Combining Macro-Operators with Control Knowledge

Rocío García-Durán, Fernando Fernández and Daniel Borrajo

Universidad Carlos III de Madrid
Avda de la Universidad 30, 28911-Leganés (Madrid), Spain
rgduran@inf.uc3m.es, ffernand@inf.uc3m.es, dborrajo@ia.uc3m.es

Abstract. Inductive Logic Programming (ILP) methods have proven to succesfully acquire knowledge in very different learning paradigms, such as supervised and unsupervised learning or relational reinforcement learning. However, very little has been done on General Problem Solving (GPS). One of the ILP-based approaches applied to GPS is HAMLET. This method is able to learn control rules (heuristics) for a non linear planner, PRODIGY4.0, which is integrated into the IPSS system; control rules are used as an effective guide when building the planning search tree. Other learning approaches applied to planning generate macro-operators, building high-level blocks of actions, but increasing the branching factor of the search tree. In this paper, we focus on integrating the two different learning approaches (HAMLET and macro-operators learning), to improve a planning process. The goal is to learn control rules that decide when to use the macro-operators. This process is succesfully applied in several classical planning domains.

1 Introduction

Planning is a problem solving task that consists on given a domain theory (set of states and operators) and a problem (initial state and set of goals), obtain a plan (set of operators and an partial order of execution among them), such that, when executed, transforms the initial state in a state where all goals are achieved. Planning is computationally hard (PSPACE). To reduce the difficulty of finding a solution to a problem, many solvers employ learning techniques, that improve noticeably the original behaviour of the solvers.

In this paper, we propose to use two learning techniques: macro-operators and control rules. Both of them have been applied under the integrated tool for planning and scheduling IPSS [1].

Macro-operators are compositions of simpler operators, and generate the same result than executing the simple operators sequentially. Their main advantage is the decrease of the depth in the search tree, reducing the number of nodes and the number of variables to bind; that is memory and time. However, their drawback is the utility problem [2]. The addition of macro-operators in the domain description increases the branching factor and the processing cost per node, which can worsen search performance.

The second learning approach obtains control rules from search episodes. It is based on a relational approach that combines analytical (EBL) and inductive techniques. As most relational learning systems, it learns by generalizing and specializing a set of rules [3].

In this paper we use HAMLET to find a set of control rules that are able to decide when to use the acquired macro-operators. Using control rules, IPSS reduces the number of nodes of the search tree expanded and, thus, the planning time. A very similar combination was used in [4].

2 Combining two learning techniques

With the aim of generating control rules that define when a specific macro-operator shall be used, we have used both techniques together according to the following steps:

1. The first step is to generate and select some macro-operators composed by two and three simple operators. We provide IPSS a set of random training problems to be solved. From the resulting total-ordered plans, all the different combinations of two and three operators are obtained. These operators must appear in sequence and have, at least, one object in common. The most common sequences are selected for the second step. Another technique to generate macro-operators can be found in [5].
2. The next step is: we let the system learn control rules for each macro-operator separately, using always the same training set of random problems. We also learn control rules using the original domain (without macro-operators) for comparison.
3. Finally, the same test set is used for each resulting domain: (i) the original domain, (ii) the domain with each selected macro-operator, (iii) the original domain with its own learned control rules, and (iv) the macro-operators and the control rules together.

3 Experiments

Table 3 describes the experiments performed in four classical planning domains: Logistics, Miconic, Zenotravel and Blocksworld (from the International Planing Competition). The macro-operators are represented in Table 3 by m2-1, m2-2 and m2-3 to refer to the three more common macro-operators composed by two simple operators and by m3-1, m3-2 and m3-3 for the three more common macro-operators composed by three simple operators. For instance, macro-operator m2-1 in the Zenotravel domain is: *refuel+fly*. Also, the percentages of solved problems with the original domain is represented in Table 3 by the first row and the IPSS column of each domain. The percentages for each selected macro-operator are represented by the first row of results and the HAMLET column. The results with the original domain and its own learned control rules are shown in the first results row of the table and the IPSS column. Finally, the percentages of solved problems with macro-operators and control rules together are represented in the rows with the specific macro-operator and the HAMLET column.

The columns named *Rule* respresents the number of control rules obtained by HAMLET.

Analysing the results, we can observe that all macro-operators that solved less percentage of problem than IPSS (original domain), never improves the results when adding control rules. However, all macro-operators that improve the performance in the original domain, also improve the results in the original domain with control rules. The only exception is macro m3-1 in the Miconic domain. Besides, using control rules with these good macro-operators, the performance improves in most of the cases. Only macro m2-1 in the Miconic and Blocksworld domains is worse than without control rules. However, m2-2 and m3-3 in Logistics domain, have noticeably improved the results.

Domain	LOGISTICS			MICONIC			ZENOTRAVEL			BLOCKSWORLD		
	IPSS	HAMLET		IPSS	HAMLET		IPSS	HAMLET		IPSS	HAMLET	
	Solved	**Solved**	**Rule**	**Solved**	**Solved**	**Rule**	**Solved**	**Solved**	**Rule**	**Solved**	**Solved**	**Rule**
D	10%	11%	9	3%	10%	2	0%	0%	3	43%	46%	5
D + m2-1	6%	9%	4	**16%**	15%	4	**86%**	**87%**	2	**74%**	**70%**	8
D + m2-2	26%	**36%**	8	1%	1%	3	3%	3%	0	41%	43%	6
D + m2-3	3%	3%	6	11%	10%	3	42%	42%	0	28%	29%	3
D + m3-1	6%	9%	4	4%	7%	3	8%	10%	2	33%	33%	0
D + m3-2	9%	9%	9	1%	4%	3	7%	21%	6	18%	23%	9
D + m3-3	16%	**66%**	9	15%	**16%**	3	46%	47%	2	19%	22%	7

Table 1. Percentage of solved problems of test sets in the Logistics, Miconic, Zenotravel and Blocksworld domains.

4 Conclusions

In this paper, we have shown that the combination of macro-operators and control rules obtained from a relational learning technique can improve the results of the IPSS planner alone. We have also shown that the different learned macro-operators do not always outperform the results of the base planner. However, in most cases, when learned control rules are applied over a good macro-operator, the results improve over using the macro-operator alone.

References

1. Rodríguez-Moreno, M.D., Oddi, A., Borrajo, D., Cesta, A.: A hybrid approach for integrating planning and scheduling: IPSS. IEEE Transactions on Knowledge and Data Engineering (In Press)
2. Minton, S.: Learning Effective Search Control Knowledge: An Explanation-Based Approach. Kluwer Academic Publishers, Boston, MA (1988)
3. Borrajo, D., Veloso, M.: Lazy incremental learning of control knowledge for efficiently obtaining quality plans. AI Review Journal **11**(1-5) (1997) 371–405
4. McCluskey, T.: Combining weak learning heuristics in general problems solvers. In: Proceedings of the 10th International Joint Conference on Artificial Intelligence. (1987) 331–333
5. Botea, A., Mueller, M., Schaeffer, J.: Learning partial-order macros from solutions. In: Proceedings of ICAPS'05, Monterrey (USA) (2005)