Does Representation Matter in the Planning Competition?

Patricia J. Riddle
Department of Computer Science
University of Auckland
Auckland, New Zealand
(pat@cs.auckland.ac.nz)

Robert C. Holte
Computing Science Department
University of Alberta
Edmonton, AB, Canada T6G 2E8
(holte@cs.ualberta.ca)

Michael W. Barley
Department of Computer Science
University of Auckland
Auckland, New Zealand
(barley@cs.auckland.ac.nz)

Abstract

This paper explores six different representations of the BlocksWorld Domain. It compares the results of seven planners run on these representations. It shows that the rankings for the International Planning Competition, using the non-satisficing scoring function, would change for every representation.

Introduction

This paper explores whether different representations of a problem could affect the outcome of the International Planning Competition (IPC).[1] Specifically we ask whether the order of planners, as determined using the scoring formula from the IPC, changes when different representations are used for BlocksWorld problems. Of particular interest is whether the winner of the competition is affected by the choice of problem representation.

This paper presents results for six different representations of the BlocksWorld Domain.[2] **Orig** is the standard representation that comes with the Fast Downward planners as a benchmark dataset. **InfT** (for "infinite table") is the same standard representation, with a very slight change to make it as similar as possible to **LarT**. **LarT** (for "large table") introduces names for the locations on the table, but has enough locations that it is always possible to place a block on the table at any time. In addition to having table locations, the three "level" representations (**L1**, **L2**, **L3**) explicitly represent the position of a block within a stack—its height (or level) above the table top. An example of this representation is given in the "Level Representations" section.

On each of these representations we ran seven modern planners. Downward-classic, LAMA, Auto1 and Auto2, were downloaded on February 26, 2011 from http://hg.fast-downward.org. Downward-classic (Helmert 2006) is an updated implementation of the classic Fast Downward planner. LAMA (Richter and Westphal 2010) is the LAMA 2011 version as used in the satisficing track of the IPC2011. Auto1 is Fast Downward Autotune Satisficing (variant 1) based on the automated parameter tuning work of Hutter *et al.* (2006). Auto2 is Fast Downward Autotune Satisficing (variant 2), same as above but with different settings. Auto1 and Auto2 are the versions used in the satisficing track of IPC2011.

Metric-FF, MIPS and LPG-quality were all downloaded from the Strathclyde University Planning Group's planner suite. Metric-FF (Hoffmann 2003) is a domain independent planning system developed by Joerg Hoffmann, which is based on a fast-forward type planner. MIPS (Edelkamp 2003) is Edelkamp's model checking integrated planning system. LPG-quality (Gerevini, Saetti, and Serina 2003) is a planner based on local search and planning graphs. It is an incremental anytime planner producing a sequence of plans trying to improve the quality of the previous one.

The key finding of our study is that different rankings for the planners were found for every representation tried.

Experimental Setup

The experiments are all run on the BlocksWorld Domain. The PDDL (Ghallab et al. 1998) definition and problem set that comes with the Fast Downward planners was used as the starting point. It contains 35 separate problems ranging from problems with 4 to 17 blocks.

Because not all the planners are deterministic, each planner was run on each problem 100 times. The planning competition formula for non-optimal planners was used; it is optimal plan length divided by actual plan length. Note that the planning competition formula only uses solution path length. Time is only used in the sense that the planner only has 30 minutes to return the shortest solution path it finds.

Some of the planners are frequently nondeterministic, like the anytime planner LPG-quality. Downward-classic, LAMA, Auto1, and Auto2 also exhibit nondeterminism as the problems get harder. When the planners arrived at different results, an average planning competition score over the 100 runs is used. When a planner does not achieve a solution path for a run, it receives a value of 0 for the planning competition formula. Because of this last point, all the planners are treated as nondeterministic and run 100 times.

Each experiment was run with 30 minutes to finish each problem and with a virtual memory limit of 2.7GB. The experiments were run on Virtual Machines on a Xeon E7330 which were sandboxed to reserve both memory and cpu.

[1] http://ipc.icaps-conference.org/

[2] All the pddl files can be found at the location www.cs.auckland.ac.nz/∼prid013/blocks2011pddl

	Downward-classic	Metric-FF	MIPS	LPG-quality	LAMA	Auto1	Auto2
4-0	0.6 (0.0)	0.6 (0.0)	1 (0.0)	0.972 (0.124)	1 (0.0)	1 (0.0)	1 (0.0)
4-1	1 (0.0)	1 (0.0)	1 (0.0)	0.980 (0.100)	1 (0.0)	1 (0.0)	1 (0.0)
4-2	1 (0.0)	1 (0.0)	1 (0.0)	0.969 (0.125)	1 (0.0)	1 (0.0)	1 (0.0)
5-0	1 (0.0)	1 (0.0)	1 (0.0)	0.984 (0.095)	1 (0.0)	1 (0.0)	1 (0.0)
5-1	0.556 (0.0)	1 (0.0)	0.556 (0.0)	0.988 (0.069)	1 (0.0)	1 (0.0)	1 (0.0)
5-2	0.571 (0.0)	0.615 (0.0)	0.8 (0.0)	0.992 (0.057)	1 (0.0)	1 (0.0)	1 (0.0)
6-0	1 (0.0)	1 (0.0)	0.857 (0.0)	0.991 (0.067)	1 (0.0)	1 (0.0)	1 (0.0)
6-1	0.556 (0.0)	0.625 (0.0)	0.714 (0.0)	0.986 (0.069)	1 (0.0)	1 (0.0)	1 (0.0)
6-2	0.714 (0.0)	0.625 (0.0)	1 (0.0)	0.996 (0.028)	1 (0.0)	1 (0.0)	1 (0.0)
7-0	0.909 (0.0)	0.556 (0.0)	0.588 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
7-1	0.786 (0.0)	0.846 (0.0)	0.917 (0.0)	0.993 (0.037)	1 (0.0)	1 (0.0)	1 (0.0)
7-2	0.526 (0.0)	0.714 (0.0)	0.714 (0.0)	0.990 (0.062)	1 (0.0)	1 (0.0)	1 (0.0)
8-0	0.36 (0.0)	0.6 (0.0)	0.750 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
8-1	0.833 (0.0)	0.714 (0.0)	1 (0.0)	0.995 (0.031)	1 (0.0)	1 (0.0)	1 (0.0)
8-2	0.533 (0.0)	0.615 (0.0)	0.8 (0.0)	0.996 (0.047)	1 (0.0)	1 (0.0)	1 (0.0)
9-0	0.429 (0.0)	0.577 (0.0)	0.652 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
9-1	0.368 (0.0)	0.636 (0.0)	0.737 (0.0)	0.990 (0.029)	1 (0.0)	1 (0.0)	1 (0.0)
9-2	0.542 (0.0)	0.929 (0.0)	0.765 (0.0)	0.999 (0.010)	1 (0.0)	1 (0.0)	1 (0.0)
10-0	0.708 (0.0)	0.810 (0.0)	0.773 (0.0)	0.999 (0.006)	1 (0.0)	.708 (0.0)	1 (0.0)
10-1	0.5 (0.0)	0	0.762 (0.0)	0.994 (0.019)	1 (0.0)	1 (0.0)	1 (0.0)
10-2	0.386 (0.0)	0.548 (0.0)	0.654 (0.0)	0.999 (0.006)	1 (0.0)	0.708 (0.0)	1 (0.0)
11-0	0.372 (0.0)	0.727 (0.0)	0.727 (0.0)	0.984 (0.027)	1 (0.0)	0.696 (0.0)	1 (0.0)
11-1	0.306 (0.0)	0	0.556 (0.0)	0.976 (0.034)	1 (0.0)	0.417 (0.0)	0.882 (0.0)
11-2	0.447 (0.0)	0.472 (0.0)	0.773 (0.0)	0.965 (0.037)	1 (0.0)	0.472 (0.0)	1 (0.0)
12-0	0.362 (0.0)	0.708 (0.0)	0.654 (0.0)	0.908 (0.057)	1 (0.0)	0.944 (0.0)	1 (0.0)
12-1	0.486 (0.0)	0	0.68 (0.0)	0.937 (0.062)	1 (0.0)	0.515 (0.0)	0.810 (0.0)
13-0	0.447 (0.0)	0	0.7 (0.0)	0.974 (0.028)	1 (0.0)	0.831 (0.015)	0.955 (0.0)
13-1	0.611 (0.0)	0	0.667 (0.0)	0.982 (0.029)	1 (0.0)	0.611 (0.0)	0.957 (0.0)
14-0	0.475 (0.0)	0	0.679 (0.0)	0.864 (0.081)	1 (0.0)	0.475 (0.0)	1 (0.0)
14-1	0.45 (0.0)	0	0.581 (0.0)	0.918 (0.064)	1 (0.0)	0.462 (0.0)	0.818 (0.0)
15-0	0.294 (0.0)	0	0.645 (0.0)	0.900 (0.070)	0.870 (0.0)	0.625 (0.0)	0.8 (0.0)
15-1	0.361 (0.0)	0	0.591 (0.0)	0.965 (0.035)	0.867 (0.0)	0.743 (0.0)	0.929 (0.0)
16-1	0.458 (0.0)	0	0.643 (0.0)	0.952 (0.035)	0.964 (0.0)	0.529 (0.0)	0.964 (0.0)
16-2	0.325 (0.0)	0	0	0.891 (0.040)	0.867 (0.0)	0.419 (0.0)	0.963 (0.0)
17-0	0.167 (0.0)	0	0.676 (0.0)	0.873 (0.063)	0.821 (0.0)	0.595 (0.012)	0.676 (0.0)
sum	19.438 (0.0)	16.919 (0.0)	25.609 (0.0)	33.898 (0.336)	34.389 (0.0)	28.751 (0.019)	33.753 (0.0)
rank	6	7	5	2	1	4	3

Table 1: The Results for **Orig**. The total of the planners' scores is 192.757 (std = 0.337).

Each Virtual Machine has a single 2.4GHz CPU and 3GB RAM.

Orig versus InfT

The **InfT** representation differs from **Orig** in just one small detail: the table is represented explicitly instead of implicitly. In **InfT** the table has one location, P1, and *clear(P1)* is never removed. In addition *ontable(A)* is changed to *on(A,P1)* and *table(P1)*. The optimal solutions' path lengths in these two representations are identical. The PDDL representation for these two representations are as follows. To save space, only a single operator is shown. The domain representation for **Orig** is:

```
(define (domain BLOCKS)
  (:requirements :strips)
  (:predicates (on ?x ?y) (ontable ?x) (clear ?x)
             (handempty) (holding ?x))

  (:action put-down
     :parameters (?x)
     :precondition (holding ?x)
     :effect
     (and (not (holding ?x))
          (clear ?x) (handempty) (ontable ?x))))
```

The 4-0 problem representation for **Orig** is:

```
(define (problem BLOCKS-4-0)
(:domain BLOCKS)
(:objects D B A C )
(:INIT (CLEAR C) (CLEAR A) (CLEAR B) (CLEAR D) (ONTABLE C)
     (ONTABLE A) (ONTABLE B) (ONTABLE D) (HANDEMPTY))
(:goal (AND (ON D C) (ON C B) (ON B A))))
```

The domain representation for **InfT** is:

```
(define (domain patblock2)
  (:requirements :strips)
  (:predicates (clear ?pos) (table ?place2) (handempty)
             (on ?block ?place) (holding ?block))

   (:action putdown2
  :parameters (?block ?place1)
  :precondition (and (holding ?block) (clear ?place1)
              (table ?place1))
  :effect (and (handempty) (on ?block ?place1)
             (not (holding ?block)) (clear ?block))))
```

The 4-0 problem representation for **InfT** is:

	Downward-classic	Metric-FF	MIPS	LPG-quality	LAMA	Auto1	Auto2
4-0	0.6 (0.0)	0.6 (0.0)	1 (0.0)	0.948 (0.142)	1 (0.0)	1 (0.0)	1 (0.0)
4-1	1 (0.0)	1 (0.0)	1 (0.0)	0.973 (0.108)	1 (0.0)	1 (0.0)	1 (0.0)
4-2	1 (0.0)	1 (0.0)	1 (0.0)	0.988 (0.084)	1 (0.0)	1 (0.0)	1 (0.0)
5-0	1 (0.0)	1 (0.0)	1 (0.0)	0.971 (0.111)	1 (0.0)	1 (0.0)	1 (0.0)
5-1	1 (0.0)	0.714 (0.0)	0.714 (0.0)	0.987 (0.070)	1 (0.0)	1 (0.0)	1 (0.0)
5-2	1 (0.0)	0.8 (0.0)	0.615 (0.0)	0.975 (0.120)	1 (0.0)	1 (0.0)	1 (0.0)
6-0	0.316 (0.0)	0.6 (0.0)	0.857 (0.0)	0.978 (0.096)	1 (0.0)	1 (0.0)	1 (0.0)
6-1	0.714 (0.0)	0.714 (0.0)	0.714 (0.0)	0.968 (0.113)	1 (0.0)	1 (0.0)	1 (0.0)
6-2	0.714 (0.0)	0.769 (0.0)	1 (0.0)	0.988 (0.059)	1 (0.0)	1 (0.0)	1 (0.0)
7-0	0.909 (0.0)	0.909 (0.0)	0.667 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
7-1	0.611 (0.0)	0.786 (0.0)	0.786 (0.0)	0.993 (0.035)	1 (0.0)	1 (0.0)	1 (0.0)
7-2	0.455 (0.0)	0.714 (0.0)	0.714 (0.0)	0.998 (0.023)	1 (0.0)	1 (0.0)	1 (0.0)
8-0	0.360 (0.0)	0.529 (0.0)	0.750 (0.0)	0.999 (0.010)	1 (0.0)	1 (0.0)	1 (0.0)
8-1	0.714 (0.0)	0.667 (0.0)	0.833 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
8-2	0.471 (0.0)	0.615 (0.0)	0.8 (0.0)	0.997 (0.027)	1 (0.0)	1 (0.0)	1 (0.0)
9-0	0.556 (0.0)	0.556 (0.0)	0.652 (0.0)	0.999 (0.006)	1 (0.0)	1 (0.0)	1 (0.0)
9-1	0.467 (0.0)	0.737 (0.0)	0.583 (0.0)	0.987 (0.032)	1 (0.0)	1 (0.0)	1 (0.0)
9-2	0.591 (0.0)	0.619 (0.0)	0.591 (0.0)	0.999 (0.010)	1 (0.0)	0.591 (0.0)	1 (0.0)
10-0	0.607 (0.0)	0.654 (0.0)	0.773 (0.0)	1 (0.0)	1 (0.0)	0.708 (0.0)	1 (0.0)
10-1	0.533 (0.0)	0 (0.0)	0.762 (0.0)	0.994 (0.018)	1 (0.0)	1 (0.0)	1 (0.0)
10-2	0.607 (0.0)	0.773 (0.0)	0.68 (0.0)	0.999 (0.006)	1 (0.0)	0.607 (0.0)	1 (0.0)
11-0	0.593 (0.0)	0.727 (0.0)	0.696 (0.0)	0.984 (0.032)	1 (0.0)	0.8 (0.0)	1 (0.0)
11-1	0.375 (0.0)	0.5 (0.0)	0.625 (0.0)	0.975 (0.036)	1 (0.0)	0.375 (0.0)	1 (0.0)
11-2	0.472 (0.0)	0.4595 (0.0)	0.739 (0.0)	0.966 (0.038)	1 (0.0)	1 (0.0)	1 (0.0)
12-0	0.362 (0.0)	0 (0.0)	0.708 (0.0)	0.911 (0.060)	1 (0.0)	0.944 (0.0)	1 (0.0)
12-1	0.515 (0.0)	0.630 (0.0)	0.548 (0.0)	0.921 (0.063)	1 (0.0)	0.773 (0.041)	1 (0.0)
13-0	0.467 (0.0)	0.618 (0.0)	0.75 (0.0)	0.972 (0.033)	0.875 (0.0)	0.583 (0.0)	0.913 (0.0)
13-1	0.379 (0.0)	0.564 (0.0)	0.647 (0.0)	0.977 (0.030)	0.88 (0.0)	0.611 (0.0)	0.957 (0.0)
14-0	0.594 (0.0)	0.528 (0.0)	0.731 (0.0)	0.873 (0.081)	1 (0.0)	0.594 (0.0)	1 (0.0)
14-1	0.4 (0.0)	0.529 (0.0)	0.621 (0.0)	0.908 (0.066)	1 (0.0)	0.514 (0.0)	1 (0.0)
15-0	0.213 (0.0)	0 (0.0)	0.645 (0.0)	0.879 (0.065)	0.870 (0.0)	0.625 (0.0)	0.909 (0.0)
15-1	0.413 (0.0)	0 (0.0)	0.619 (0.0)	0.946 (0.038)	0.813 (0.0)	0.565 (0.0)	0.867 (0.0)
16-1	0.491 (0.0)	0.659 (0.0)	0.692 (0.0)	0.938 (0.043)	0.844 (0.0)	0.529 (0.0)	0.9 (0.0)
16-2	0.302 (0.0)	0 (0.0)	0.743 (0.0)	0.876 (0.044)	0.867 (0.0)	0.652 (0.028)	0.963 (0.0)
17-0	0.319 (0.0)	0 (0.0)	0.575 (0.0)	0.849 (0.066)	0.119 (0.132)	0.479 (0.0)	0.719 (0.0)
sum	20.119 (0.0)	19.970 (0.0)	25.832 (0.0)	33.719 (0.372)	33.266 (0.132)	28.952 (0.050)	34.227 (0.0)
rank	6	7	5	2	3	4	1

Table 2: The Results for **InfT**. The total of the planners' scores is 196.085 (std = 0.554).

```
(define (problem BLOCKS-4-0)
  (:domain patblock2)
  (:objects P1 D B A C )
  (:INIT (CLEAR C) (CLEAR A) (CLEAR B) (CLEAR D)
         (ON C P1) (ON A P1) (CLEAR P1) (ON B P1)
         (ON D P1) (HANDEMPTY) (TABLE P1))
  (:goal (AND (ON D C) (ON C B) (ON B A))))
```

The expectation going into this experiment, was that **Orig** and **InfT** should return identical results. Tables 1 and 2 contain the values for the IPC scoring formula averaged over 100 runs (the values in parentheses are the standard deviations of the scores). The "sum" row in each table shows the total for each column, *i.e.,* the total score over all the problems by a specific planner. The caption gives the total of the scores for all the planners, the sum of the "sum" row, and its standard deviation. The bottom row of each table shows the rank of each planner for that representation.

The small change between **Orig** and **InfT** does have some affect on the planners. **Orig** generates 32 operators for the Downward family of planners while **InfT** generates 40. This means that for the larger problems LAMA, Auto1, and Auto2 will become nondeterministic sooner on the **InfT** representation. While most planners perform a bit better on the **InfT** representation, LAMAand LPG-quality actually perform slightly worse on the **InfT** representation.

Looking at the results, LAMA starts to produce non-optimal solutions earlier in **InfT** although it only becomes nondeterministic on the final problem for which it performs very poorly. Auto1 also starts to produce non-optimal solutions slightly earlier on **InfT**, while Auto2 actually produces non-optimal solutions later on **InfT**. Metric-FF performs better because there are many fewer problems for which it gets a segmentation error. Note that LPG-quality was only run for 80 runs on problem 16-2 which might cause a higher standard deviation.

What is causing the differing results on these two very similar representations? Further testing reveals that with Downward-classic changing from *ontable(A)* to *table(P1) on(A,P1)* changed the results as did removing *P1* from the parameter list of the operators. Downward-classic was de-

	Downward-classic	Metric-FF	MIPS	LPG-quality	LAMA	Auto1	Auto2
4-0	0.6 (0.0)	0.6 (0.0)	1 (0.0)	0.973 (0.120)	1 (0.0)	1 (0.0)	1 (0.0)
4-1	1 (0.0)	0.833 (0.0)	1 (0.0)	0.956 (0.143)	1 (0.0)	1 (0.0)	1 (0.0)
4-2	1 (0.0)	0.75 (0.0)	1 (0.0)	0.995 (0.05)	1 (0.0)	1 (0.0)	1 (0.0)
5-0	1 (0.0)	0.75 (0.0)	1 (0.0)	0.995 (0.050)	1 (0.0)	1 (0.0)	1 (0.0)
5-1	1 (0.0)	0.714 (0.0)	0.556 (0.0)	0.987 (0.073)	1 (0.0)	1 (0.0)	1 (0.0)
5-2	0.533 (0.0)	0.615 (0.0)	0.8 (0.0)	0.991 (0.054)	1 (0.0)	1 (0.0)	1 (0.0)
6-0	0.534 (0.151)	0.857 (0.0)	0.857 (0.0)	0.999 (0.014)	1 (0.0)	1 (0.0)	1 (0.0)
6-1	0.59 (0.056)	0.625 (0.0)	0.714 (0.0)	0.991 (0.049)	1 (0.0)	1 (0.0)	1 (0.0)
6-2	0.822 (0.050)	0.625 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
7-0	0.769 (0.0)	0.435 (0.0)	0.588 (0.0)	1 (0.0)	1 (0.0)	0.909 (0.0)	1 (0.0)
7-1	0.469 (0.149)	0.786 (0.0)	0.917 (0.0)	1 (0.0)	1 (0.0)	0.999 (0.008)	1 (0.0)
7-2	0.625 (0.0)	0.417 (0.0)	0.714 (0.0)	0.998 (0.013)	1 (0.0)	1 (0.0)	1 (0.0)
8-0	0.472 (0.099)	0.600 (0.0)	0.750 (0.0)	0.996 (0.023)	1 (0.0)	0.824 (0.057)	1 (0.0)
8-1	0.333 (0.030)	0.714 (0.0)	1 (0.0)	0.998 (0.017)	1 (0.0)	0.928 (0.079)	1 (0.0)
8-2	0.408 (0.109)	0.615 (0.0)	0.8 (0.0)	0.999 (0.011)	1 (0.0)	0.964 (0.052)	1 (0.0)
9-0	0.280 (0.042)	0 (0.0)	0.652 (0.0)	0.999 (0.006)	0.75 (0.0)	0.657 (0.090)	0.195 (0.359)
9-1	0.367 (0.114)	0.636 (0.0)	0.737 (0.0)	0.985 (0.035)	0.992 (0.055)	0.957(0.077)	0.960 (0.086)
9-2	0.568 (0.072)	0.867 (0.0)	0.765 (0.0)	0.996 (0.017)	1 (0.0)	0.923 (0.119)	0.944 (0.030)
10-0	0.44 (0.109)	0.810 (0.0)	0 (0.0)	1 (0.0)	0.85 (0.0)	0.715 (0.031)	0.840 (0.037)
10-1	0.226 (0.041)	0 (0.0)	0.762 (0.0)	0.988 (0.024)	0.789 (0.053)	0.677 (0.093)	0.072 (0.216)
10-2	0.261 (0.068)	0.548 (0.0)	0.654 (0.0)	0.999 (0.006)	0.996 (0.022)	0.760 (0.092)	0 (0.0)
11-0	0.323 (0.051)	0 (0.0)	0.727 (0.0)	0.976 (0.036)	0.955 (0.025)	0.815 (0.024)	0.943 (0.010)
11-1	0.221 (0.037)	0 (0.0)	0 (0.0)	0.980 (0.034)	0.947 (0.078)	0.634 (0.061)	0.872 (0.034)
11-2	0.219 (0.079)	0 (0.0)	0.773 (0.0)	0.960 (0.037)	0.767 (0.036)	0.661 (0.034)	0.220 (0.338)
12-0	0.161 (0.016)	0.68 (0.0)	0.654 (0.0)	0.931 (0.063)	0.630 (0.0)	0.541 (0.044)	0.968 (0.028)
12-1	0.299 (0.077)	0 (0.0)	0 (0.0)	0.899 (0.112)	0.844 (0.153)	0.661 (0.056)	0.944 (0.0)
13-0	0.287 (0.053)	0 (0.0)	0 (0.0)	0.978 (0.028)	0.945 (0.042)	0.664 (0.010)	0.063 (0.201)
13-1	0.313 (0.040)	0 (0.0)	0 (0.0)	0.985 (0.030)	0.856 (0.050)	0.663 (0.013)	0.693 (0.188)
14-0	0.137 (0.021)	0 (0.0)	0 (0.0)	0.873 (0.081)	0 (0.0)	0.630 (0.018)	0.806 (0.150)
14-1	0.342 (0.029)	0 (0.0)	0.441 (0.249)	0.927 (0.062)	0.857 (0.0)	0.419 (0.0)	0.839 (0.023)
15-0	0.067 (0.087)	0 (0.0)	0 (0.0)	0.896 (0.073)	0.140 (0.097)	0 (0.0)	0 (0.0)
15-1	0.186 (0.029)	0 (0.0)	0 (0.0)	0.960 (0.105)	0 (0.0)	0.531 (0.055)	0.646 (0.104)
16-1	0.140 (0.047)	0 (0.0)	0 (0.0)	0.961 (0.044)	0.756 (0.013)	0.509 (0.0)	0.614 (0.223)
16-2	0.275 (0.033)	0 (0.0)	0 (0.0)	0.877 (0.137)	0.426 (0.0)	0.741 (0.011)	0.798 (0.012)
17-0	0.004 (0.025)	0 (0.0)	0.676 (0.0)	0.907 (0.060)	0.119 (0.132)	0 (0.0)	0 (0.0)
sum	15.274 (0.382)	13.478 (0.0)	19.537 (0.0)	33.956 (0.698)	28.619 (0.263)	26.782 (0.270)	26.419 (1.089)
rank	6	7	5	1	2	3	4

Table 3: The Results for **LarT**. The total of the planners' scores is 164.065 (std = 2.702).

terministic over all the other changes. Metric-FF and MIPS were affected by the same two changes as Downward-classic but also by the order of the operators in the domain file. LPG-quality was affected by the same two changes as Downward-classic, in that the number of facts and actions changed, but it is the stablest of all the planners and shows very little change in solution path length across representations.

The main question being investigated in this paper is whether the IPC rankings, including the winner, could change if a different representation was used. The answer is "yes". We see that the three bottom-ranked planners remain the same. In the top four positions, LAMA and Auto2 switch positions. This is caused by Auto2 improving with **InfT**, while LAMA degrades. LAMA reduced its performance from 34.389 to 33.266 while Auto2 actually improved from 33.753 to 34.227.

InfT versus LarT

In **Orig** and **InfT** the table has infinite capacity. In **LarT** the table capacity is finite, but there are the same number of named table locations as blocks so it is always possible to place a block on the table. For example, when there are 4 blocks **LarT** would have 4 locations, P1...P4, each of which can be either clear or occupied (have a block on it). Since the different locations are distinguishable from one another, the state space defined by **LarT** is larger than the state space defined by **Orig** and **InfT**. But since the goal conditions do not specify any table locations, the optimal solution length is the same in all the representations. The domain representation for **LarT** is:

```
(define (domain patblock2)
  (:requirements :strips)
  (:predicates (clear ?pos) (on ?block ?place)
               (handempty) (holding ?block))

   (:action putdown
  :parameters (?block ?place1)
  :precondition (and (holding ?block) (clear ?place1))
```

```
:effect (and (handempty) (on ?block ?place1) (clear ?block)
             (not (holding ?block)) (not (clear ?place1)))))
```

The 4-0 problem representation for **LarT** is:

```
(define (problem BLOCKS-4-0)
(:domain patblock2)
(:objects P1 P2 P3 P4 D B A C )
(:INIT (CLEAR C) (CLEAR A) (CLEAR B) (CLEAR D) (ON C P1)
      (ON A P2) (ON B P3) (ON D P4) (HANDEMPTY))
(:goal (AND (ON D C) (ON C B) (ON B A))))
```

Because of its smaller state space, the expectation going into this experiment was that **InfT** should be preferred by all the planners over **LarT**. Table 3 contains the values from the IPC scoring formula. A run which doesn't finish gets a value of 0 for the planning competition formula. Notice that Metric-FF gets a 0 for problem 9-0 and a number of the larger problems in the **LarT** representation, this is because it gets a segmentation error for the problem. The planners overall do better with **InfT** than with **LarT**. The combined planners' score for **InfT** is 196.085 compared to the score of 164.065 with **LarT**, out of a maximum possible value of 245 (35 problems times 7 planners).

This difference is not especially large, but bigger differences are seen by looking at specific planners. Surprisingly, one planner seems to do better with **LarT** than with **InfT**. LPG-quality's IPC score is 33.956 with **LarT** compared to 33.719 with **InfT**. Note that LPG-quality was only run for 80 runs on problem 16-1 and 17-0 which might cause a higher standard deviation. All the other planners, as expected, do better with **InfT**.

The other trend apparent in Table 3 is that **LarT** tends to increase the nondeterminism of most of the planners. The standard deviations in Tables 2 and 3 show that Downward-classic, LPG-quality, LAMA, Auto1 and Auto2 all have a higher standard deviation with **LarT**. Some of this can be explained by **LarT** being a more difficult representation, and LAMA, Auto1 and Auto2 all become nondeterministic earlier in the problem set. Downward-classic is totally deterministic in all the problems with **InfT**, but with **LarT** it becomes nondeterministic as early as problem 6-0.

The main question being investigated in this paper is whether the IPC rankings, including the winner, could change if a different representation could be used. The answer again is "yes". The bottom rows of Tables 1, 2, and 3 show the rank of each planner using each representation. The three bottom-ranked planners are ranked the same. The top four planners have changed positions. LPG-quality is now first while Auto2 which was first in **InfT** is now in fourth place. The reason for this switch is that LPG-quality is fairly immune to the representation changes, while LAMA, Auto1 and Auto2 all got worse. For instance Auto2 dropped from 34.227 in **InfT** to 26.419 in **LarT**.

The results comparing which planners do better or worse on each representation are explored next. Table 4 shows the number of problems on which one representation or the other finds a shorter solution path in the given time (a win in the IPC), or whether both representations find the same length solution paths. Note that a number of planners failed to find solutions for some of the problems in the **LarT** representation. Most of the planners show a preference for the **InfT** representation, except LPG-quality which prefers **LarT**. Note that although LPG-quality only showed slight improvement from 33.719 on **InfT** to 33.956 on **LarT**, it did receive a higher score on 21 of the 35 problems.

	LarT	Draws	**InfT**
Downward-classic	4	5	26
Metric-FF	6	9	20
MIPS	8	12	15
LPG-quality	21	5	9
LAMA	1	17	17
Auto1	9	10	16
Auto2	0	15	20
total	49	73	123

Table 4: Number of Wins per Representation

Table 4 makes it abundantly clear that neither representation is uniformly better than the other for a given planner; the best representation for each planner (except Auto2) varies from from problem to problem. Therefore it would be advantageous if a planner could change its representation to suit the given problem.

Level Representations

The **LarT** representation opens the way to a totally different representation. In this representation, instead of using the normal "on" representation, a representation is used which specifies for each block, what table location it sits above and at which level (height above the table) it resides. Thus there is no longer any direct connection between two blocks. Figure 1 shows an example of this representation. In the **Orig** representation this would be described as *on(B,A)*, *ontable(A)*, *ontable(C)*, whereas in the level representation this would be *contents(P1,L1,A)*, *contents(P1,L2,B)*, and *contents(P2,L1,C)*.

This representation is very similar to the **LarT** representation and has an optimal solution path which is exactly the same length. One thing to bear in mind is that now, what location the goal stack is on must be specified. This was done systematically. An example might make this clearer. Assume that in the **InfT** representation *on(D,P1)* was in the initial state, where *P1* is the table and the goal state has everything stacked on *D*, such as *on(A,B)*, *on(B,C)*, *on(C,D)* but does not specify where *D* is. In the level representation you must specify where *D* is, so in this case we would spec-

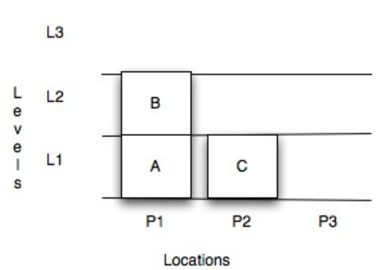

Figure 1: Level Representation

ify *contents(P1,L1,D)*, saying that *D* is on the first level of location *P1*. It would be important to use *P1* instead of *P2* or *P3*, because the optimal solution would be longer than the original infinite table representation (i.e., we would have to move block *D*). If the bottom blocks in the goal state are not directly on a location in the initial state, then a location is picked that is empty in the initial state. Note that the planners might be getting more direction from knowing where the bottom block should go.

There are 3 different level representations. These three representations are again very similar. The **L1** and **L2** representations differ from each other only by the fact that the **L2** representation stores a constant *Z* in the empty levels of each location. The **L3** representation differs from the **L1** representation by the fact that it breaks up the predicate *contents(?location ?level ?block)* into two predicates *contentsa(?block ?level)* and *contentsb(?block ?location)*. Even though these are very small changes, they make a big difference to the planners. For the **L1** representation of the 4-0 problem, Downward-classic produces 85 variables and 219,600 operators; it generates 32,937 nodes and solves the problem in 4.96 seconds. For the **L2** representation of the 4-0 problem, Downward-classic produces 105 variables, but only 200 operators; it generates only 21 nodes and solves the problem in 0.01 seconds. For the **L3** representation of the 4-0 problem, Downward-classic produces only 13 variables but still has 219,600 operators; it generates 483,059 nodes and solves the problem in 5.98 seconds.

The domain representation for **L1** is:

```
(define (domain patblock)
  (:requirements :strips)
  (:predicates (top ?pos ?index) (contents ?pos ?index ?block)
               (notmaxblock ?index) (notzero ?index)
               (holding ?block) (handempty)
               (lower ?index1 ?index2) (notequal ?pos1 ?pos2))

(:action putdown
     :parameters (?pos1 ?pos2 ?block ?index1 ?index2
                  ?newindex1 ?newindex2)
     :precondition (and (holding ?block)
                        (top ?pos2 ?index2)
                        (notmaxblock ?index2)
                        (lower ?newindex2 ?index2))
     :effect
     (and (not (top ?pos2 ?index2))
          (top ?pos2 ?newindex2)
          (contents ?pos2 ?newindex2 ?block)
          (not (holding ?block))
          (handempty))))
```

The 4-0 problem representation for **L1** is:

```
(define (problem BLOCKS-4-0)
(:domain patblock)
(:objects P1 P2 P3 P4 I1 I2 I3 I4 I5 I0 D B A C )
(:INIT (contents P1 I1 C) (contents P2 I1 A)
       (contents P3 I1 B) (contents P4 I1 D) (HANDEMPTY)
       (top P1 I1) (top P2 I1) (top P3 I1) (top P4 I1)
       (notmaxblock I0) (notmaxblock I1) (notmaxblock I2)
       (notmaxblock I3) (notmaxblock I4) (notzero I1)
       (notzero I2) (notzero I3) (notzero I4) (notzero I5)
       (lower I5 I4) (lower I4 I3) (lower I3 I2) (lower I2 I1)
       (lower I1 I0)
       (notequal P1 P2) (notequal P1 P3) (notequal P1 P4)
       (notequal P2 P1) (notequal P2 P3) (notequal P2 P4)
       (notequal P3 P1) (notequal P3 P2) (notequal P3 P4)
       (notequal P4 P1) (notequal P4 P2) (notequal P4 P3))
(:goal (AND (contents P2 I1 A) (contents P2 I2 B)
   (contents P2 I3 C) (contents P2 I4 D))))
```

The domain representation for **L2** is:

```
(define (domain patblock)
  (:requirements :strips)
  (:constants Z)
  (:predicates (top ?pos ?index) (contents ?pos ?index ?block)
               (notmaxblock ?index) (notzero ?index)
               (holding ?block) (handempty)
               (lower ?index1 ?index2) (notequal ?pos1 ?pos2))

 (:action putdown
     :parameters (?pos2 ?block ?index2 ?newindex2)
     :precondition (and (holding ?block)
                        (lower ?newindex2 ?index2)
                        (top ?pos2 ?index2) (notmaxblock ?index2))
     :effect
     (and (not (top ?pos2 ?index2)) (top ?pos2 ?newindex2)
          (not(contents ?pos2 ?newindex2 Z)) (handempty)
          (contents ?pos2 ?newindex2 ?block)
          (not (holding ?block)))))
```

The 4-0 problem representation for **L2** is:

```
(define (problem BLOCKS-4-0)
(:domain patblock)
(:objects P1 P2 P3 P4 I1 I2 I3 I4 I5 I0 D B A C )
(:INIT (contents P1 I1 C) (contents P1 I2 Z) (contents P1 I3 Z)
      (contents P1 I4 Z) (contents P2 I1 A) (contents P2 I2 Z)
      (contents P2 I3 Z) (contents P2 I4 Z) (contents P3 I1 B)
      (contents P3 I2 Z) (contents P3 I3 Z) (contents P3 I4 Z)
      (contents P4 I1 D) (contents P4 I2 Z) (contents P4 I3 Z)
      (contents P4 I4 Z) (HANDEMPTY) (top P1 I1) (top P2 I1)
      (top P3 I1) (top P4 I1) (notmaxblock I0) (notmaxblock I1)
      (notmaxblock I2) (notmaxblock I3) (notmaxblock I4)
      (notzero I1) (notzero I2) (notzero I3) (notzero I4)
      (notzero I5) (lower I5 I4) (lower I4 I3) (lower I3 I2)
      (lower I2 I1) (lower I1 I0)
      (notequal P1 P2) (notequal P1 P3) (notequal P1 P4)
      (notequal P2 P1) (notequal P2 P3) (notequal P2 P4)
      (notequal P3 P1) (notequal P3 P2) (notequal P3 P4)
      (notequal P4 P1) (notequal P4 P2) (notequal P4 P3))
(:goal (AND (contents P2 I1 A) (contents P2 I2 B)
         (contents P2 I3 C) (contents P2 I4 D))))
```

The results for these representations are difficult to explain. The **L1** representation will not run in the 30 minutes allowed for any problem past 4-2 on any of the planners. Metric-FF dies with a segmentation fault at 5-0. LPG-quality will only run on 4-0. MIPS will not run on this representation at all. The Fast Downward family of planners dies at 5-0, because the preprocessing portion which turns PDDL into SAS+(Bäckström 1992) takes more than 30 minutes at that point.

The **L3** representation performs almost as badly as **L1**, MIPS again does not run at all. All the other planners run through problem 4-2, but none will run past that. Because of this poor performance, only the results for **L2** will be analyzed.

The results for the **L2** representation are shown in Tables 5 and 6. Notice that some of the results for LPG-quality and Auto2 are in Table 6, but the totals in Table 5 include those results. The **L2** representation performs worse than **InfT** and **LarT**. The **L2** representation has an IPC score of 103.519 while the **InfT** representation's IPC score is 196.085, the **Orig** representation's IPC score is 192.757 and the **LarT** representation's IPC score is 164.065. It is much worse because all the planners have trouble with this representation, although LPG-quality does much better than all the others. Auto2 can also solve a few of the harder problems. Note problems 15-0 and 15-1 were only run on 80 problems and therefore might have a larger standard deviation.

Some planners do better on **L2** because they do not get confused by the ordering of the subgoals which is a classic problem with planners in the original blocks world representation, for instance in problem 4-0. Note that **L2** was the best representation for LPG-quality on some of the harder problems such as problems 15-0 and 16-2.

	Downward-classic	Metric-FF	MIPS	LPG-quality	LAMA	Auto1	Auto2
4-0	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
4-1	1 (0.0)	0.556 (0.0)	1 (0.0)	0.987 (0.081)	1 (0.0)	1 (0.0)	1 (0.0)
4-2	1 (0.0)	1 (0.0)	1 (0.0)	0.988 (0.084)	1 (0.0)	1 (0.0)	1 (0.0)
5-0	1 (0.0)	1 (0.0)	1 (0.0)	0.955 (0.178)	1 (0.0)	1 (0.0)	1 (0.0)
5-1	0.714 (0.0)	0.5 (0.0)	1 (0.0)	0.801 (0.331)	1 (0.0)	1 (0.0)	1 (0.0)
5-2	0.191 (0.0)	0.571 (0.0)	0.8 (0.0)	0.992 (0.082)	1 (0.0)	1 (0.0)	1(0.0)
6-0	1 (0.0)	1 (0.0)	0.8571 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)	1 (0.0)
6-1	1 (0.0)	1 (0.0)	1 (0.0)	0.928 (0.219)	1 (0.0)	1 (0.0)	1 (0.0)
6-2	0.7143 (0.0)	0.556 (0.0)	0.833 (0.0)	0.979 (0.118)	1 (0.0)	1 (0.0)	1 (0.0)
7-0	0.7692 (0.0)	0.455 (0.0)	0.833 (0.0)	1 (0.0)	1 (0.0)	0.769 (0.0)	1 (0.0)
7-1	0.367 (0.0)	0.647 (0.0)	0 (0.0)	0.965 (0.167)	1 (0.0)	1 (0.0)	1 (0.0)
7-2	0 (0.0)	0 (0.0)	0.769 (0.0)	1 (0.0)	1 (0.0)	0.833 (0.0)	1 (0.0)
8-0	0 (0.0)	0 (0.0)	0.9 (0.0)	0.969 (0.149)	1 (0.0)	1 (0.0)	1 (0.0)
8-1	0 (0.0)	1 (0.0)	0.909 (0.0)	1 (0.0)	0.526 (0.0)	1 (0.0)	1 (0.0)
8-2	0 (0.0)	1 (0.0)	0.8 (0.0)	0.9978 (0.016)	1 (0.0)	1 (0.0)	1 (0.0)
sum	8.755 (0.0)	10.284 (0.0)	12.702 (0.0)	23.564 (1.261)	14.526 (0.0)	14.603 (0.0)	19.085 (0.0)
rank	7	6	5	1	4	3	2

Table 5: Results for Level-2. The total of the planners' scores is 103.519 (std = 1.832).

	LPG-quality	Auto2
9-0	0.394 (0.409)	0
9-1	0.774 (0.300)	0.933 (0.0)
9-2	0.701 (0.419)	0.88 (0.327)
10-1	0.154 (0.271)	0
10-1	0.269 (0.393)	0
10-2	0.149 (0.271)	0
11-0	0.245 (0.379)	0
11-1	0.439 (0.441)	0.838 (0.193)
11-2	0.021 (0.104)	1 (0.0)
12-0	0.188 (0.320)	0.434 (0.427)
12-1	0.124 (0.269)	0
13-0	0.190 (0.108)	0
13-1	0.004 (0.020)	0
14-0	0.956 (0.051)	0
14-1	0.939 (0.069)	0
15-0	0.907 (0.078)	0
15-1	0.882 (0.066)	0
16-1	0.869 (0.070)	0
16-2	0.938 (0.057)	0
17-0	0.030 (0.138)	0

Table 6: Additional Results for Level-2.

The main question being investigated in this paper is whether the rankings in the planning competition, including the winner, could change if a different representation could be used. The answer to this question is "yes". The three bottom-ranked planners have changed ordering for the first time with the **L2** representation, with Downward-classic being worse than Metric-FF for the first time. LPG-quality does the best because it can solve all the problems within 30 minutes. Auto2 does better than LAMA and Auto1 because it can finish more problems.

Comparing All the Representations

How does the representation affect the nondeterminism of the planner? The **Orig** representation has a standard deviation of 0.337 while the **InfT** representation has a standard deviation of 0.554. The LPG-quality planner and to a lesser extent LAMA and Auto1 are responsible for the nondeterminism in both these representations. In the **LarT** representation, there is a higher standard deviation of 2.702, because all the planners except Metric-FF and MIPS are now nondeterministic. In the larger problems some of this nondeterminism is caused by the planners not finishing, but not all of it. In the **L2** representation, no planner but LPG-quality will run past problem 9-0 (except for Auto1 in problems 11-1 through 12-0), so it is hard to assess this representation.

Table 7 summarizes the results of each planner using each representation: the second to fifth columns in this table are the "sum" rows from the previous tables. The last column shows the score each planner would obtain if it used the best representation for each problem. It was the case for every planner that each representation had problems for which it was the best representation for that planner (even **L2**)— Table 8 shows which representation is best for each planner and problem. All the planners except LAMA show improvements by changing representation on a problem-by-problem basis. Downward-classic, Metric-FF, and MIPS improve the most, possibly because they were the worst performing planners. The other planners improve less, possibly because of a ceiling effect. Thus changing representation to suit the problem is a clear win, providing a planner can determine accurately which representation is best.

Conclusions

It is clear from this work that the representation used makes a large difference to the planner's ability to solve a problem. We also clearly saw that some planners seemed to be more sensitive to the type of representation than others. But even

	Orig	InfT	LarT	L2	Best
Downward-classic	19.438	20.119	15.274	8.755	22.704
Metric-FF	16.919	19.970	13.478	10.284	23.398
MIPS	25.604	25.832	19.537	12.702	27.902
LPG-quality	33.898	33.719	33.956	23.564	34.159
LAMA	34.389	33.266	28.619	14.526	34.389
Auto1	28.751	28.952	26.782	14.603	30.512
Auto2	33.753	34.227	26.419	19.085	34.395
total	192.757	196.085	164.065	103.519	207.583

Table 7: **Best** shows the IPC score if the best representation is used for each problem.

	Downward-classic	Metric-FF	MIPS	LPG-quality	LAMA	Auto1	Auto2
4-0	L2	L2	All	L2	All	All	All
4-1	All	Orig, InfT	All	L2	All	All	All
4-2	All	Orig, InfT, L2	All	LarT	All	All	All
5-0	All	Orig, InfT, L2	All	LarT	All	All	All
5-1	InfT, LarT	Orig	L2	Orig	All	All	All
5-2	InfT	InfT	Orig, LarT, L2	Orig	All	All	All
6-0	Orig, L2	Orig, L2	All	L2	All	All	All
6-1	L2	L2	L2	LarT	All	All	All
6-2	LarT	InfT	Orig, InfT, LarT	LarT	All	All	All
7-0	Orig, InfT	InfT	L2	All	All	Orig, InfT	All
7-1	Orig	Orig	Orig, LarT	LarT	All	Orig, InfT, L2	All
7-2	LarT	Orig, InfT	L2	L2	All	Orig, InfT, LarT	All
8-0	LarT	Orig, LarT	L2	Orig	All	Orig, InfT, L2	All
8-1	Orig	L2	LarT	InfT, L2	Orig, InfT, LarT	Orig, InfT, L2	All
8-2	Orig	L2	All	LarT	All	Orig, InfT, L2	All
9-0	InfT	Orig	Orig, InfT, LarT	Orig	Orig, InfT	Orig, InfT	Orig, InfT
9-1	InfT	InfT	Orig, LarT	Orig	Orig, InfT	Orig, InfT	Orig, InfT
9-2	InfT	Orig	Orig, LarT	Orig, InfT	Orig, InfT, LarT	Orig	Orig, InfT
10-0	Orig	Orig, LarT	Orig, InfT	InfT, LarT	Orig, InfT	LarT	Orig, InfT
10-1	InfT	None	Orig, InfT, LarT	InfT	Orig, InfT	Orig, InfT	Orig, InfT
10-2	InfT	InfT	InfT	Orig, InfT, LarT	Orig, InfT	LarT	Orig, InfT
11-0	InfT	Orig, InfT	Orig, LarT	InfT	Orig, InfT	LarT	Orig, InfT
11-1	InfT	InfT	InfT	LarT	Orig, InfT	LarT	InfT
11-2	InfT	Orig	Orig, LarT	InfT	Orig, InfT	InfT	Orig, InfT
12-0	Orig, InfT	Orig	InfT	LarT	Orig, InfT	Orig, InfT	Orig, InfT
12-1	InfT	InfT	Orig	Orig	Orig, InfT	InfT	InfT
13-0	InfT	InfT	InfT	LarT	Orig	Orig	Orig
13-1	Orig	InfT	Orig	LarT	Orig	LarT	Orig, InfT
14-0	InfT	InfT	InfT	InfT, LarT	Orig, InfT	LarT	Orig, InfT
14-1	Orig	InfT	InfT	LarT	Orig, InfT	InfT	InfT
15-0	Orig	None	Orig, InfT	L2	Orig, InfT	Orig, InfT	InfT
15-1	InfT	None	InfT	Orig	Orig	Orig	Orig
16-1	InfT	InfT	InfT	Orig	Orig	Orig, InfT	Orig
16-2	Orig	None	InfT	L2	Orig, InfT	LarT	Orig, InfT
17-0	InfT	None	Orig, LarT	LarT	Orig	Orig	InfT

Table 8: What is the Best Representation For Each Problem

for the LPG-quality planner (which was the least affected by any of the representation changes), it was the case that for some problems the planner produced better results for some representations than for others.

Specifically, the order of planners, as determined using the scoring formula from the IPC, changes when different representations for BlocksWorld problems are used. Different rankings for the planners are found for all the representations explored and three different planners are declared the winner, as is shown in Table 9.

Acknowledgments

Thanks to Malte Helmert for all his help with the Fast Downward family of planners. He gave freely of his time and was a huge help. We would also like to thank Derek Long and Amanda Cole for their assistance with the other planners and

	Downward-classic	Metric-FF	MIPS	LPG-quality	LAMA	Auto1	Auto2
Orig	6	7	5	2	1	4	3
InfT	6	7	5	2	3	4	1
LarT	6	7	5	1	2	3	4
L2	7	6	5	1	4	3	2

Table 9: Ranking of Planners in each Representation

for discussion about this work. Robert Holte thanks the Natural Sciences and Engineering Research Council (Canada) for its financial support of this research.

References

Bäckström, C. 1992. Equivalence and tractability results for SAS+ planning. In *Proceedings of the 3rd International Conference on Principles on Knowledge Representation and Reasoning (KR-92)*.

Edelkamp, S. 2003. Taming numbers and durations in the model checking integrated planning system. *Journal of Artificial Intelligence Research* 20(1):195–238.

Gerevini, A.; Saetti, A.; and Serina, I. 2003. Planning through stochastic local search and temporal action graphs in lpg. *Journal of Artificial Intelligence Research* 20(1):239–290.

Ghallab, M.; Howe, A.; Knoblock, C.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D.; and Wilkens, D. 1998. PDDL-the planning domain definition language.

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

Hoffmann, J. 2003. The Metric-FF planning system: Translating “ignoring delete lists” to numeric state variables. *Journal of Artificial Intelligence Research* 20(1):291–341.

Hutter, F.; Hamadi, Y.; Hoos, H.; and Leyton-Brown, K. 2006. Performance prediction and automated tuning of randomized and parametric algorithms. *Principles and Practice of Constraint Programming-CP 2006* 213–228.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research* 39(1):127–177.