Challenges of Portfolio-based Planning

Mauro Vallati and **Diane Kitchin**
School of Computing and Engineering
University of Huddersfield, UK
{m.vallati,d.kitchin}@hud.ac.uk

Abstract

In the recent years the field of automated planing has significantly advanced and several powerful domain-independent planners have been developed. However, none of these systems clearly outperforms all the others in every known benchmark domain. This observation motivated the idea of configuring and exploiting a portfolio of planners to achieve better performances than any individual planner: some recent planning systems based on this idea obtained significantly good results in experimental analysis and International Planning Competitions. Such results lead us to think that future challenges for the automated planning community will converge on designing different approaches for combining existing planning algorithms.

This paper focuses on the challenges and open issues of existing approaches and highlights the possible future evolution of these techniques. In addition the paper introduces algorithm portfolios, reviews existing techniques, and describes the decisions that have to be taken during the configuration.

Introduction

Automated planning is one of the most prominent AI challenges; it has been studied extensively for several decades and led to many real-world applications (see, e.g., (Ghallab, Nau, & Traverso 2004)). During the last decade, automated planning has achieved significant advancements. However, while several powerful domain-independent planners have been developed, none of them clearly outperforms all others in every known benchmark domain. These observations motivate the idea of configuring and exploiting a portfolio of planners to achieve better overall performance than any individual planner. Moreover, portfolio-based approaches have been successfully applied to a number of combinatorial search domains, most notably the satisfiability problem (Xu *et al.* 2008).

Very recently, a number of planners based on the portfolio approach have been developed, and achieved impressive results in the last two International Planning Competitions (IPC6-7) (Fern, Khardon, & Tadepalli 2011; Coles *et al.* 2012): they won, or got very close to winning, in every track in which they took part. These include the deterministic track, learning track and, obviously, the multicore track. Achieved results let us presume that the future of AI planning will not only be focused on developing new planning algorithms, like in the last decade, but particularly on designing promising techniques for combining and exploiting existing planning systems.

This paper focuses on challenges that current and future portfolio-based planning systems have to face, in order to stimulate the development of new high-performance planning frameworks, and improve existing systems.

In the remainder of this paper, we first provide some background and further information on algorithm portfolios, existing portfolio-based planners and the portfolio configuration process. Next, we describe in detail challenges and future evolution of existing portfolio-based planners, followed by concluding remarks.

Background

This section introduces the necessary background information.

Algorithm portfolios

The term *algorithm portfolio* was firstly introduced by (Huberman, Lukose, & Hogg 1997) to describe the strategy of running several algorithms in parallel. The idea was taken from economics, where portfolios are used to maximize a utility that has an associated risk. Several authors have since used the term for describing any strategy that combines multiple algorithms, used as black-boxes, to solve a single problem instance (see, e.g. (Gomes & Selman 2001)).

The space of algorithm portfolios includes approaches ranging from those that use all available algorithms to those that always select only a single algorithm. The advantage of using the term portfolio to refer to this broader class of algorithms is that they all work for the same reason: select several algorithms in order to obtain improved performance in the average case.

Existing portfolio-based planners

In the field of automated planning, the idea of configuring and using a portfolio of techniques has become a very interesting topic in the last few years. The first work on planner portfolios was done by Roberts and Howe (Howe *et al.* 1999; Roberts & Howe 2007); in this work they showed how to generate a domain-independent portfolio of planners and compared different strategies for its configuration. It was not a completely automatic planning framework, but it was the first study on the configuration and use of portfolios of planning techniques for: (i) maximizing solved problems or (ii) minimizing runtimes.

Inspired by Roberts and Howe's work, but with several significant differences, Gerevini and collaborators developed PbP (Gerevini, Saetti, & Vallati 2009) (and lately, an enhanced version called PbP2 (Gerevini, Saetti, & Vallati 2011)); this planner extracts additional knowledge about the given domain and automatically configures a domain-specific portfolio of planners. Both versions of PbP are able to configure two different domain-specific portfolios: one focusing on speed and one focusing on plan quality, in terms of the number of actions.

Fast Downward Stone Soup (here abbreviated FDSS) (Helmert, Röger, & Karpas 2011) is a recent approach to selecting and combining a set of forward-state planning techniques included in the well known domain-independent planner Fast Downward (Helmert 2006). Very recently, an extended version of FDSS (from now on, FDSS2) has been proposed (Seipp *et al.* 2012). In this work different techniques for configuring a portfolio of automatic-obtained configuration of Fast Downward are proposed and experimentally evaluated.

ArvandHerd (Valenzano *et al.* 2012) is a very recent pure parallel portfolio that simultaneously runs on different cores an instance of the domain-independent planner LAMA (Richter & Westphal 2010), and a set of instances of the random walk based planner Arvand (Nakhost & Müller 2009). In the multicore track of the last IPC (Coles *et al.* 2012) there were several planners based on the idea of running simultaneously different planning algorithms. For the purposes of this paper all of them have the same structure, and we selected only ArvandHerd, the winner of the track, for representing the category.

Finally, a portfolio approach (Núnez, Borrajo, & Lòpez 2012) has been used by the organizers of the IPC-7 (Coles *et al.* 2012) for evaluating the state-of-the-art of domain-independent planners. They presented a general method, based on linear programming, to define the baseline sequential portfolio for a specific set of problems against which the real performance of planners can be measured and evaluated.

Portfolio configuration

There are several decisions that have to be taken while developing a framework based on a portfolio of planning techniques. We can divide the decisions into two main sets: decisions to take *offline* and decisions to take *online*, w.r.t. the performance achieved by incorporated planners on learning problems used for the portfolio configuration. The former set concerns:

- *Scope*; the resulting configured portfolio can be domain-independent, domain-specific, and instance-specific.
- *Target*; the function that the portfolio is configured for optimizing (e.g., runtime, quality of solutions).
- *Portfolio size*; minimum and maximum number of planners that can be selected during the configuration.
- *Scheduling strategy*; the strategy that will be used for running the selected planners (e.g., pure parallel, sequential, mixed, ...).
- *Incorporated planners*; the selection of planning algorithms to consider.
- *Training instances*; the selection of instances used for evaluating the performances of incorporated planners.

The set of decisions to take online is:

- *Evaluation of the planners*; the performance metrics used for evaluating the planners on training instances.
- *Planner selection*; the techniques used for selecting the planners to include in the portfolio (e.g., number of solved problems, statistical tests, ...).
- *Allocation strategies and planner ordering*; the strategy for deciding the CPU-time allocated to selected planners and planners' execution order.

Finally, it is good practice to define the strategy for the evaluation of performances of configured portfolio on a subset of testing problems. A configured portfolio must achieve, at least, better performances than every individual incorporated planner, so it is good practice to compare against all of them for a first evaluation.

It must be noted that the decisions, from both online and offline sets, are strictly related. It is important to consider all the phases together.

Challenges

We provide a list of what we consider to be open issues or future avenues in portfolio-based approaches for automated planning. We are aware that this list is not complete, but we are highlighting the most important ones: we have not included those challenges and future avenues that are dependent on the selected ones. This is the case, for instance, for challenges related to determining the minimum and maximum portfolio size, which mainly depends on the planner selection strategy.

Target

Every portfolio must have a target function to optimize. Typically these functions are very easy and concern three different performances, usually taken individually: runtimes, quality of solution plans (in terms of number of actions or actions cost) and number of solved problems.

Most of the existing approaches are optimized for finding good quality plans (in terms of number of actions or action costs) or for maximizing the number of solved problems, and all of them exploit configured portfolios composed by several different planners. The only existing system that is able to configure a domain-specific portfolio (also considering additional domain-related knowledge) for minimizing runtime is PbP (and its latest version, PbP2). Analyzing the runtimes-configured portfolios that PbP generated for IPC6 benchmark domains (Gerevini, Saetti, & Vallati 2009; Fern, Khardon, & Tadepalli 2011), it is easy to see that usually a single planner (possibly with additional knowledge extracted from the domain in the form of macro-actions) is selected. It would be interesting, for all the automated planning community, to offer an in-depth analysis for better understanding this behaviour. Is it related to the scheduling strategy, with the other knowledge extracted from the domain or is it typical of domain-specific portfolios focusing on speed?

Learning problems

Implementing mechanisms to autonomously collect learning examples for automated planning is still an open issue. Traditionally, training problems are selected from IPCs benchmark or obtained by random generators that offer some parameters to tune the problems difficulty. The former approach is limited to the small number of already existing domains and instances; moreover many benchmark problems from earlier IPCs are trivial for most of the current state-of-the-art planning systems. On the other hand, the latter approach has two main limitations: (i) it is not trivial to guarantee problems' solvability; (ii) the generators' parameters are domain-specific and tuning them to generate good quality learning examples implies significant domain expertise.

Planners selection and ordering

A striking result in (Seipp *et al.* 2012) is that, in terms of solution quality, none of the more sophisticated strategies for configuring portfolios, performs better than the uniform portfolio (i.e., *all* the incorporated planners are selected and have the same amount of CPU time). This result supports the assumption they made that most planners either solve a problem fast or not at all. Additionally, their work indicates that portfolio performance can be improved much more by diversifying the set of incorporated planners than by adjusting selected planners' runtimes and ordering. This result is very strong; it seems that the configuration of a portfolio for maximizing plan quality is completely wasteful. We are not totally convinced by this result, and we would like to see a comparison of strategies for configuring portfolios on very hard and challenging benchmark instances.

On the other hand the selection of planners and their ordering is critical for optimizing runtimes, except in the cases where a single planner is selected.

Predictive model

The only existing work in automated planning that builds and exploits predictive models for configuring a portfolio of planning systems is (Roberts & Howe 2007). In this work Roberts and Howe showed that predictive models can achieve significantly good results. Despite this work, all the recent approaches have abandoned this way and even so obtained significant results. This seems to be counterintuitive with the portfolio approaches applied to different fields of Artificial Intelligence (see, e.g. (Xu *et al.* 2008)) where predictive models are extensively and efficiently used.

Automated framework

Most of the existing systems do not have a completely automated configuration process. It would be useful, for a better understanding of portfolios and planners performances, to have a framework that is able to automatically generate several different classes of portfolios and to compare all of them through different techniques. Such a framework will provide an easy tool for studying the performances of portfolios and to evaluate the impact of new ideas in configuration steps. Moreover, that framework would also suggest a potential method for testing new planners, based on measuring the performance improvements obtained in several different portfolios by adding them as incorporated planners.

In different fields of Artificial Intelligence some tools already exist like the one we just outlined. A full working example, that the automated planning community should regard, is the HAL system (Nell *et al.* 2011). It has been designed for supporting the empirical analysis and design tasks encountered during the development, evaluation and application of high-performance algorithms.

Regarding AI planning, a well known existing tool is it-SIMPLE (Vaquero *et al.* 2012). This Knowledge Engineering environment supports the design of planning applications; in particular it focuses on the initial phases of the design cycle, for facilitating the transition of requirements to formal specifications. The resulting domains can be evaluated by comparing the performances achieved on them by different existing planners. We believe that improving the experimental evaluation by including portfolio techniques would result in a complete and helpful tool for designing and evaluating applications, and also for obtaining the best selection of planners that works well on them.

Share information

Existing portfolio approaches use planning systems as black-boxes. Selected planners do not share any kind of information, knowledge or evaluations about the search space of the current problem. In order to push forward the performances of a portfolio of planning systems, we believe that they should share information, communicate and cooperate for reaching the goal (e.g. by exploring different areas of the search space or by trying to satisfy independent goals). A possible way for sharing information between selected planning algorithms is by exploiting techniques from multiagents systems.

Evaluation

After the configuration of the portfolio of planning techniques, it is important to find a method for evaluating its performances. For evaluating the configured portfolio, there are two main questions to answer: (i) given the selected structure of the portfolio, did we correctly configure it? and, (ii) is the selected portfolio structure suitable for our target and scope? For finding an answer to the former question, the best strategy is to compare the configured portfolio with an oracle: a portfolio with same structure but configured exactly on the testing problems.[1] For the latter question, there is still no answer. This problem has not been considered yet. Ideally, it would be enough to compare against differently structured portfolios, but it should be noted that selecting the most appropriate portfolio structure for this comparison

[1]This is exactly the strategy adopted by Núñez et al. (Núnez, Borrajo, & Lòpez 2012) for generating a baseline of the performance, to compare with other planners.

is -at least- as difficult as selecting the preferred portfolio configuration.

Conclusions

The existing automated planning technology offers a large, growing set of powerful techniques and efficient domain-independent planners, but none of them outperforms all the others in every known planning domain. From a practical perspective, it is then useful to consider a portfolio-based approach to planning involving several techniques and planners: recently, several different high-performance portfolio-based planners have been developed.

In this paper we briefly introduced the idea of algorithm portfolio, existing planning systems based on this approach, and the steps required for defining the structure of a portfolio-based planner. Then we focused on the challenges of existing techniques for configuring a portfolio of planners, in order to (i) give an overview of the state-of-the-art of portfolio-based planners and, (ii) stimulate development of new high-performance planning systems based on this approach.

This analysis is motivated by the excellent results achieved by portfolio-based planning systems in recent International Planning Competitions: they won, or got very close to, in every track in which they took part. These impressive results let us suppose that future automated planning challenges will be related to algorithms and techniques for effectively combining planners, in order to obtain results that can not be achieved by a single domain-independent planner.

Further studies are needed to analyze the highlighted open issues and to increase the performances that can be achieved by exploiting a portfolio approach in automated planning. We are confident that these techniques, only recently applied in automated planning, will lead to further significant improvements in the close future.

References

Coles, A.; Coles, A.; Olaya, A. G.; Jiménez, S.; Lòpez, C. L.; Sanner, S.; and Yoon, S. 2012. A survey of the seventh international planning competition. *AI Magazine* 33:83–88.

Fern, A.; Khardon, R.; and Tadepalli, P. 2011. The first learning track of the international planning competition. *Machine Learning* 84:81 – 107.

Gerevini, A.; Saetti, A.; and Vallati, M. 2009. An automatically configurable portfolio-based planner with macro-actions: PbP. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS-09)*, 350 – 353.

Gerevini, A.; Saetti, A.; and Vallati, M. 2011. PbP2: Automatic configuration of a portfolio-based multiplanner. In *Working notes of 21st International Conference on Automated Planning and Scheduling (ICAPS-11) 7th International Planning Competition*.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers.

Gomes, C. P., and Selman, B. 2001. Algorithm portfolios. *Artificial Intelligence* 126(1-2):43–62.

Helmert, M.; Röger, G.; and Karpas, E. 2011. Fast Downward Stone Soup: A baseline for building planner portfolios. In *Proceedings of the ICAPS-11 Workshop of AI Planning and Learning (PAL)*.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research (JAIR)* 26:191 – 246.

Howe, A.; Dahlman, E.; Hansen, C.; vonMayrhauser, A.; and Scheetz, M. 1999. Exploiting competitive planner performance. In *Proceedings of the 5th European Conference on Planning (ECP-99)*, 62–72.

Huberman, B.; Lukose, R.; and Hogg, T. 1997. An economics approach to hard computational problems. *Science* 265:51–54.

Nakhost, H., and Müller, M. 2009. Monte-carlo exploration for deterministic planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 1766–1771.

Nell, C.; Fawcett, C.; Hoos, H. H.; and Leyton-Brown, K. 2011. HAL: A framework for the automated analysis and design of high-performance algorithms. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization (LION-5)*, 600–615.

Núnez, S.; Borrajo, D.; and Lòpez, C. L. 2012. How good is the performance of the best portfolio in ipc-2011? In *Proceedings of ICAPS-12 Workshop on International Planning Competition*.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research (JAIR)* 39:127 – 177.

Roberts, M., and Howe, A. 2007. Learned models of performance for many planners. In *Proceedings of the ICAPS-07 Workshop of AI Planning and Learning (PAL)*.

Seipp, J.; Braun, M.; Garimort, J.; and Helmert, M. 2012. Learning portfolios of automatically tuned planners. In *Proceedings of the 22nd International Conference on Automated Planning & Scheduling (ICAPS-12)*.

Valenzano, R.; Nakhost, H.; Müller, M.; Schaeffer, J.; and Sturtevant, N. 2012. Arvandherd: Parallel planning with a portfolio. In *Proceedings of the 20st European Conference on AI (ECAI-12)*.

Vaquero, T.; Tonaco, R.; Costa, G.; Tonidandel, F.; Silva, J. R.; and Beck, J. C. 2012. itSIMPLE4.0: Enhancing the modeling experience of planning problems. In *Proceedings of the 22nd International Conference on Automated Planning & Scheduling (ICAPS-12) – System Demonstration*, 11–14.

Xu, L.; Hutter, F.; Hoos, H.; and Leyton-Brown, K. 2008. SATzilla: Portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research (JAIR)* 32:565–606.