AI Magazine Volume 22 Number 3 (2001) (© AAAI)

AIPS'00 Planning Competition

The Fifth International Conference on Artificial Intelligence Planning and Scheduling Systems

Fahiem Bacchus

■ The planning competition has become a regular part of the biennial Artificial Intelligence Planning and Scheduling (AIPS) conferences. AIPS'98 featured the very first competition, and for AIPS'00, we built on this foundation to run the second competition. The 2000 competition featured a much larger group of participants and a wide variety of different approaches to planning. Some of these approaches were refinements of known techniques, and others were quite different from anything that had been tried before. Besides the dramatic increase in participation, the 2000 competition demonstrated that planning technology has taken a giant leap forward in performance since 1998. The 2000 competition featured planning systems that were orders of magnitude faster than the planners of just two years prior. This article presents an overview of the competition and reviews the main results.

The Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS'00) was held in April 2000. AIPS is a biennial event, and in 1998, it featured its first planning competition, organized by Drew McDermott.[1] The competition proved to be a successful and interesting addition to the conference, and it was decided to make the competition a regular feature of AIPS. I was asked to organize the competition for the 2000 conference, and in this article, I describe the results of this competition.

First, I briefly describe the manner in which the competition was structured along with a discussion of some of the various trade-offs involved in its structure. This discussion will be of greatest interest to those in other areas who might be considering developing similar competitions. Second, I present a description of the problem domains used in the competition. Finally, I summarize the results of the competition. Besides the summary presented here, all the raw data generated during the competition are available online. The hope is that these data can be further analyzed by researchers to provide additional insights into planner performance.

The Competitors

Fifteen different teams participated in the 2000 competition. The participants are listed in table 1. The systems themselves were quite diverse in design, and as explained later, they fell into two different tracks: (1) fully automated planners and (2) hand-tailored planners that could take as input additional domain-specific information. More information about many of these systems appears in the subsequent articles. Here, I provide a short summary of the approaches used.[2]

Heuristically guided forward-chaining planners: Plans are searched for by chaining forward from the initial state, guided by domain-independent heuristic estimates of the distance to the goal. Systems using this approach include FF; GRT; HSP2; and, to some extent, STAN.

Binary decision diagram (BDD)–based planners: These planners utilize BDDs, a representation originally developed in the verification literature, to compactly represent a set of states in the search space. Systems using this technique include MIPS, PROPPLAN, and BDDPLAN.

Regression-progression planners: SYSTEM R uses a formally correct version of the original

Team	System
Joerg Hoffmann, Albert Ludwigs University, Germany	FF
Ioannis Refanidis, Ioannis Vlahavas, and Dimitris Vrakas, Aristotle University, Greece	GRT
Fangzhen Lin, The Hong Kong University of Science and Technology, Hong Kong	SYSTEM R
Stefan Edelkamp and Malte Helmert, University of Freiburg, Germany	MIPS
Jana Koehler, Schindler Lifts Ltd. Switzerland; Joerg Hoffmann and Michael Brenner, University of Freiburg, Germany	IPP
Hector Geffner and Blai Bonet, Universidad Simon Bolivar, Venezuela	HSP2
Michael Fourman, University of Edinburgh, United Kingdom	PropPlan
Derek Long and Maria Fox, University of Durham, United Kingdom	STAN
Henry Kautz, University of Washington; Bart Selman and Yi-Cheng Huang, Cornell University	BLACKBOX
Biplav Srivastava, Terry Zimmerman, BinhMinh Do, XuanLong Nguyen, Zaiqing Nie, Ullas Nambiar, and Romeo Sanchez, Arizona State University	ALTALT
Yannick Meiller and Patrick Fabiani, ONERA—Center of Toulouse, France	TOKENPLAN
Hans-Peter Stoerr, Dresden University of Technology, Germany	BDDPLAN
Dana Nau, Hector Munoz-Avila, Yue (Jason) Cao, and Amnon Lotem, University of Maryland	SHOP
Jose Luis Ambite, Craig Knoblock, and Steve Minton, University of Southern California Information Sciences Institute	PBR
Jonas Kvarnstrom, Patrick Doherty, and Patrik Haslum, Linkoping University, Sweden	TALPLANNER

Table 1. The Competitors.

regression-progression algorithm of STRIPS.

GRAPHPLAN-based planners: IPP and, to some extent, STAN base their planners on GRAPHPLAN.[3]

SAT-based planners: BLACKBOX operates by converting a planning problem to a satisfiability problem and then applies SAT-solver technology.

Petri net planners: TOKENPLAN uses a novel approach that uses colored Petri nets.

Hierarchical task network (HTN) planners: SHOP is an HTN planner that uses a novel left-to-right plan-elaboration scheme.

Plan rewriting: PBR takes the output of a simple, low-quality planner and rewrites the plan to produce a better plan.

Temporal logic search control: TALPLANNER uses the TLPLAN approach to planning,[4] where additional domain-specific control is specified using a temporal logic. These logical expressions are used to prune the search space while a plan is being searched for.

The Competition's Structure

The guiding principle of the AIPS competition is to be as inclusive as possible, start slow, and develop and expand the range of the competition as time goes on. The first competition, held in 1998, had two separate tracks and a total of five competitors. Building on this foundation, organizers were able to attract 15 competitors to the 2000 competition. Clearly, the first competition had served to pique interest in the research community. In the 1998 competition, an attempt was made to include HTN-style planners in a separate track. Such planners use additional domain-specific information, information that goes well beyond the simple specification of the primitive actions used in domain-independent planners.[5] A serious attempt was made to supply a precise specification of the input such planners could take. However, the resulting specification was complex, and it would have been time consuming to adapt existing HTN planners to it. Another difficulty was that the specification only addressed the HTN style of supplying additional domain-specific information.

In the 2000 competition, I decided to go a different direction. In particular, I am convinced by my own work in planning that domain-specific information must be used if we want to reach the next level of planning performance; so, I felt it was important to include planning systems that take advantage of such information. At the same time, however, I felt that it was inappropriate to prescribe the manner in which this knowledge should be specified or used by a planning system. In fact, one of the major complexities in mounting a competition of this sort is that considerable care must be taken to minimize building assumptions about solution techniques into the competition's structure. A competition should not prescribe how the problem is to be solved; it should try to allow for approaches never thought of before.

To address the problem of domain-specific information, we allowed the planning systems to compete in two separate tracks: (1) planners that took as input a specification of the primitive operators in the domain only and (2) planners that could take extra domain-specific information. Organizers made no attempt to limit the kinds of extra information the planning systems of the second track could use; we only required that they produce correct plans: executable sequences of primitive actions that achieved the goal. In particular, it would have been possible for a competitor in the second track to have written a special program for each domain. None of the competitors chose to go this route, however. Rather in this hand-tailored track, we had systems that used domain-specific heuristics, search control specified as expressions of temporal logic, HTNs, and plan rewriting. It is an interesting question whether such a competition should prohibit solutions based on programming. My personal opinion is that such "solutions" should be permitted but that adequate ways of evaluating planning systems must also include some measurement of the development time (not of implementing the planning system but the time required to configure it to solve a new domain). As pointed out during one of the panels at the conference, for commercial applications of planning, ad hoc programs are the competition. AI technologies such as AI planning must eventually prove themselves to be more cost effective than writing ad hoc programs, and in practice, once a certain minimal level of performance is achieved, development time is probably the most important cost measure.

During the competition, there was a relatively limited time available to the competitors to get their system ready for a new domain. Typically, the new domain would be released, and the competitors were given a week to produce their results. We measured the time required to solve the planning problems and the length of the plan, mainly using the time required to rank the planners. Thus, for the configurable planners, we did not have a good measure of development time. We hope to improve on this in future competitions.

In general, no serious attempt was made to measure the quality of the plans produced in either of the competition tracks. Only the length of the plan was measured because there was no facility for specifying the cost of different solutions. In fact, most of the planning systems were incapable of taking such a metric into account during planning. I believe that the whole notion of trying to measure plan quality is flawed unless different cost metrics can be specified for each planning problem, and the planning system can modify its behavior based on the supplied metric. Different situations can require radically different cost metrics; there can even be situations where longer plans are to be preferred (for example, when earning an hourly wage for a fixed task). Thus, if a planning system generates the same plan irrespective of the cost metric, any attempt to measure the quality of the plans it produces will yield a number that will be based entirely on whether the planner had a lucky match with the cost metric. This measurement is hardly useful for judging the quality of the plans the planner is capable of producing. Thus, for this competition, we stayed with the simple notion of plan length, with the caveat that even this element should not be taken too seriously. For the 2002 competition, there are plans to have a track with problems involving metric quantities, which will provide the necessary substrate to start dealing appropriately with cost metrics.

All the test domains were specified using a subset of the PDDL language (the exact form of which is contained in a document stored on the competition's home page).[6] Some of the test domains were specified solely using STRIPS operators, and some used the more expressive ADL operators.[7] Some of the planning systems were not capable of handling ADL operators, so to be as inclusive as possible, an attempt was made to use the simplest form of operator feasible for each domain.

Organization

To run the competition, a single machine was provided to the competitors. This machine was located at the University of Toronto. It was a relatively powerful workstation—a 500-megahertz Intel PENTIUM III with 1 gigabyte of random-access memory—but certainly nothing extraordinary. The competition itself consisted of five rounds, with rounds 1 through 4 being run by remote access to the machine in Toronto. Each round consisted of a specific problem domain, along with a suite of test problems. A harder set of test problems were made available for those planners in the domain-specific track because typically these planners are able to solve larger problems. The domain and problem suites were released at particular times, and the competitors were required to produce their results within a set time period. To make the process manageable, the competitors were responsible for running

The guiding principle of the AIPS competition is to be as inclusive as possible, start slow, and develop and expand the range of the competition as time goes on.

their own systems on the problem suite and reporting their results in a particular format. In addition, as a check against fraudulent reporting, they were required to make available a batch file that I could run to ensure that their system was in fact generating solutions in approximately the times reported. I would randomly pick a few systems after each round and double check the reported results. I am happy to report that there was never even a hint of fraudulent reporting!

After the results had been submitted, we ran the reported solutions through a plan checker. For the most part, the plans were correct, but on more than one occasion, incorrect plans were detected by the plan checker. Most of these instances arose from software bugs—many of the systems were in a state of continual development during the course of the competition. The competition was designed to illuminate the potential of alternate algorithmic approaches to planning. Hence, a balance had to be achieved between testing the competitor's abilities at building software and testing their abilities at designing innovative solutions to the planning problem. Thus, in each of these cases, the competitor was given an opportunity to see if the invalid plans arose from a simple bug in their system, fix the bug, and rerun the test suite. If the competitor was unable to do this in a short timeframe, all the results for the domain on which an incorrect plan was found were disqualified.

The improvement in performance over the two-year period is remarkable.

The Problem Domains

There were 5 problem domains and on the order of 50 to 100 test problems in each domain. The test suite contained a sequence of problems of increasing size and complexity. The first two domains were standard domains, designed to help the competitors become accustomed with the mechanics of the competition. The first domain was the standard four-operator blocks-world domain (all STRIPS operators). The blocks world is probably the oldest AI planning domain and has come under increasing criticism as a prototypically uninteresting and unrealistic toy domain, something AI should move beyond. However, in experiments I conducted with the planners in the 1998 competition, and older planning systems, I found that none of them were able to solve blocks-world problems effectively. Typically, they hit a computational cliff once they got to problems containing 13 to 14 blocks. However, the blocks world has a simple and natural structure that intelligent systems should be able to take advantage of, just as humans can. I have no objection to discarding a toy problem that contains no natural structure or is easy to solve, but it seems to be scientifically unsatisfactory to discard a toy problem simply because it is embarrassing to admit that it can't be solved with current techniques. This said, I am pleased to report that now the field can in good conscience leave the blocks world behind; as the reader can see, many of the planners in the 2000 competition performed well on this domain.

The second domain was the logistics world, a domain that was used in the 1998 competition. This domain encodes a very simplified form of logistics planning. In particular, there are a collection of packages at various locations in various cities. The packages can be transported between locations in the same city by trucks and between airports in different cities by airplanes. The goal is to move the various packages from their initial location to a final location. The domain contains six STRIPS operators, and its main complexity comes from the size of the state space. On some of the larger problems, the search space can have branching factors as large as 1000 and required plans containing over 200 steps. This domain proved to be hard for the systems in the 1998 competition but much easier for the systems in the 2000 competition. The improvement in performance over the two-year period is remarkable.

The third domain was the schedule world, a simplified machining domain. The domain consists of nine machining operators that do things such as paint, grind, and drill an object. The goal is to transform some collection of objects by performing various machining operations on them. Some operations destroy the effects or preconditions of other operations; for example, grinding destroys the paint on an object, and rolling an object makes it hot, after which there is no way to polish it. Furthermore, the domain has a simple notion of time: An object cannot be operated on by two different machines at the same time, and a machine cannot operate on two different objects at the same time. A correct plan has to sequence these operations in the right way and account for the resource constraints. Most of the operators in this domain are ADL operators with conditional and universal effects.

The fourth domain was an encoding of the solitaire card game *FreeCell*. This free computer game comes with every WINDOWS operating system—it has been claimed to be the world's most popular computer game.[8] The game layout is shown in figure 1. The four squares at the upper right are the home cells, one for each suit. The goal is to move all the cards home in

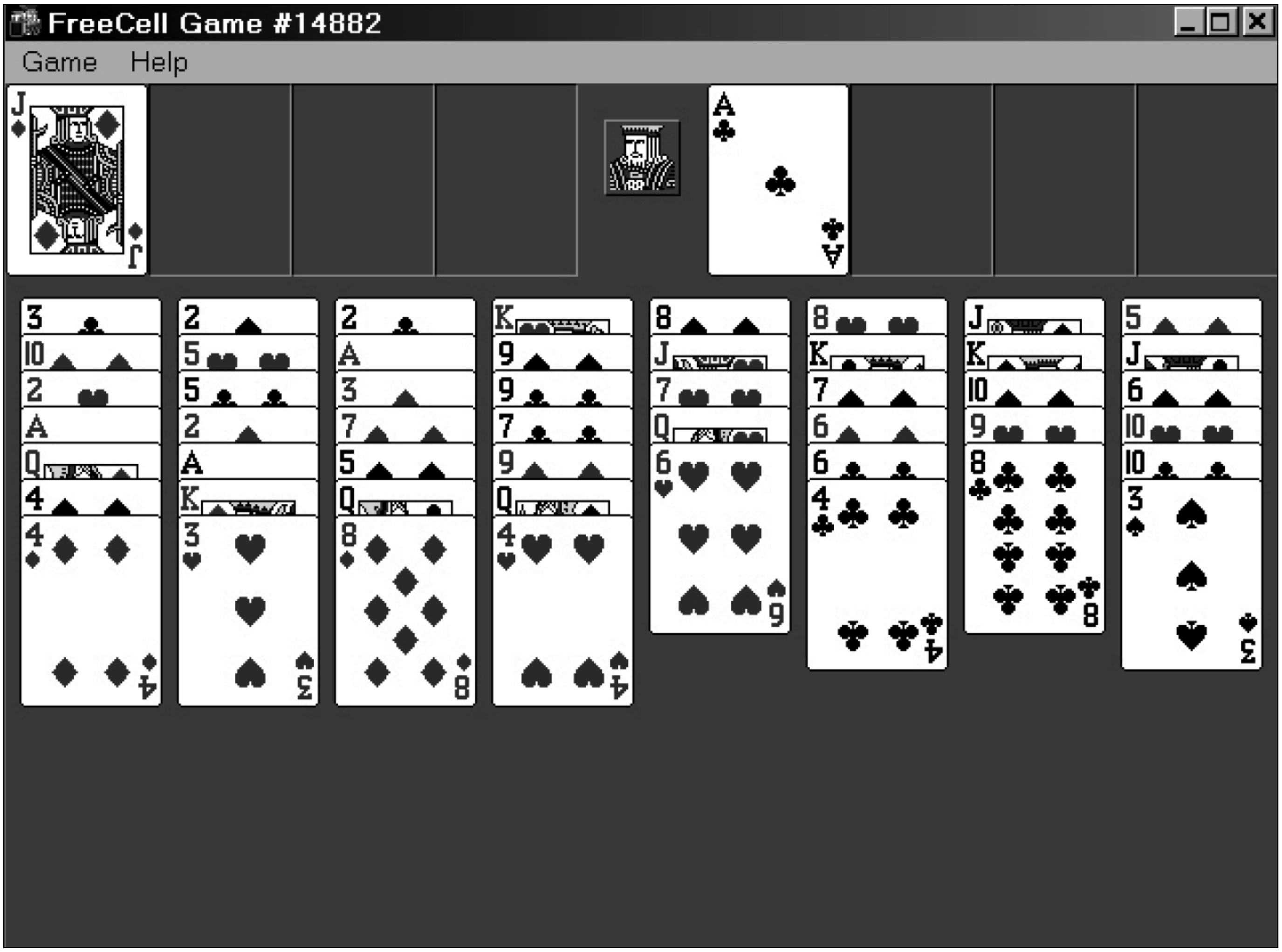

Figure 1. The FreeCell Domain.

the order ace, two, …, King. The four squares at the upper left are the free cells. They act as buffers, a temporary space to hold as many as four different cards.

The main board contains eight columns of cards, randomly configured. In this section of the board, only exposed cards can be moved. They can be moved to a home cell if all previous cards are already there (for example, in figure 1 if any of A♠, A♥, A♦, or 2♣ were exposed, they could be moved home); moved to a free cell if there is space available; or moved in a legal solitaire move, that is, moved on top of a next-highest card of opposite color (for example, in figure 2, the 3♠ could be moved on top of the 4♦, thus exposing the 10♣). This domain was encoded using 10 STRIPS operators. Starting with a fixed set of initial games, simpler versions of each game were generated by removing the top *k* cards of every suit. For example, by removing the face cards in this configuration, we create a simpler game containing only 40 cards. The complexity of the game reduces considerably as the number of cards is decreased: Fewer deadlocks are created. It can be proved that this domain contains unsolvable games, although they are relatively rare.[9]

The fifth and final domain was an encoding of a sophisticated elevator controller produced by Schindler Lifts Ltd.[10] This controller, in operation in some of the world's tallest buildings, controls a bank of elevators. Users go to a central station and punch in their floor request. The controller then tells them which elevator they should use. In this way, the controller attempts to minimize the delay time by scheduling an entire bank of elevators together instead of having each elevator run separately. The controller also allows for various special situations, for example, VIP users or sets of users who should

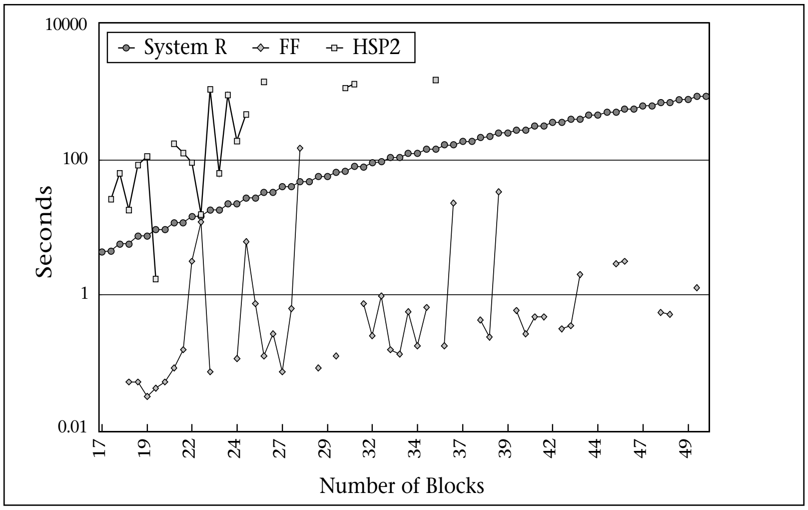

Figure 2. Fully Automated Blocks Central Processing Unit Time.

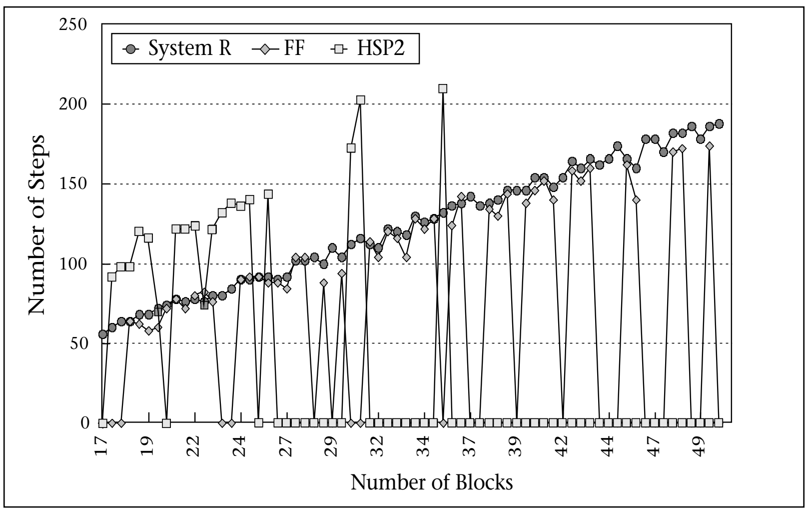

Figure 3. Fully Automated Blocks Plan Length.

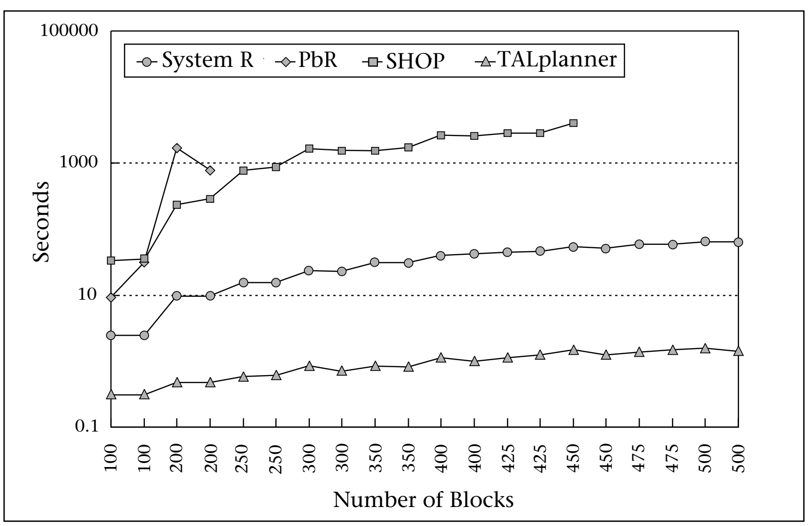

Figure 4. Hand-Tailored Blocks Central Processing Unit Time.

not be riding an elevator together. The planning domain was configured in a number of different ways to accommodate the representational power of different planning systems.[11]

Results

I present here only a summary of the main results. A set of presentation slides are available on the competition's web site that contain a number of other performance graphs. Even more detailed information is available in a set of EXCEL spreadsheets and a set of log files that contain all the data collected during the competition.

In the blocks world, the problems consisted of transforming a random configuration of *N* blocks to another random configuration. In this domain, the number of blocks was a natural measure of problem complexity. For many of the fully automatic planning systems, 14 to 15 blocks proved to be the limit of their reasonable performance. However, there were three fully automated systems that were able to solve much larger problems. Figure 2 shows the time performance of SYSTEM R, FF, and HSP2 on these larger problems. The horizontal axis shows the number of blocks in the problem, starting at 17 and advancing to 50. We see that SYSTEM R is a very robust solver of blocks-world problems, and FF and HSP2 are both able to solve large problems but with *dropouts* (a missing segment indicates that the problems of this size were not solved). Figure 3 shows the length of the plans generated in solving these problems (the dropouts to zero occur when no plan is found). We see that HSP2 sometimes generates plans that are much longer than necessary, but FF, which uses a similar forward-chaining technology, is able to find short plans. SYSTEM R does well both in terms of plan length and time to find a plan (it also has no dropouts in this range). In this domain, it was able to generate plans containing almost 200 steps in about 900 seconds.

The hand-tailored planners were able to demonstrate impressive performance in this domain: They were able to take, as input, information about the special structure of the blocks world that makes this domain easy. The performance of the best hand-tailored planners on a set of much larger problems is shown in figure 4. These planners all generated plans of similar length, with the plans for the 500-block problems almost 2000 steps in length. It is worth noting that TALPLANNER was able to solve these large problems in about 1.5 seconds. SYSTEM R in this figure is configured to use additional domain-specific heuristic information;

thus, it was able to increase its performance significantly over its domain-independent mode (see figure 2).

In the logistics world, the problems consist of transporting a varying number of packages. The number of cities, trucks, and airplanes were increased linearly with the number of packages, thus increasing the size of the search space even further. The packages were randomly distributed among the possible cities and locations in these cities in both the initial and the goal state. In this domain, a number of the fully automated systems were able to achieve good performance. Figures 5 and 6 show the performance of the top six systems on some of the larger logistics problems. We see that except for SYSTEM R, there is very little to choose between the other five planning systems (GRT, HSP2, STAN, MIPS, and FF) in terms of time or plan length. FF and MIPS were the fastest, but STAN consistently generated the shortest plans. In this domain, plans containing more than 200 steps were being generated in about 100 seconds.

The performance of the top three hand-tailored planners is shown in figures 7 and 8. Again, we see that at the price of additional domain-specific information, performance can be increased. SHOP and TALPLANNER generate plans of an almost identical length and significantly shorter than SYSTEM R. In terms of planning time, however, TALPLANNER has a significant edge over the other two systems.

The schedule world posed an additional difficulty for the planners. It required ADL operators. Hence, of the fully automated planning systems, only the systems MIPS, FF, HSP2, IPP, PROPPLAN, and BDDPLAN were able to deal with this domain. The problems in this domain are parameterized by the number of objects to be machined. Each object was assigned a random set of properties in the initial state and required to have a random collection of new properties in the goal state. In this domain, the techniques used by FF were clearly superior, as figure 9 demonstrates. The graph showing the length of the plans is omitted because the plans generated were all of approximately the same length. In figure 10, the performance of the hand-tailored planners is shown. Only three systems were able to generate correct plans in this domain: (1) BDDPLAN, (2) PBR, and (3) TALPLANNER. Of them, again TALPLANNER was superior both in terms of planning time and plan length.

The *FreeCell* domain as a planning problem arose from a class project that I set in my introductory AI class. I give the students access to my TLPLAN system and ask them to find a prob-

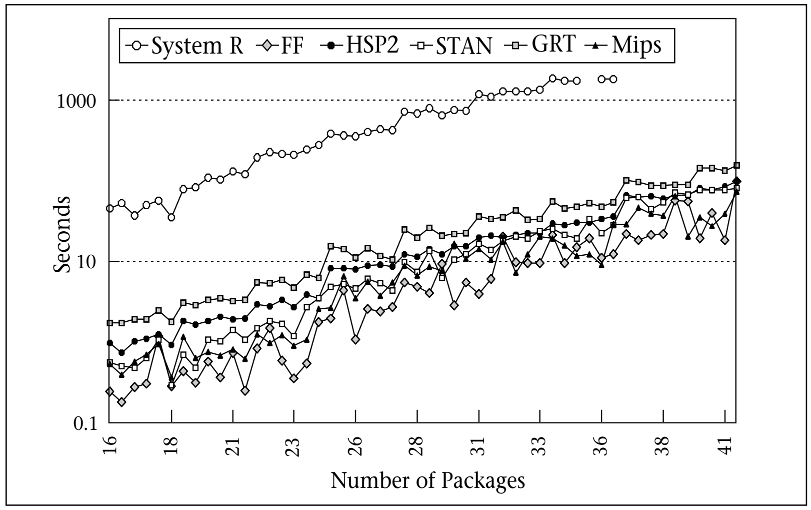

Figure 5. Fully Automated Logistics Central Processing Unit Time.

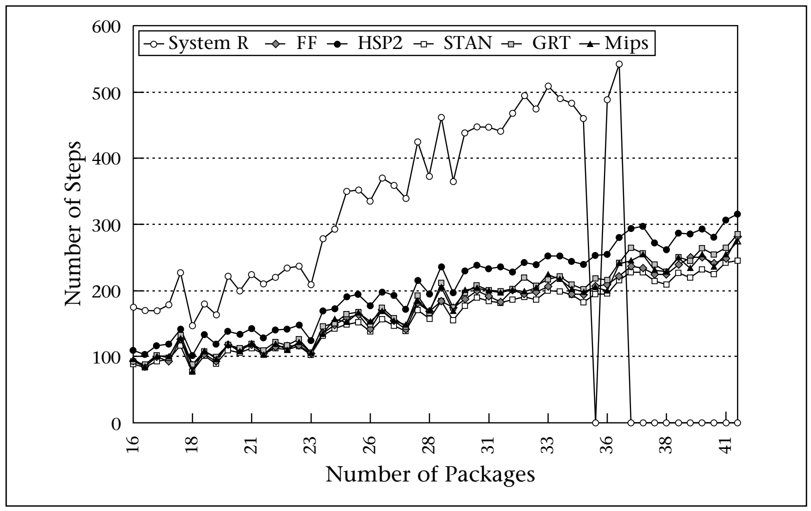

Figure 6. Fully Automated Logistics Plan Length.

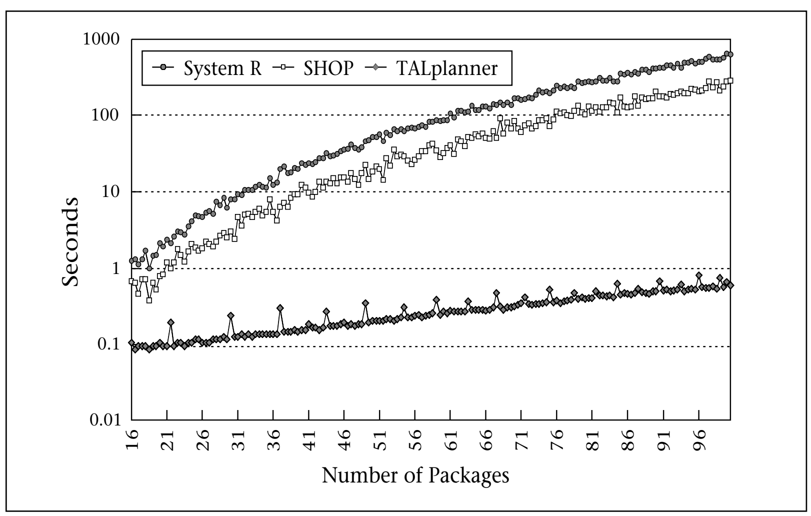

Figure 7. Hand-Tailored Logistic Central Processing Unit Time.

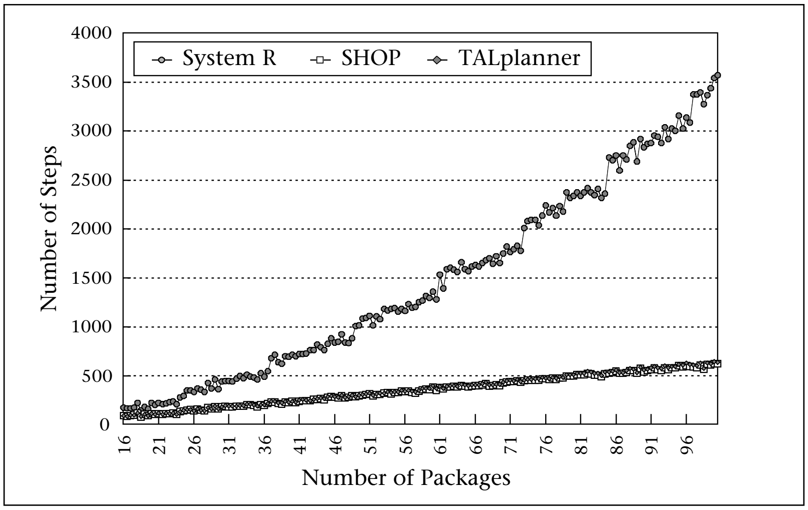

Figure 8. Hand-Tailored Logistics Plan Length.

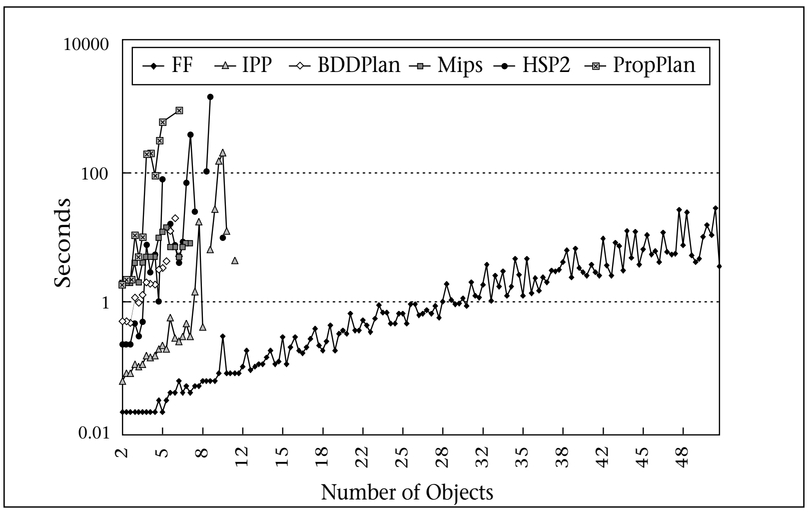

Figure 9. Fully Automated Schedule Central Processing Unit Time.

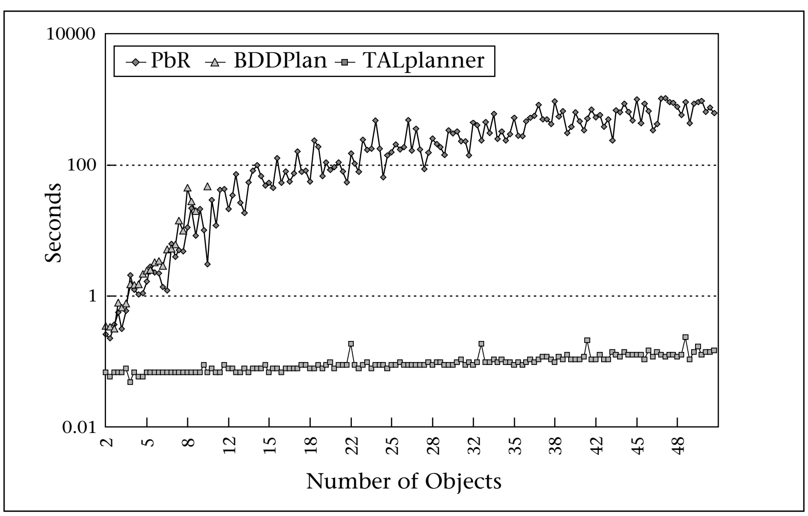

Figure 10. Hand-Tailored Schedule Central Processing Unit Time.

lem they can encode as a planning problem (and solve by writing appropriate temporal control logic). Invariably, they display far more imagination at this task than I have! The *Free-Cell* domain is one of the more interesting domains that has arisen from these projects. As described previously, problems were parameterized by the number of cards in each suite and were created by reducing some standard full-deck problems. This domain proved to be quite challenging for the planning systems, and interestingly, the hand-tailored planners fared no better on it than the fully automatic planners. In experiments, I have been able to get my own TLPLAN system to solve all but 75 of the 32,000 standard full-deck problems (in an average of about 25 seconds and 100 plan steps for each problem). However, my solution required a fair amount of experimentation to find a good heuristic; so, this domain does demonstrate that the task of finding useful domain-specific information can be difficult. Figure 11 shows the planners' performance in this domain. Among the fully automatic planners, STAN, FF, and MIPS were the best performers in terms of solution time. However, STAN did tend to generate considerably longer plans, and MIPS tended to generate the shortest plans. These plans were about 125 steps long for the full-deck problems. With the hand-tailored planners, TALPLANNER was again the fastest planner, but this time, it was not significantly faster than the fastest fully automatic planner (figure 12). Furthermore, figure 13 shows something telling about its performance in this domain: It is generating plans that are an order of magnitude longer then necessary (almost 5000 steps for some of the full-deck problems!). Note that in this graph, the *y*-axis is in a log scale.

I won't include the graphs derived from the final domain, the elevator domain; they are available on the competition's web site. However, it is interesting to note that for the most part, the problems in this domain were quite easy. The real difficulty in this domain came from representing all the features of the domain. The domain contained complexities that were beyond what most of the planning systems could represent. For this reason, various versions of the domain were created, from a very simplified STRIPS version to a more realistic ADL version. The most complex version included the constraint that no more than six passengers could be on an elevator at the same time. Only two of the systems could deal correctly with this most general version: TALPLANNER and PROPPLAN.

The Prize Winners

Two separate prizes were given out, courtesy of the Cel Corporation (CelCorp) and Schindler Lifts Ltd. CelCorp provided some cash prizes for performance, and Schindler Lifts provided a prize for performance in the elevator domain. The prize winners for this domain were FF and TALPLANNER. The prizes for performance, however, were much more difficult to evaluate because there was considerable variance in performance across domains. These winners became a judgment call on my part, and I chose two groups of planning systems; a set of distinguished planners; and two exceptional planners, one from each track. My selections were as follows:

Honorable Mention

1. STAN, Derek Long and Maria Fox, University of Durham, United Kingdom
2. HSP2, Hector Geffner and Blai Bonet, Universidad Simon Bolivar, Venezuela
3. MIPS, Stefan Edelkamp and Malte Helmert, University of Freiburg. Germany
4. SYSTEM R, Fangzhen Lin, The Hong Kong University of Science and Technology, Hong Kong

Exceptional Performance

1. FF, domain-independent planning system, Joerg Hoffman, Albert Ludwigs University, Germany
2. TALPLANNER, hand-tailored planning system, Jonas Kvarnstrom, Patrick Doherty, and Patrik Haslum, Linkoping University, Sweden

Congratulations to these researchers.

Conclusion

The planning competition has proved to be a useful mechanism for furthering interest and research in planning. It is not intended to take the place of normal peer-reviewed research results published in conference proceedings and journals. In particular, there is a considerable gap between ideas worthy of publication and notions that are ready to be used in a robust and efficient planning system. Research that is capable of increasing our understanding or contributing novel ideas to the field is essential if we want to build better planning systems 2, 3, 5, or even 20 years in the future. The competition provides a useful additional test of some of our research ideas, which if used properly can generate excitement, increase interest, and tell us a great deal about the potential of various approaches to the planning problem.

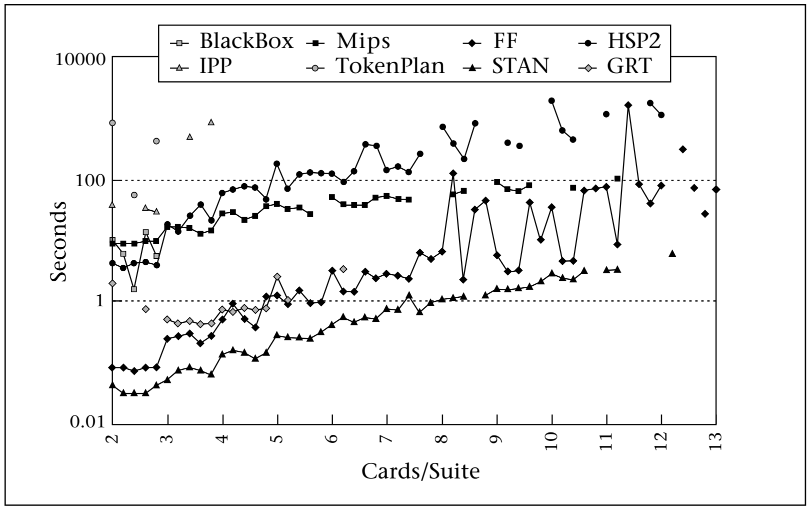

Figure 11. Fully Automatic FreeCell Central Processing Unit Time.

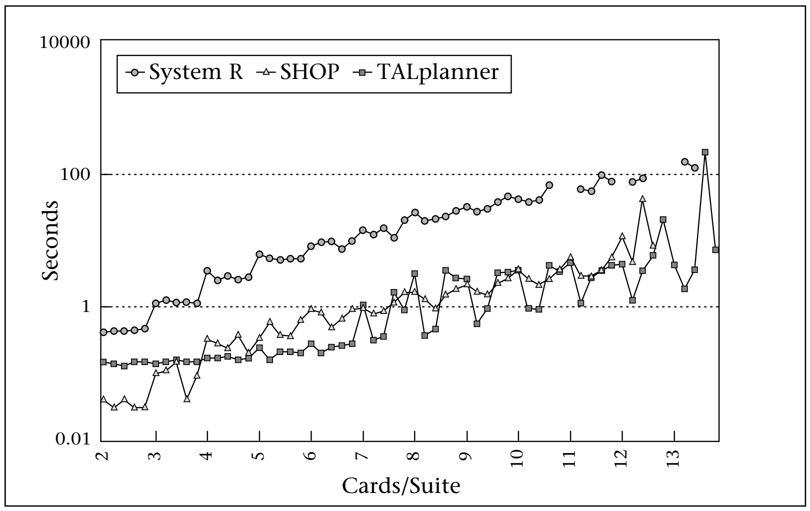

Figure 12. Hand-Tailored FreeCell.

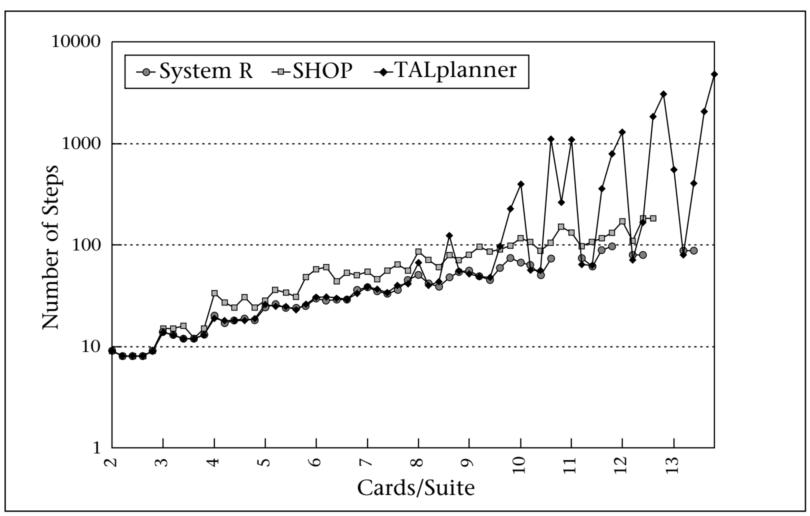

Figure 13. Hand-Tailored FreeCell Plan Length Log Scale.

For 2000, what was especially impressive was the stunning improvement in performance from 1998 and the fact that quite different techniques were being used in the best planners. Plans are proceeding for the next planning competition to be held in conjunction with AIPS'02 in April 2002 in Toulouse, France. Derek Long and Maria Fox are organizing this competition, and one of its key components will be to encompass richer models of planning, including resources and concurrency.[12]

Acknowledgments

Let me now turn to the important and pleasant task of acknowledging the assistance of the people and organizations whose help was essential in making the competition a success. Michael Ady and Janet Droppo run a software development company called Winter-City Software in Edmonton, Alberta, Canada. Ady has done most of development of the TLPLAN planning and did a tremendous amount of work in helping me set up the planning problems.[13] He also wrote a plan checker to verify the plans and used it to verify all the plans generated by the competitors. Droppo developed a set of EXCEL spreadsheets that made tracking and reporting the results feasible. Henry Kautz, David E. Smith, Derek Long, Hector Geffner, and Jana Koller volunteered to be on the organizing committee that I chaired. They provided important feedback on various decisions that had to be made about the competition's structure. Franz Inc. provided us with a copy of their ALLEGRO COMMON LISP system for the Linux system we used to run the competition—Lisp was the implementation language used by some of the competitors. Schindler Lifts Ltd. made public the elevator domain that was used in the competition and provided prizes for performance on this domain. The Cel Corporation (CelCorp) provided generous cash prizes for the top-performing competitors. John DiMarco and the systems support staff in the Department of Computer Science at the University of Toronto did an exceptional job of setting up and maintaining the system used for the competition. Adele Howe did the local arrangements for AIPS'00 and also arranged for the local computers that were used for the final phase of the competition at the conference site. John-Paul Watson and Eric Dahlman set up these computers and provided amazing support throughout the conference. Finally, last but not least, the competitors must be acknowledged for their willingness to undertake the sometimes stressful task of getting their systems ready for the competition.

Notes

1. The 1998 AI Planning Systems Competition, Drew McDermott, *AI Magazine* 21(2): 35–55.
2. For more information, see the subsequent articles, and check the competition's home page (www.cs.toronto.edu/aips2000) for additional links.
3. For more information on IPP and GRAPHPLAN, see The AIPS-98 Planning Competition, Derek Long, *AI Magazine* 21(2): 13–34.
4. Using Temporal Logics to Express Search Control Knowledge for Planning, F. Bacchus and F. Kabanza, *Artificial Intelligence* 16:123–191.
5. For example, see A Call for Knowledge-Based Planning, David E. Wilkins and Marie desJardins, *AI Magazine* 22(1): 99–115.
6. www.cs.toronto.edu/aips2000.
7. ADL operators allow for universal and conditional actions effects.
8. See www.freecell.org/ for more information about the game.
9. An unsolvable instance of FreeCell: A Game from Windows'95, Hans L. Bodlaender, www.cs.ruu.nl/~hansb/d.freecell/freecellhtml.
10. Elevator Control as a Planning Problem, Jana Koehler and Kilian Schuster, Proceedings of AIPS'00, 331–338, Menlo Park, Calif.: AAAI Press.
11. The complete specification of these domains and all the test problems can be obtained from the competition's web site at www.cs.toronto.edu/aips2000.
12. For up-to-date information on the upcoming competition, visit www.dur.ac.uk/d.p.long/competition.html.
13. Using Temporal Logics to Express Search Control Knowledge for Planning, F. Bacchus and F. Kabanza, *Artificial Intelligence* 16: 123–191.

Fahiem Bacchus is a professor of computer science at the University of Toronto. He received his Ph.D. in 1988 from the University of Alberta and spent 11 years as a faculty member at the University of Waterloo before joining the University of Toronto in 1999. He has published on topics such as uncertainty in AI, knowledge representation, utility modeling, learning of Bayesian networks, nonmonotonic reasoning, constraint-satisfaction problems, and planning. In the area of planning, his main contribution has been the tlplan approach to planning, an approach that was well represented in the 2000 planning competition. Currently, his time is split between planning, constraint satisfaction, and utility modeling. His e-mail address is fbacchus@cs.toronto.edu.