MIPLAN

Sergio Núñez and Daniel Borrajo and Carlos Linares López
Departamento de Informática, Universidad Carlos III de Madrid
Avda. de la Universidad, 30. 28911 Leganes (Madrid). Spain
sergio.nunez@uc3m.es, dborrajo@ia.uc3m.es, clinares@inf.uc3m.es

Abstract
We describe MIPLAN, the planning portfolio system submitted to the learning track of the International Planning Competition 2014. MIPLAN uses Mixed-Integer Programming techniques to automatically configure a sequential portfolio of domain-independent planners for each input domain, so that it derives the best achievable performance for the given training data set. MIPLAN competes in the quality subtrack of the learning track.

Introduction
MIPLAN is a participant planning system in the learning track of the International Planning Competition 2014. It is able to automatically generate a portfolio configuration of domain-independent planners for a specific input domain (learning phase) and runs a specific sequential portfolio for each input instance (execution phase).

Learning Phase
The learning phase focuses on generating a portfolio configuration for a given input domain (and a set of training instances of that domain). The resulting portfolio is a linear combination of candidate planners defined as a sorted set of pairs $(p_i, t_i)$, where $p_i$ is the $i$-th planner and $t_i$ is the time allotted to its execution.

The learning process is composed of two tasks. The first one runs every candidate planner with all the training instances of the given domain with the settings of the competition. Thus, each candidate planner is executed with every planning task for 15 minutes with a memory limit of 4 GB. The second task aims to derive the best linear combination of candidate planners using the particular Mixed-Integer Programming (MIP) task described in (Núñez, Borrajo, and Linares López 2014; 2012). This MIP task computes the portfolio with the best achievable performance with respect to the selection of training planning tasks and an objective function, which consists of a weighted combination of quality and runtime to assess the performance of planners.

Execution Phase
MIPLAN can be executed with the particular knowledge generated in the learning phase (i.e., the portfolio configuration computed for the input domain) and as a domain-independent portfolio. If it is executed with knowledge, MIPLAN will run the linear combination of planners defined in the learning phase. Otherwise, it will run the default portfolio configuration shown in Table 1. This configuration was computed using the same MIP task. However, we considered a set of training problems composed of planning tasks from several domains. Specifically, we selected the list of domains initially suggested in the quality track: BAR-MA*N (2011), DEPOTS (2011), GOLD-MINER (2008), PARKING (2011), ROVERS (2011), SOKOBAN (2008), SPANNER (2011) and THOUGHTFUL (2008).

<table>
<thead>
<tr>
<th>Planner</th>
<th>Allotted time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMAR</td>
<td>8</td>
</tr>
<tr>
<td>YAHSP-2-MT</td>
<td>16</td>
</tr>
<tr>
<td>LAMA-2008</td>
<td>16</td>
</tr>
<tr>
<td>LAMA-2011</td>
<td>16</td>
</tr>
<tr>
<td>ARVAND</td>
<td>58</td>
</tr>
<tr>
<td>SGPLAN</td>
<td>83</td>
</tr>
<tr>
<td>PROBE</td>
<td>105</td>
</tr>
<tr>
<td>LPG</td>
<td>273</td>
</tr>
<tr>
<td>WA* - LAZY - CG</td>
<td>325</td>
</tr>
<tr>
<td>Total Time</td>
<td>900</td>
</tr>
</tbody>
</table>

Table 1: Default portfolio configuration.

The MIP task does not specify the execution sequence of the generated portfolios. However, we have sorted the execution sequence of the default portfolio configuration in increasing order of the allotted time.

Candidate Planners
MIPLAN considers a set of candidate planners to configure every sequential portfolio for a given set of training instances. Most of these candidate planners are participant planners from previous competitions. In particular, some planners are based on the Fast-Downward planning system (Helmert 2006). These planners are defined by a search algorithm, an evaluation method and a set of heuristics. Specifically, we considered weighted-A* (WA*) with $w = 3$ and greedy best-first search (GBFS), with EAGER (standard) and LAZY (deferred evaluation) variants of both search algorithms. Also, we considered three heuristics: additive heuristic ADD (Bonet and Geffner 2001), FF/additive
heuristic FF (Hoffmann and Nebel 2001; Keyder and Geffner 2008) and causal graph heuristic CG (Helmert 2004). The list of candidate planners considered is shown below:

- ARVAND (Nakhost, Valenzano, and Xie 2011)
- FD-AUTOTUNE 1 & 2 (Fawcett et al. 2011)
- FD STONE SOUP 1 & 2 (Helmert, Röger, and Karpas 2011)
- LAMA 2008 & 2011 (Richter, Westphal, and Helmer 2011)
- PROBE (Lipovetzky and Geffner 2011)
- MADAGASCAR (Rintanen 2011)
- RANDWARD (Olsen and Bryce 2011)
- YAHSP2-MT (Vidal 2011)
- LPG-TN (Gerevini et al. 2004)
- LAMAR (Olsen and Bryce 2011)
- DAE-YAHSP (Dréö et al. 2011)
- SGPLAN (Hsu and Wah 2008)
- Greedy best-first, with Eager evaluation and FF
- Greedy best-first, with Eager evaluation and FF, CG
- Weighted-A$^*$ w=3, with Eager evaluation and ADD
- Greedy best-first, with Eager evaluation and CG
- Weighted-A$^*$ w=3, with Lazy evaluation and CG
- Greedy best-first, with Lazy evaluation and CG

We have used the optimization flags, command-line options and other parameters provided by the authors of planners shown in Table 1. To compile and run MIPLAN just run build.sh and solve <domain file> <problem file> <plan file> [<dck folder>] . To run the learning phase just run learner -o <domain file> -t <training folder> -k <dck folder>.

Acknowledgments
We automatically generated sequential portfolios of existing planners by means of a MIP task 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 PlanInter action project TIN2011-27652-C03-02.

References