wolfman and Weld (2001) propose a promising hybrid approach for solving AI resource planning problems, by combining linear programming and satisfiability techniques. They introduce a language, LCNF (Linear Conjunctive Normal Form), which combines propositional logic with linear constraints, and a solver, LPSAT. The LPSAT solver is based on the Davis–Putnam–Loveland (DPL) backtrack search procedure enhanced with an incremental LP solver to check the consistency of the linear constraints. The LPSAT engine is implemented in RelSAT. RelSat is a DPL solver which incorporates learning mechanisms for identifying conflict sets (nogoods). LPSAT takes advantage of this feature of RelSAT, combined with an added feature for the discovery of the corresponding conflict sets in terms of linear constraints, to perform a highly effective backtrack search.

I would like to thank all authors for their contributions, and for making these two special issues a comprehensive and highly interesting guide to the rich research domain that lies at the interface between AI and OR.

References

Caseau, Y, Laburthe, F, Le Pape, C and Rottembourg B, 2001, "Combining local and global search in a constraint-programming environment" *Knowledge Engineering Review* **16**(2) 41–68.

De Farias, I, Johnson, E and Nemhauser, G, 2001, "Branch-and-cut for combinatorial optimisation problems without auxiliary binary variables" *Knowledge Engineering Review* **16**(2) 25–39.

Dixon, H and Ginsberg, M, 2000, "Combining satisfiability techniques from AI and OR" *Knowledge Engineering Review* **15**(1) 31–47.

Gomes, C, 2000, "Artificial intelligence and operations research: challenges and opportunities in planning and scheduling" *Knowledge Engineering Review* **15**(1) 1–11.

Hooker, J, Ottosson, G, Thorsteinsson, E and Kim, H, 2000, "A scheme for unifying optimisation and constraint-satisfaction methods" *Knowledge Engineering Review* **15**(1) 11–31.

Kautz, H and Walser, J, 2000, "Integer optimisation models of AI planning problems" *Knowledge Engineering Review* **15**(1) 101–117.

Puget, J and Lustig, I, 2001, “Constraint programming and maths programming” *Knowledge Engineering Review* **16**(2) 5–23.
Smith, D, Frank, J and Jonsson A, 2000, “Bridging the gap between planning and scheduling” *Knowledge Engineering Review* **15**(1) 47–85.
Westfold, S and Smith, D, 2001, “Synthesis of efficient constraint-satisfaction programs” *Knowledge Engineering Review* **16**(2) 69–84.
Wolfman, S and Weld, D, 2001, “Combining linear programming and satisfiability-solving for resource planning” *Knowledge Engineering Review* **16**(2) 85–99.
Vossen, T, Ball, M, Lotem, A and Nau, D, 2000, “Applying integer-programming techniques to AI planning” *Knowledge Engineering Review* **15**(1) 85–101.