ELSEVIER

Artificial Intelligence 111 (1999) 329–338

Artificial Intelligence

www.elsevier.com/locate/artint

Book Review

Intelligent planning: A decomposition and abstraction based approach to classical planning ☆

Fausto Giunchiglia [a,b,*], Luca Spalazzi [c,1]

[a] *ITC-IRST, via Sommarive, 38050 Povo, Trento, Italy*

[b] *DISA, University of Trento, Via Inama, 38100 Trento, Italy*

[c] *Istituto di Informatica, University of Ancona, Via Brecce Bianche, 60131 Ancona, Italy*

1. Introduction

This is a review of the book *Intelligent Planning*: *A Decomposition and Abstraction Based Approach* by Qiang Yang [45], a book focused on classical planning. In recent years, the classical approach to planning has been challenged by many authors (e.g., [1,2,8,11,17,18]). The question is whether its fundamental assumptions and complexity make it of any use in real world applications. The arguments against classical planning (e.g., reactive planning [7,26]) are valid in a number of domains. Nevertheless, in our opinion, there are applications where classical planning is still the only reasonable approach, e.g., logistics, process planning, scheduling. Furthermore, in the last few years, many new approaches have been proposed which are elaborations of classical planning, [2] e.g., case-based planning [21–23,31,43], multi-agent planning [13,15,25], non-STRIPS style planning [6,29], and model checking based planning [12].

Yang's book provides a set of basic techniques for generating plans and a set of formal and empirical tools for evaluating and comparing them. The book is divided in three parts. The first part is devoted to the fundamentals of planning. Its title is *Representation, Basic Algorithms, and Analytical Techniques*. The second describes the "divide et impera" approach and is entitled *Problem Decomposition and Solution Combination*. The third, *Hierarchical Abstraction*, is devoted to the use of abstraction in planning.

☆ Qiang Yang, Intelligent Planning: A Decomposition and Abstraction Based Approach, Springer, Berlin, 1997. 264 + xv pp. ISBN 3-S40-61901-1. Hardcover: $ 49.95.

* Corresponding author. Email: fausto@irst.itc.it.

1 Email: spalazzi@inform.unian.it.

2 In various parts Yang hints at the relation between classical planning and the other approaches (see, for example, pp. 95 and 112).

PII: S0004-3702(99)00042-9

2. Classical planning

> "Classical AI planning is concerned mainly with the generation of plans to achieve a set of pre-defined goals in situations where most relevant conditions in the outside world are known, and where the plan's success is not affected by changes in the outside world."
>
> (p. VIII).

This is the definition of classical planning according to Qiang Yang. Let us analyse it in detail.

Generation of plans: A planning system can be described in terms of a set of basic activities, namely: plan generation (i.e., the activity of producing a plan), plan execution (i.e., the execution of plan actions), acting (i.e., the activation and control of actuators), sensing (i.e., the acquisition of information from the outside world), failure handling (i.e., the recovery from failures), and so on [42]. It is possible to obtain different planning systems depending on how these activities are combined. Classical planning [10,14,38,44] is concerned only with the generation of plans. Plan generation can be based on state space search [14], partial plan space search [44], theorem proving [5], or case memory search [20]. Yang focuses his attention on state and partial plan space search.

Set of pre-defined goals: A planning system can generate a plan in order to either achieve a goal or react to an external event. The former case is the realm of classical planning; the latter that of reactive planning. According to its aim, Yang's book focuses on how to achieve goals. In general, a goal is a condition on intermediate and final states of actions, i.e., a set of desired behaviours (e.g., [16]). Nevertheless, in classical planning two simplifying assumptions are usually made, namely, that a goal is a condition on final states (see, for example, [5,14]) and that this condition is a conjunction of clauses (see, for example, [10,14]). Both these assumptions are also made in Yang's book. [3]

Outside world: Classical planning makes some fundamental assumptions about the world, namely: the initial situation is known; all the effects of the actions are known; and the world does not change or the changes are negligible. Two main consequences arise from these assumptions. First, if we assume complete knowledge of the world, then we must deal with problems such as the qualification problem, the ramification problem, and the frame problem. Classical planning usually solves these problems by making the closed-world assumption: i.e., "only what is explicitly stated holds". This solution is adopted also in Yang's book. The other consequence is that classical planning can be applied only to domains which are not dynamic and unpredictable. This cuts off applications such as robotics, network navigation, and so on, [4] and limits somewhat the scope of the book.

Goal achievement and plan success: Goal achievement and plan success are in general two different notions. Plan success is related to the proper termination of a plan execution: a plan succeeds if it does not abort. Goal achievement is related to the satisfaction of some conditions on goal states. A plan may fail and, nevertheless, it may achieve its goal(s). For

[3] Yang makes an exception when he discusses SATPLAN (see Chapter 2, p. 34).

[4] The above assumptions can be partially relaxed by introducing the notion of replanning. We have replanning when, if a plan execution fails, then a new plan is generated and executed [44].

example, suppose you have the plan which moves a robot to position A. It may have been generated to achieve the goal of having the robot in the region A' which contains A. If the plan execution aborts when the robot is inside A' but not at A, the plan fails but the goal is achieved. In classical planning, the notion of plan success is usually assumed to coincide with the notion of goal achievement (this is a consequence of the fact that classical planning does not deal with plan execution and monitoring). Yang also makes this assumption.

Starting from the definition of classical planning, as it emerges from the above discussion, we can summarise Yang's view on classical planning as follows:

(i) "Plan generation can be viewed as a search in a space of nodes."

(p. 39).

Since classical planning deals with plan generation, it can be considered as a search problem. Indeed most of the material which can be found in the literature (and also in the book) is focused on improving search algorithms.

(ii) "The issue of efficiency is of major concern [here] because of the potentially large number of action combinations as plans."

(p. 2).

Efficiency is the parameter that takes into account the amount of resources (e.g., time and memory) needed in order to solve a problem. Again, efficiency has always been an important issue in classical planning.

(iii) "An equally important issue is the quality of a plan generated."

(p. 2).

This point boils down to an evaluation of the effectiveness of a given algorithm. We know that classical planning produces plans that work well in application domains which are not dynamic, unpredictable, and real-time. Furthermore, sometimes classical planning algorithms generate alternative plans for the same goal. In this situation it may be necessary to find the plan with the lowest cost (whatever cost means). A related issue is how large is the class of problems solved by an algorithm.

(iv) "The central thesis of this book is that to be effective, a planning algorithm must go beyond a basic one."

(p. 2).

Each planning algorithm works well for a class of problems. By combining different algorithms we can solve a wider class of problems.

3. Evaluation criteria

Following Yang's view of classical planning, we have reviewed the book according to four different points of view.

- *Algorithms*. Classical planning is research on algorithms; the more algorithms there are, the more problems can be solved.
- *Examples*. Examples are useful for understanding how a planning algorithm works, and in which domains. It is essential that new ideas are explained by clear examples. Examples must also give intuitions about how things work in practice.
- *Theory*. Soundness, completeness, and optimality are important formal properties related to effectiveness. Roughly speaking, to be sound, an algorithm has to produce plans which do the right thing and, to be complete, it must be able to produce all the

correct plans. Algorithm complexity is an important property as well. It is related to efficiency and it allows us to study how resources grow with the problem size.
- *Empirical evaluation*. From a practical perspective, empirical properties can be more important than formal properties. They provide an evaluation of the real effectiveness and efficiency of an algorithm. Empirical properties are usually estimated through experiments. The more realistic the testbed is, the better the evaluation is.

Another important aspect we have considered is the intended audience of Yang's book. In the preface, Yang explicitly mentions the people he wants to address, namely:
- *Students*. The book could be used as a textbook for undergraduate and graduate students in Artificial Intelligence and Computer Science. The author suggests its use in a one-semester or one-quarter AI course.
- *Researchers*. The book is also directed to researchers in Artificial Intelligence and other related fields. In this case, the idea is to use it as a reference to the planning literature.

4. Algorithms

Algorithms are the crucial and most important part of the book. In the first part Yang presents a set of basic search algorithms. Chapter 4 presents a set of useful basic computer science algorithms. It contains, for instance, algorithms on graphs, dynamic programming, branch-and-bound, constraint satisfaction algorithms, and GSAT. These algorithms are used in the rest of the book in order to solve specific problems which can arise in planning. Chapters 2 and 5 contain a set of basic planning algorithms. The author classifies them into systematic search and local search algorithms. The former algorithm starts from an empty solution and systematically explores the space of solutions. A local search algorithm starts from a randomly generated solution and proceeds by "repairing" the conflicts and errors of the solution itself. The author further classifies the planning algorithms in total-order and partial-order depending on the ordering relation among plan actions. Among all the possible algorithms, the author proposes two partial-order backward-chaining algorithms (POPLAN—the plain version—and the version capable to reason about resources), a total-order forward-chaining algorithm (TOPLAN), and a total-order GSAT-based algorithm (SATPLAN). The first three algorithms proceed systematically, the last is an example of local search in planning.

The part devoted to the "divide et impera" paradigm provides the basic algorithm of decomposition-based planning (Chapter 6). This algorithm is presented so that, for each sub-problem, it can use any of the algorithms described in Part I. The main goal of this part is to present algorithms for decomposing a problem, for selecting a solution from a set of alternatives, for resolving conflicts among solutions, and for merging plans. Chapter 6 concentrates on goal-directed decomposition. According to this technique, the set of goals is split into smaller sub-sets which are then analyzed separately. Chapter 7 is devoted to conflict resolution. "Conflicts rarely exist alone. Therefore, conflict-resolution decisions should not be made in a local manner." (p. 101). Yang proposes to solve this problem by constraint satisfaction. Plan merging is reduced to an optimization problem (see Chapter 8). Indeed the solution of single sub-problems can produce redundancy. This redundancy must be removed during plan merging in order to generate a plan that is less costly to execute. The author proposes both an optimal and a sub-optimal merging algorithm. The former is based on dynamic programming while the latter is based on a greedy algorithm. Chapter 9 deals with the problem of selecting plans in the case of multiple solutions. Sometimes it is sufficient to select a solution which is compatible with the others. In such a case, plan selection is still a constraint satisfaction problem. Sometimes it is important to minimize the total cost of the merged plan. This situation requires an optimization-based method. Yang provides a branch-and-bound algorithm.

The final part of the book is devoted to hierarchical abstraction. The basic hierarchical planner (HIPLAN) is explained in Chapter 10. Its fundamental step is *plan refinement*, i.e., the task of increasing the details of a plan at a certain level in order to obtain a plan at a lower level. Yang's book provides three different plan refinement methods: precondition-elimination abstraction, task abstraction, and effect abstraction. In precondition elimination the simpler problem is obtained by considering (operators with) fewer preconditions (Chapter 10). The author provides both a total-order forward chaining and a partial-order backward chaining algorithm. Chapter 11 is devoted to the presentation of an algorithm which automatically generates an operator hierarchy. In task abstraction, planning problems and operators are called tasks. Each task is reduced to a set of sub-tasks according to a predefined reduction schemata (Chapter 12). Chapter 12 also gives an algorithm which automatically checks the properties of the reduction schemata. In effect abstraction the simpler problem is obtained by considering only the primary effects of operators. The side effects are computed during the refinement process: in Chapter 13, the problem of generating a hierarchy of effects is solved by proposing an inductive learning algorithm.

The extensive collection of algorithms is, in our opinion, one of the main values of Yang's book. Some are by the author himself. Yang's effort to expose the material from a unified perspective gives added value to the presentation. For example, in Part III, different planning approaches receive a unified presentation by means of abstraction. Indeed the three approaches considered in Part III are proposed as different ways of performing the plan refinement step of HIPLAN.

In addition, for each algorithm, Yang provides an insightful discussion about advantages, limitations, and a wide reference list. The book also contains a description of the open problems (a detailed list is reported in each chapter) and pointers to the most recent papers trying to go beyond the current limitations. All of this makes this book a really useful tool for researchers in AI and in other fields related to planning. A typical example is the algorithm for goal-directed decomposition (Chapter 6). The algorithm is explained step by step. Besides this, Yang provides a list of benefits of goal decomposition and a review of alternative approaches to problem decomposition.

5. Examples

The algorithms and the theoretical analysis are explained through examples. Most of the examples are extracted from the so called "painting domain". This domain is introduced in Chapter 1 and is used in almost all the book. Yang describes it as follows: "A robot is given the task of painting both a ceiling and a ladder.” (p. 1). For example, Yang uses this domain to illustrate the partial-order backward-chaining planning both in the plain version (see Chapter 2) and in the version that reason about resources (see Chapter 5). Besides the painting example, the author employs further domains. These range from toy examples to simple real world domains. The toy domains are examples such as the variable-sized blocks world and the tower of Hanoi. The variable-sized blocks world is used to explain the reasoning about resources (Chapter 5) and plan merging (Chapter 8). The tower of Hanoi is used to illustrate abstraction hierarchies (see Chapter 10). More realistic domains such as logistics, process planning, query planning, and common sense planning are considered in the experiments and in the critical discussion about the applicability of planning.

Examples could also be useful to evaluate for which problems the planning techniques presented in the book can be exploited. However, from this point of view, the examples presented in the book are sometimes idealized or simply sketched. In these situations, it is not obvious to see how the proposed algorithms can be applied to more concrete applications. This is the case with SATPLAN (Chapter 2). The author explains with an example how a domain model is built and how a plan is obtained from a sequence of states, but he leaves out how the sequence of states is obtained from the domain model. The situation does not improve in Chapter 4 when the author explains GSAT, an algorithm which plays a crucial role in SATPLAN. A reader (especially a student) would benefit a lot from an example covering all the steps of the algorithm.

6. Theory

Yang’s book employs the STRIPS representation [14]. This is extended only when necessary. The author starts from this representation and, for each algorithm, analyzes formal properties such as correctness, completeness, optimality, and computational complexity. Let us consider computational complexity. Its fundamentals are given in Chapter 3. This chapter is also devoted to the analysis of the basic planning algorithms. It contains a comparison between total-order and partial order planning. From this comparison we get the fundamental results that “... neither [approach] is a clear winner. While TOPLAN has a potentially large branching factor, the search depth is fixed at N, the total number of operators in an optimal plan. POPLAN, on the other hand, has a depth that is P times larger than that of TOPLAN, where P is the maximum number of preconditions for a plan step. In addition, although POPLAN has a bounded branching factor, [...] when the number of preconditions for each plan step and the threats are large, the effective branching factor for partial-order planning could also be extremely large!” (p. 46).

The analysis becomes more interesting in the second and third parts of the book. In the second part Yang compares decomposition based algorithms with those that do not use decomposition (see Chapter 6). Yang extends Korf’s analysis on decomposition [33] in two directions. First, he considers the possibility of interacting goals. From this analysis, it turns out that “decomposition no longer has definite advantages. For decomposition to win, the numbers of operator-overlapping interactions and deleted-condition interactions must be small.” [5] (p. 90). Second, Yang considers decomposition in partial-order backward chaining planning. He reaches similar conclusions. “For decomposition to win, the numbers of [...] interactions must still be small.” (p. 91). These results suggest that “interactions play a central role in determining whether a decomposition is good” (p. 92). Chapter 8 deals with plan merging and contains a comparison between optimal and approximate methods. From the point of view of efficiency, the approximate algorithm is better than the optimal one (the complexity is linear instead of polynomial). On the other hand, concerning effectiveness, an approximate algorithm generates a plan that costs slightly more than the plan generated by the optimal algorithm (see Theorems 8.5.1 and 8.5.2). In the third part of the book, the author extends his complexity analysis to abstraction and compares it with his previous analysis (see Chapter 10). In addition, he considers the formal properties of the hierarchies (of operators, tasks, and effects) related to effectiveness. The author studies if an abstract solution can guarantee the existence of more (less) abstract solutions. He also studies the properties of finding all the solutions and solutions close to the optimal one. These properties are exploited to generate hierarchies which are useful in order to reduce the planning complexity and preserve all the possible solutions.

Anybody approaching planning for the first time needs a theoretical perspective for a proper evaluation of the most important planning algorithms. The theory reported in Yang’s book gives, in our opinion, this perspective. For instance, we agree with the choice of using the STRIPS representation. This choice is justified by its vast popularity [9,22,23, 36,37,39], by its clear and well-founded semantics [35], and by the fact that most of the more sophisticated representations can be easily derived from it [11,27,41]. Of course, this leaves out the theoretical analysis of non-STRIPS style planning.

Finally, it is worth noticing the importance, within the book, of the valuable original contributions of the author. This is the case, for example, of the complexity analysis of different partial-order planning algorithms (e.g., [28,30]). These complexity results substantially integrate the previous work in the literature. With these results, the complexity analysis of STRIPS representation based classical planning has, in our opinion, a quite complete and unified presentation.

7. Empirical evaluation

The author makes empirical evaluations of most of the algorithms described in the book. This allows him to draw various conclusions. One example concerns complexity. From a theoretical point of view, planning problems are usually intractable, but some classes of problems may be tractable in practice. One such example is given in Chapter 9 for optimization-based plan selection. From a theoretical perspective this problem is NP-hard. Nevertheless, as the book reports, Nau and Yang experimented with the proposed algorithm in a manufacturing planning domain and verified that: “The performance of the algorithm is quite good—especially since the test problem was chosen to be significantly more

[5] An operator-overlapping (delete-condition) interaction occurs when an operator relevant for achieving a goal adds (deletes) the precondition of an operator relevant for another goal.

difficult than the kind of problem that would arise in real-world process planning." (p. 155). Another consideration concerns the utility of constraint satisfaction problem solvers in planning. Generally speaking, this is still an open question. Nevertheless, in the case of reasoning about resources in planning, for some simple domains such as the variable-sized blocks world, the author is able to propose some criteria for deciding when this approach is convenient (see Chapter 5). This suggests how to obtain similar criteria for different application domains.

Finally, some algorithms have also been compared with other algorithms extracted from the literature, e.g., TWEAK and ALPINE. In Chapter 9, TWEAK is compared (in the blocks world domain) with the proposed algorithm for constraint-based plan selection (WATPLAN). "This test [...] demonstrates that with WATPLAN the computational cost of the combination phase is much lower than TWEAK." (p. 146). Finally, in Chapter 11, the robot-box domain is used to compare ALPINE with the algorithm for generating a precondition-elimination hierarchy reported in the book (called HIPOINT). HIPOINT may be slower than ALPINE in small size problems, but it is almost four times faster for large size problems.

Empirical evaluations are useful, especially for problems which are untractable from a theoretical point of view. Nevertheless, in general, a further step is needed in order to understand what constitutes the hard planning problems. For instance, this is what has been done for GSAT, when it was discovered that some problems with certain proposition/clause ratio are the hardest to solve [19,40]. Yang's book does not deal with this problem. However this is not a defect of the book, but of the current state of the art in planning.

8. Conclusions

Several areas of Artificial Intelligence have been adequately covered by monographs, e.g., case-based reasoning [32], natural language [4], and machine learning [34]. Concerning planning, we can find two kinds of books: books whose topic is the presentation of specific systems (e.g., [39,44]) and readings (e.g., [3,24]). In both cases there is no effort on providing a general and/or theoretical framework of planning. Yang's book fills this gap. It is a comprehensive monograph on classical planning and gives the added value of providing a unified theoretical framework as well as empirical evaluations. The best part of the book lies in the two original perspectives from whence the classical planning is depicted: "divide et impera" and abstraction. The coverage of basic principles of computer science is very important as well.

The book is directed to students, researchers in AI and other disciplines. AI researchers will value the extensive collection of algorithms presented in this book, the complete reference list, and the critical discussion of advantages, limitations, and open problems. They may find the theoretical analysis and empirical evaluations very useful, as well. In practice, they may use the text as a planning handbook. The people who will most value this book are students, researchers in related fields, and anyone else who wants to learn about planning. These people may learn about planning methodologies within a unified formal framework. The book is self-contained. The only prerequisites are some fundamentals of mathematics and computer science, and some programming skills. Yang intended to write a textbook and indeed he did a great job.

One criticism is that the book would have been even better with some exercises, which are absent in the book.

References

[1] P. Agre, D. Chapman, Pengi: An implementation of a theory of activity, in: Proc. AAAI-87, Seattle, WA, 1987, pp. 268–272.
[2] P. Agre, D. Chapman, What are plans for?, Technical Report AI Memo 1050a, AI Laboratory, MIT, Cambridge, MA, 1989.
[3] J. Allen, H. Kautz, R. Pelavin, J. Tenenberg (Eds.), Reasoning about Plans, Morgan Kaufmann, San Mateo, CA, 1991.
[4] J.F. Allen, Natural Language Understanding, Benjamin/Cummings, Menlo Park, CA, 1987.
[5] S. Biundo, D. Dengler, J. Köhler, Deductive planning and plan reuse in a command language environment, in: Proc. 10th European Conference on Artificial Intelligence, Vienna, Austria, 1992, pp. 628–632.
[6] A. Blum, M.L. Furst, Fast planning through planning graph analysis, in: Proc. IJCAI-95, Montreal, Quebec, Morgan Kaufmann, San Mateo, CA, 1995, pp. 1636–1642.
[7] R.A. Brooks, A robust layered control system for a mobile robot, IEEE J. Robotics and Automation RA-2 (1) (1986) 14–23.
[8] R.A. Brooks, Intelligence without representation, Artificial Intelligence 47 (1991) 139–160.
[9] T. Bylander, Complexity results for planning, in: Proc. IJCAI-91, Sydney, Australia, Morgan Kaufmann, San Mateo, CA, 1991, pp. 274–279.
[10] D. Chapman, Planning for conjunctive goals, Artificial Intelligence 32 (1987) 333–377.
[11] D. Chapman, Penguins can make cake, AI Magazine 10 (4) (1989) 45–50.
[12] A. Cimatti, E. Giunchiglia, F. Giunchiglia, P. Traverso, Planning via model checking: A decision procedure for $\mathcal{AR}$, in: Proc. 4th European Conference in Planning (ECP-97), Toulouse, France, 1997.
[13] E.H. Durfee, Coordination of Distributed Problem Solvers, Kluwer Academic, Dordrecht, Netherlands, 1988.
[14] R.E. Fikes, N.J. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (3–4) (1971) 189–208.
[15] M.R. Genesereth, S.P. Ketchpel, Software agents, Comm. ACM 37 (7) (1994) 48–53.
[16] M. Georgeff, A.L. Lansky, Procedural knowledge, Proc. IEEE 74 (10) (1986) 1383–1398.
[17] M.P. Georgeff, Situated reasoning and rational behaviour, Technical Report 21, Australian AI Institute, Carlton, Victoria, Australia, 1991.
[18] M.L. Ginsberg, Universal planning: An (almost) universally bad idea, AI Magazine 10 (4) (1989) 40–44.
[19] F. Giunchiglia, M. Roveri, R. Sebastiani, A new method for testing decision procedures in modal and terminological logics, in: Proc. 1996 International Workshop on Description Logics (DL-96), Cambridge, MA, 1996.
[20] K.J. Hammond, Explaining and repairing plans that fail, Artificial Intelligence 45 (1–2) (1990) 173–228.
[21] K.J. Hammond, Case-based planning: A framework for planning from experience, Cognitive Science 14 (3) (1990) 385–443.
[22] S. Hanks, D. Weld, A domain-independent algorithm for plan adaptation, J. Artificial Intelligence Res. 2 (1995) 319–360.
[23] L.H. Ihrig, S. Kambhampati, Storing and indexing plan derivations through explanation-based analysis of retrieval failures, J. Artificial Intelligence Res. 7 (1997) 161–198.
[24] J. Hendler, J. Allen, A. Tate (Eds.), Reading in Planning, Morgan Kaufmann, San Mateo, CA, 1990.
[25] N.R. Jennings, M. Wooldridge, Application of intelligent agents, in: N.R. Jennings, M. Wooldridge (Eds.), Agent Technology: Foundations, Applications, and Markets, Springer, Berlin, 1998, pp. 3–28.
[26] L.P. Kaelbling, An architecture for intelligent reactive systems, in: Reasoning about Actions and Plans, Proc. 1986 Workshop, Morgan Kaufmann, San Mateo, CA, 1987.

[27] S. Kambhampati, J. Hendler, A validation-structure-based theory of plan modification and reuse, Artificial Intelligence 55 (1992) 193–258.
[28] S. Kambhampati, C.A. Knoblock, Q. Yang, Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning, Artificial Intelligence 75 (3) (1995) 167–238. Special Issue on Planning and Scheduling, Edited by J. Hendler and D. McDermott.
[29] H. Kautz, B. Selman, Pushing the envelope: Planning, propositional logic and stochastic search, in: Proc. AAAI-96, Portland, OR, AAAI Press / MIT Press, Menlo Park, CA, 1996, pp. 1194–1201.
[30] C.A. Knoblock, Q. Yang, Evaluating the tradeoffs in partial-order planning algorithms, in: Proc. Canadian Artificial Intelligence Conference (AI-94), 1994, pp. 279–286.
[31] J. Koehler, Planning from second principles, Artificial Intelligence 87 (1996) 145–186.
[32] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, San Mateo, CA, 1993.
[33] R. Korf, Planning as search: A quantitative approach, Artificial Intelligence 33 (1985) 65–68.
[34] P. Langley, Elements of Machine Learning, Morgan Kaufmann, San Francisco, CA, 1995.
[35] V. Lifschitz, On the semantics of strips, in: M. Georgeff, A. Lansky (Eds.), Reasoning about Actions and Plans, Proc. 1986 Workshop, Morgan Kaufmann, San Mateo, CA, 1986, pp. 1–9.
[36] B. Nebel, J. Koehler, Plan reuse versus plan generation: A theoretical and empirical analysis, Artificial Intelligence 76 (1–2) (1995) 427–454.
[37] N.J. Nilsson, Principles of Artificial Intelligence, Tioga Publishing, Los Altos, CA, 1980.
[38] E.D. Sacerdoti, Planning in a hierarchy of abstraction spaces, in: Proc. 3rd International Joint Conference on Artificial Intelligence (IJCAI-73), Stanford, CA, 1973.
[39] E.D. Sacerdoti, A Structure for Plans and Behaviour, Elsevier/North-Holland, Amsterdam, 1977.
[40] B. Selman, H. Levesque, D. Mitchell, Hard and easy distributions of SAT problems, in: Proc. AAAI-92, San Jose, CA, 1992, pp. 440–446.
[41] J.D. Tenenberg, Abstraction in planning, Ph.D. Thesis, Computer Science Department, University of Rochester, 1988. Also TR 250.
[42] P. Traverso, A. Cimatti, L. Spalazzi, Beyond the single planning paradigm: Introspective planning, in: Proc. ECAI-92, Vienna, Austria, 1992, pp. 643–647.
[43] M.M. Veloso, H. Muñoz-Avila, R. Bergmann, Case-baed planning: Selected methods and systems, AI Communications 9 (3) (1996) 128–137.
[44] D.E. Wilkins, Practical Planning: Extending the Classical AI Planning Paradigm, Morgan Kaufmann, San Mateo, CA, 1988.
[45] Q. Yang, Intelligent Planning: A Decomposition and Abstraction Based Approach, Springer, Berlin, 1997.