# IBACOP and IBACOP2 Planner

**Isabel Cenamor** and **Tomás de la Rosa** and **Fernando Fernández**
Departamento de Informática, Universidad Carlos III de Madrid
Avda. de la Universidad, 30. Leganés (Madrid). Spain
icenamor@inf.uc3m.es, trosa@inf.uc3m.es, ffernand@inf.uc3m.es

### Abstract

This manuscript describes several planning portfolios that use the same base planners. Our Instance Based Configured Portfolios follow two different strategies. IBaCoP is configured a priori following a Pareto efficiency approach to select a sub-set of planners (baseline strategy), which receive the same execution time for all planning problems. On the contrary, IBaCoP2 decides for each problem the sub-set of planners to use. Such decisions are based on predictive models learnt also with training instances gathered from previous executions of the base planners. Both portfolios compete in the sequential satisficing, agile and multi-core tracks.

## Introduction

In the state of the art, there are several portfolios that define different ways to combine simple base planners. All of them are motivated by the general idea that none of existing planners dominates all other in all cases. The most popular strategy is the static, where the portfolio components and the time for each planner is defined "a priori" and maintained for all domains and problems. Fast Downward Stone Soup (FDSS) (Helmert 2006) is an example of this type of portfolio. It has various configuration based on previous planning results. It obtained good results in the last IPC (*International Planning Competition*).

The family of PbP portfolios, PbP and PbP2 (Gerevini, Saetti, and Vallati 2009; 2011), generates domain-specific multi-planners from a set of domain-independent planning techniques. It generates macro-actions, optimizes planner parameters and selects specific planners for each domain. Therefore, it generates a different configuration for each domain. This family of portfolios won both learning tracks that were held in past IPCs.

All our portfolios use as base planners the ones competing in the last IPC in the sequential satisficing track (27 planners), plus LPG-td. The different configurations are obtained applying two different strategies. One is a static configuration that obtains a sub-group of planners (IBaCoP), and the other is a dynamic portfolio configured through CRISP-Data Mining methodology (Chapman et al. 2000; Han, Kamber, and Pei 2006) (IBaCoP2).

IBaCoP is the result of applying the Pareto efficiency technique (Censor 1977) to select a sub-set of planners from the planners in sequential satisficing track plus LPG-td. We select the planners that dominates all others in at least one domain (from a set of training domains), taking into account quality and time. We assign the same running time for all selected planners

IBaCoP2 is a portfolio auto configurable with a classification model. This portfolio is an evolution of IBaCoP, since it takes as base planners, the planners selected by IBaCoP. However, IBaCoP2 performs a second planner selection using predictive models. These models are the result of learning processes and predict the behaviour of the planners in future problems, i.e. whether they will be able to solve the problems or not. The planners with higher confidence are selected and ordered following such confidence. Then, running time is divided uniformly among them.

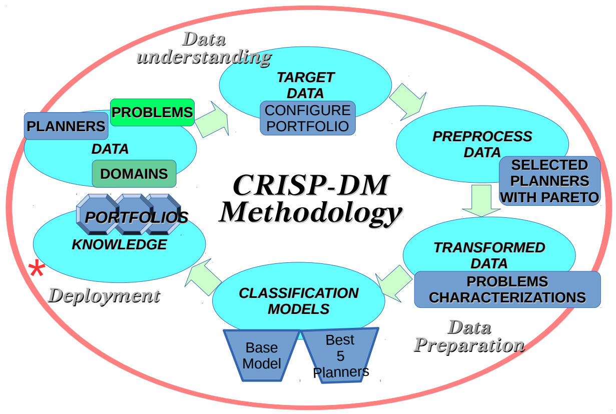
The remainder of the paper is organized as follows. In the next section we present the general ideas of the portfolios, with their components, the training data, and how we finally created the portfolios. We finish with the specific information of the planners in the different tracks.

## General System Description

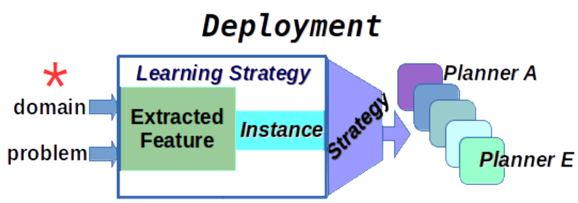
In this section, we explain the general process to configure IBaCoP planners. This system is based on a CRISP-Data Mining and the general idea is depicted in the figure 1.

The first phase in the methodology is to understand the aims of the data mining process: to extract knowledge from past planning executions to create portfolio configurations. Another important consideration is the available inputs of the process. We have two different inputs: the planners, that were extracted from the last International Planning Competition (IPC) plus LPG-td; and the domains and the problems from different competitions and different tracks. The output of the process is the configuration of the portfolio, which comprises the way to combine the initial planner components and the assigned time per planner.

The next step is to select and preprocess the data. The data is obtained from several sources as explained later, but basically, it can be considered as information about the execution of each planner for each problem of every domain. However, only execution data from the selected planners by the Pareto efficiency technique are used in following steps. In addition, data from problems that were not solved by any planner was eliminated.




(a) General Process




(b) Deployment

Figure 1: General CRISP-DM Process in IBaCoP system

The following step is problem characterization. We have created some features to better differentiate the planning executions. In this phase, we also choose the output attribute in the learning process, which is whether the planner found a solution for the problem in a 1800 seconds.

To continue with the process, we select and apply a variety of modelling techniques to find the model with higher accuracy. In this part of the process, it is necessary the division of the data into training and test sets, which allow us to estimate the future performance of the models. The output models should be evaluated in the context of the business objectives established in the first phase, i.e. planning capability of the developed portfolio.

The last phase, the deployment, is the part of the process that verifies previously held hypotheses through the knowledge discovered in the earlier phases of the CRISP-DM process. Particularly, this deployment appears in Figure 1(b), where the final system gets a new problem and domain, calculate the features, queries to the strategy, and returns the planners with their runtime.

## Data understanding

The first step in the DM process is to know the final objective. In our case, it is the configuration of portfolios through the methodology. In the following step, preprocess data, we analyze all the possibilities for the input of the system (domains, problems and planners) and decide the best possible selection. In the case of the planners, we started with all planners from the sequential satisficing track in the last IPC plus LPG-td. Nevertheless, there are some planners that obtained similar results, and that do not contribute diversity to the portfolio. The chosen planners are selected by using the Pareto efficiency (Censor 1977) technique between the quality of the best solution found and the time (in seconds) of the first solution to be found by the planner. Next, it gives to each planner a score that equals the number of tuples it Pareto-dominates for the same task.

**Selected Planners** The Pareto efficiency, performed with the results of IPC 2011, outputs 11 planners plus LPG-td planner:

- ARVAND (Nakhost, Valenzano, and Xie 2011)
- FD-AUTOTUNE 1 & 2 (Fawcett et al. 2011)
- FD STONE SOUP (FDSS) 1 & 2 (Helmert et al. 2011)
- LAMA 2008 & 2011 (Richter, Westphal, and Helmert 2011)
- PROBE (Lipovetzky and Geffner 2011)
- MADAGASCAR (Rintanen 2011)
- RANDWARD (Olsen and Bryce 2011)
- YAHSP2-MT (Vidal 2011)
- LPG-TD (Gerevini et al. 2004)

**Selected Domains** The next step is to define the set of problems and domains used to learn the models. We need a wide group of problems and domains to generalize properly. We have included the planning problems available from past IPCs, discarding the first four competitions given that problems are too easy for the state-of-the-art planners.

- IPC5: openstack, pathways, rover, storage, tpp and trucks.
- IPC6: cybersec, elevators, openstack, parcprinter, pegsol, pipesworld, scanalyzer, sokoban, transport and woodworking.
- IPC7: barman, elevators, floortile, nomystery, visitall, tidybot, openstacks, parcprinter, parking, pegsol, sokoban, scanalyzer, transport and woodworking
- Learning 2008: gold-miner, matching-bw, n-puzzle, parking and sokoban.
- Learning 2011: barman, blockworld, depots, gripper, parking, rover satellite, spanner and tpp.

We consider all the successful problems, and we did not take into account repeated domains or repeated problems. We do not know which domains would be used in the future, so we need to consider a wide and significant number of instances for the learning process. Finally, we obtained 1070 different problems to create the learning models.

## Data Preparation

The next step is the characterization of the problem (transformed data in DM process). For this task we consider some features in the planning task previously used (Roberts and Howe 2009) and include others for a better particularization of the problems complexity (Cenamor, de la Rosa, and Fernández 2012). These features have shown good accuracy for configuring portfolios (Cenamor, de la Rosa, and Fernández 2013). In addition, we create some new features to improve the characterization of the initial state of the problem.

Some basic features are directly extracted from the PDDL files. A group of elaborated features are generated from the problem translation to the $\text{SAS}^+$ formalism (Backstrom and Nebel 1995) and its induced graphs, i.e., the causal graph and the domains transition graphs. These features describe number of edges, weights, variables of the graphs. Besides, we include statistical information of the graphs, such as the sum, maximum and standard deviation of the edges and weights. We also consider other information that appears in the translation and preprocess of Fast Downward (Helmert 2006) (FD) system.

As new features we include the most representative heuristic functions computed for the initial state with unit cost, the ratio $h_{\text{FF}}/h_{max}$ and a set of features to characterize the *fact balance* of the relaxed plan ($RP$). We define the fact balance for fact $p$, as the number of times $p$ appears as an add effect of an action belonging to $RP$, minus the number of times $p$ is a delete effect of an action in $RP$, considering original actions where deletes are not ignore. The intuition behind fact balances is that high positive values would characterize easier (relaxed) problems for a given domain, since achieved facts need to be deleted many times. Given that the number of relevant facts of a planning task is variable, we compute statistics (i.e., min, max, average and variance) for the fact balance of the relevant facts. Additionally, we compute statistics only considering facts that are goals, following the same procedure.

The time to extract features is negligible given that features wrt. graphs imply basic arithmetic computations and heuristic functions are only called once for the initial state. To finalize the data preparation, we perform a feature selection process where we get the same performance with a subgroup of all features (35 features). Such features are:

- From the previous work (Cenamor, de la Rosa, and Fernández 2012), we include the number of objects, the number of goals, the number of variables in the causal graph (CG), the ratio between the high level variable and all variables in the CG, the standard deviation of the number of input edges in the CG, the average of the number of output edges in the CG, the maximum and the average weight of the output edges in the CG, the standard deviation of the number of output edges in high level variables in the CG, the maximum weight of input edges in high level variables in the CG, the number of variables in the domain transition graph (DTG), the number of edges in the same graph and the maximum weight of input edges in the DTG.
- As new features from previous work we include: the number of types of objects, the number of functions, the number of auxiliary atoms in the translate process between PDDL to $\text{SAS}^+$, the number of implied effect removed, the number of translator facts and the number of the mutex group in the translator process. In addition, the feature selection includes the number of relevant facts, the number of actions, the ratio $h_{\text{FF}}/h_{max}$, the fact balance (average and variance), the goal balance (minimum, average and variance). As well as the following heuristics: $h_{add}$, $h_{max}$ (Bonet and Geffner 2001), Context enhanced additive (Helmert and Geffner 2008), $h_{\text{FF}}$ (Hoffmann and Nebel 2011), Goal count (i.e., the number of unsatisfied goals), Landmark count (Richter, Helmert, and Westphal 2008) and Landmark cut (Helmert and Domshlak 2009) .

### Modelling the Data

One of the most important steps in this system is learning classification models to predict whether a planner will find a solution for a problem. We trained with 25 classification algorithms (for different model types: trees, rules, support vector machines and instance based learning) using WEKA (Witten and Frank 2005). WEKA is a data mining toolkit that provides a standard format for running machine learning algorithms. We selected the model with best accuracy ($99.83\%$ in training phase): Random Forest (Breiman 2001). This model is a combination of tree predictors such that each tree depends on the values of a random vector and with the same distribution.

In addition, we include two strategies to compare the performance of the system. The first one selects planners only with the Pareto efficiency, and the other uses in addition the classification model.

## Deployment

In this section, we explain the different configurations of the system in the different tracks (sequential satisficing, sequential agile and sequential multi-core). The summary is reported in Table 1.

### Sequential Satisficing Planner

The IBaCoP planner uses the 12 planners described in subsection Selected Planners, which are selected by the Pareto efficiency analysis. The execution order of the planners is arbitrary, since time is divided uniformly among all them (150 seconds per planner).

The IBaCoP2 planner uses the learned model described in section Modelling the Data . It selects the 5 planners with the highest confidence of solving the problem. The execution order of the planners is based on their confidence. The running time is assigned uniformly to each planner (360 seconds).

### Sequential Agile Planner

As in the Sequential Satisficing track, the IBaCoP planner uses the 12 planners described in subsection Selected Planners. However, we assigned as running time the average time to find the first solution in 300 seconds. The execution order for the planners is given by this average from less time to greater values. Even though all planners are included, in practice, only a few of them will have the chance to run, until consuming the time bound of 300 seconds.

The IBaCoP2 planner uses the learned model to select 5 planners; the order of the planners is decided from the confidence and the time for each planner is the same for the five planners.

### Sequential Multi-core Planner

For Multi-core track, the planners are the same as for the sequential satisficing track, but taking into account that we have more time ($1800\ seconds \times 4\ cores$). Memory is divided equally among all the planners running in the different cores.

| | Planner | seq-sat | seq-agl | seq-mco |
|---|---|---|---|---|
| 1 | yahsp2-mt | 150 | 5 | 600 |
| 2 | randward | 150 | 50 | 600 |
| 3 | arvand | 150 | 55 | 600 |
| 4 | fd-autotune-1 | 150 | 50 | 600 |
| 5 | lama-2008 | 150 | 45 | 600 |
| 6 | probe | 150 | – | 600 |
| 7 | madagascar | 150 | 45 | 600 |
| 8 | lpg-td | 150 | 50 | 600 |
| 9 | fdss-1 | 150 | – | 600 |
| 10 | lama-2011 | 150 | – | 600 |
| 11 | fd-autotune-2 | 150 | – | 600 |
| 12 | fdss-2 | 150 | – | 600 |

Table 1: Time for each planner execution in sequential tracks

## Acknowledgements

We generated sequential portfolios of existing planners to be submitted to the International Planning Competition 2014. Thus, we would like to acknowledge and thank the authors of the individual planners for their contribution and hard work.

This work has been partially supported by the Spanish project TSI-090302-2011-6 and TIN2012-38079-C03-02.

## References

Backstrom, C., and Nebel, B. 1995. Complexity results for SAS+ planning. *Computational Intelligence* 11:625–655.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1):5–33.

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

Cenamor, I.; de la Rosa, T.; and Fernández, F. 2012. Mining ipc-2011 results. In *Proceedings of the Third Workshop on the International Planning Competition*.

Cenamor, I.; de la Rosa, T.; and Fernández, F. 2013. Learning predictive models to configure planning portfolios. In *Proceedings of the Workshop on the Planning and Learning*.

Censor, Y. 1977. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization* 4(1):41–59.

Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; and Wirth, R. 2000. Crisp-dm 1.0 step-by-step data mining guide. *CRISPWP-0800*.

Fawcett, C.; Helmert, M.; Hoos, H.; Karpas, E.; Röger, G.; and Seipp, J. 2011. Fd-autotune: Automated configuration of fast downward. *The 2011 International Planning Competition* 31–37.

Gerevini, A.; Saetti, A.; Serina, I.; and Toninelli, P. 2004. Lpg-td: a fully automated planner for pddl2. 2 domains. In *In Proc. of the 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04) International Planning Competition abstracts*. Citeseer.

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)*.

Gerevini, A.; Saetti, A.; and Vallati, M. 2011. Pbp2: Automatic configuration of a portfolio-based multi-planner. *The 2011 International Planning Competition*.

Han, J.; Kamber, M.; and Pei, J. 2006. *Data mining: concepts and techniques*. Morgan kaufmann.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *ICAPS*.

Helmert, M., and Geffner, H. 2008. Unifying the causal graph and additive heuristics. In *ICAPS*, 140–147.

Helmert, M.; Röger, G.; Seipp, J.; Karpas, E.; Hoffmann, J.; Keyder, E.; Nissim, R.; Richter, S.; and Westphal, M. 2011. Fast downward stone soup. *The 2011 International Planning Competition* 38.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26(1):191–246.

Hoffmann, J., and Nebel, B. 2011. The ff planning system: Fast plan generation through heuristic search. *arXiv preprint arXiv:1106.0675*.

Lipovetzky, N., and Geffner, H. 2011. Searching with probes: The classical planner probe. *The 2011 International Planning Competition* 30(29):71.

Nakhost, H.; Valenzano, R.; and Xie, F. 2011. Arvand: the art of random walks. *The 2011 International Planning Competition* 15.

Olsen, A., and Bryce, D. 2011. Randward and lamar: Randomizing the ff heuristic. *The 2011 International Planning Competition* 55.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In *AAAI*, volume 8, 975–982.

Richter, S.; Westphal, M.; and Helmert, M. 2011. Lama 2008 and 2011. *The 2011 International Planning Competition* 50.

Rintanen, J. 2011. Madagascar: Efficient planning with sat. *The 2011 International Planning Competition* 61.

Roberts, M., and Howe, A. 2009. Learning from planner performance. *Artificial Intelligence* 173(5):536–561.

Vidal, V. 2011. Yahsp2: Keep it simple, stupid. *The 2011 International Planning Competition* 83–90.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.