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



# The GRT Planner

*Ioannis Refanidis and Ioannis Vlahavas*

■ This article presents the GRT planner, a forward heuristic state-space planner, and comments on the results obtained from the Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS'00) planning competition. The GRT planner works in two phases. In the preprocessing phase, it estimates the distances between the facts and the goals of the problem. During the search phase, the estimates are used to guide a forward-directed search. GRT participated in the STRIPS track of the competition and showed promising results. Although it did not gain any prize, it gave us good prospects for the future.

The GRT planner is a domain-independent heuristic planner (Refanidis and Vlahavas 2001b, 1999a). It adopts the pure STRIPS representation (Fikes and Nilsson 1971) and searches forward in the space of the states. The planner was inspired by the ASP planner (Bonet, Loerincs, and Geffner 1997), but it has been differentiated in several ways.

GRT solves planning problems in two phases: (1) preprocessing and (2) search. The main idea of the planner is to compute offline, in the preprocessing phase, estimates for the distances between the facts and the goals of a problem. The word *distance* refers to the number of goal-regression levels needed to achieve a specific fact. This information is stored in a table, which is indexed by the facts of the problem. We call this table the greedy regression table (GRT).

To produce better estimates, GRT introduces the notion of related facts in the goal-regression process. These are facts that have been achieved either by the same or subsequent actions, without the last actions deleting the facts achieved first. The cost of achieving simultaneously a set of unrelated facts is considered equal to the sum of their individual costs, whereas the cost of achieving a set of related facts is considered equal to the cost of the last achieved one.

The search phase consists of a simple best-first search strategy. GRT uses the distances between the individual facts and the goals and the information about their relations to estimate the distances between the intermediate states and the goals, thus guiding the search process in a forward direction.

## An Example

We illustrate the GRT phases with the *blocks-world* problem in figure 1. Part of the GRT for this problem is shown in table 1.

Let us compute the distance between the initial state and the goals based on the information in table 1. The initial state consists of the following facts:

(*on A table*) (*clear A*) (*on B table*) (*on C B*) (*clear C*)

All these facts are related, with the fact (*on C B*) being the last achieved; so, the combined distance of these facts is the distance of the last achieved, that is, 3, which in this case is also the actual distance.

This approach is followed to estimate the distances between all the intermediate states that arise during the forward search phase and the goals. GRT always selects to expand the state with the smallest estimated distance.

## Key Points of the GRT Planner

To regress the goals, GRT does not use actions such that a single add-list fact is a goal fact, but it uses actions such that all their add-list facts are within the goals. This approach succeeds in avoiding computing estimates for invalid facts in the preprocessing phase. However, it introduces some problems in situations where the goal state is not completely described because an action to regress the goals might not exist.

To cope with this situation, at the beginning of the preprocessing phase, GRT performs an achievability analysis concerning the facts of the planning problem and computes the mutual-exclusion relations between them in a GRAPHPLAN-like manner (Blum and Furst 1997). If there are facts that are not mutually exclusive with any goal fact, there is a strong possi-

Copyright © 2001, American Association for Artificial Intelligence. All rights reserved. 0738-4602-2001 / $2.00

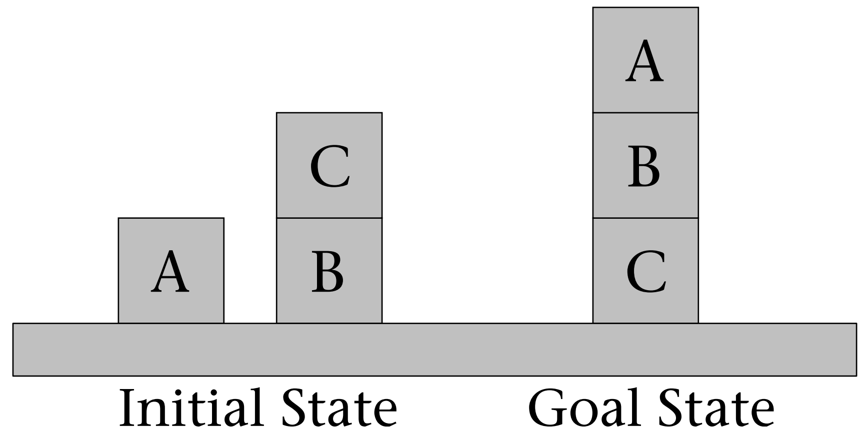



*Figure 1. A 3-Blocks Problem.*

| Fact | Distance from Goals | Related Facts |
|---|---|---|
| (*on C table*) | 0 | — |
| (*on B C*) | 0 | — |
| (*on A B*) | 0 | — |
| (*clear A*) | 0 | — |
| (*on A table*) | 1 | (*clear B*) |
| (*clear B*) | 1 | (*on A table*) |
| (*on B table*) | 2 | (*on A table*) (*clear A*) (*clear B*) (*clear C*) |
| (*clear C*) | 2 | (*on A table*) (*clear A*) (*clear B*) (*on B table*) |
| (*on C B*) | 3 | (*on A table*) (*clear A*) (*on B table*) (*clear C*) |
| ... | ... | ... |

*Table 1. Part of the Greedy Regression Table for the 3-Blocks Problem.*

bility that the goals do not form a complete state, and these facts are candidates to enhance them (Refanidis and Vlahavas 1999b). GRT supports several strategies to select which of the candidate facts to use to enhance the goals.

In some cases, there is also the need to enrich the predicate set of the domain. The situation arises in domains where negative facts are implicitly present in the initial state and the preconditions of the actions, as happens with the boarded and served facts of the elevator domain at the competition (a similar situation arose in the movie domain at the AIPS'98 competition). In such cases, GRT defines, at run time, the negative predicates (that is, not-boarder, not-served); it modifies the action schemas of the domain accordingly; and it adds new facts in the initial state. The identification of this situation is based on the detection of noninitial state facts, which are not mutually exclusive with any initial state fact, in a way similar to the goal-enhancement process.

## Comments on the Results

In the logistics domain, the GRT planner was able to solve fast all the problems, and it produced good plans. In the blocks-world domain, GRT solved only the small problems (as many as nine blocks), unable to scale up to the bigger ones. GRT faced difficulties with the specific action representation, that is, the actions *stack/unstack* and *push/pop*. We know from our experience that if *move* actions were used instead, GRT could easily solve problems with more than 20 blocks.

The schedule domain was of the ADL type, and STRIPS planners did not take part. However, experiments with GRT and an unofficial STRIPS version of this domain exhibited very good performance and good scalability.

In the *FreeCell* domain, GRT did not solve the larger problems for two reasons: First, the competition version of GRT did not instantiate actions, the arguments of which were not pairwise different, a situation that arises in this domain. Second, this version did not utilize a closed list of visited states to avoid revisiting them, a feature that would be valuable in this domain. A newer version of GRT, without these two weaknesses, is able to solve almost all the *FreeCell* problems. Note that the exploitation of a closed list of visited states improves the performance of GRT in the blocks-world domain as well, regardless of the representation that is used.

Finally, in the elevator domain, GRT solved all the problems very fast. In this domain, the use of the enriched predicate set was quite advantageous.

## Recent Extensions

In the last year, GRT has been extended in two ways. The first extension concerns the avoidance of local optimal states of the heuristic function by exploiting domain-specific knowledge in the form of state constraints. The new planner, $GRT^{SC}$, uses this knowledge to decompose a complex problem in easier subproblems that have to be solved in sequence (Refanidis and Vlahavas 2000a). The second extension, MO-GRT, concerns generating and evaluating plans on the basis of multiple criteria, such as duration, cost, safety, and planning time (Refanidis and Vlahavas 2001, 2000b).[1]

## Conclusions

The competition results have shown that the performance of the domain-independent heuristic planners is strongly affected by the representation of the domains. As for GRT, its performance varies significantly over the two alternative blocks-world representations; however, the observation concerns other similar planners too. We know also that different planners are better in different domains, so it would be interesting to investigate which features of the domains favor specific planning techniques and approaches. In any case, we believe that the heuristic state-space planners have good prospects in the area of domain-independent planning.

### Note

1. All GRT-related stuff is available at www.csd.auth.gr/~lpis/grt/main.html.

### References


Blum, A., and Furst, M. 1997. Fast Planning through Planning Graph Analysis. *Artificial Intelligence* 90(1–2): 279–298.

Bonet, B.; Loerincs, G.; and Geffner, H. 1997. A Robust and Fast Action Selection Mechanism for Planning. In Proceedings of the Fourteenth National Conference on Aritificial Intelligence, 714–719. Menlo Park, Calif.: American Association for Artificial Intelligence.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2(3–4): 189–208.

Refanidis, I., and Vlahavas, I. 1999a. GRT: A Domain-Independent Heuristic for STRIPS Worlds Based on Greedy Regression Tables. In Proceedings of the Fifth European Conference on Planning (ECP-99), 347–359. Durham, U.K.: Springer.

Refanidis, I., and Vlahavas, I. 1999b. On Determining and Completing Incomplete States in STRIPS Domains. In Proceedings of the IEEE International Conference on Information, Intelligence, and Systems, 289-296. New York: IEEE Computer Society.

Refanidis, I., and Vlahavas, I. 2000a. Exploiting State Constraints in Heuristic State-Space Planning. In *Proceedings of the Fifth International Conference on AI Planning and Scheduling Systems (AIPS'00),* 363-370. Menlo Park, Calif.: AAAI Press.

Refanidis, I., and Vlahavas, I. 2000b. Heuristic Planning with Resources. In *Proceedings of the Fourteenth European Conference on AI (ECAI-00),* 521–525. Amsterdam: IOS.

Refanidis, I., and Vlahavas, I. 2001a. A Framework for Multi-Criteria Plan Evaluation in Heuristic State-Space Planning. Paper presented at the Seventeenth International Conference on Artificial Intelligence (IJCAI-01) Workshop on Planning with Resources, 4–10 August, Seattle, Washington.

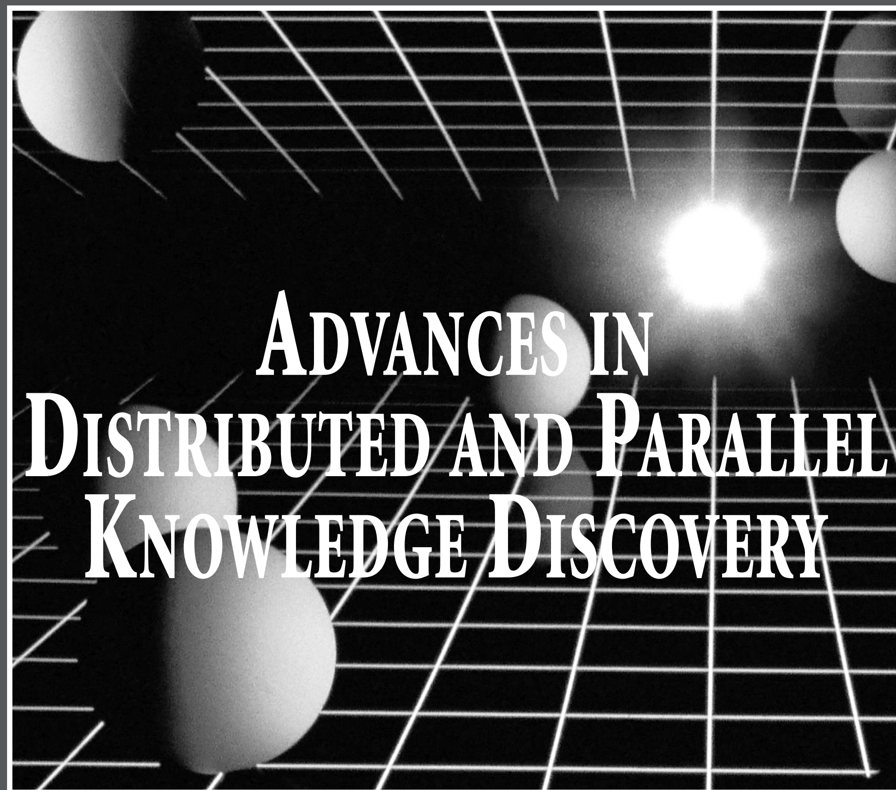
Refanidis, I., and Vlahavas, I. 2001b. The GRT Planning System: Backward Heuristic Construction in Forward State-Space Planning. *Journal of Artificial Intelligence Research.* Forthcoming.



**Ioannis Refanidis** received his B.S. in physics in 1992, his B.S. in computer science in 1997, and his Ph.D. in computer science (heuristic planning systems) in 2001 from the Aristotle University of Thessaloniki, Greece. Currently he is working as a researcher in the Department of Computer Science at the Aristotle University. His research interests include planning, resource reasoning, multicriteria evaluation, and software engineering. He is a member of AAAI and the executive council of the Greek Society on Artificial Intelligence. His e-mail address is yrefanid@csd.auth.gr.



**Ioannis Vlahavas** is an associate professor in the Department of Informatics at the Aristotle University of Thessaloniki. He received his Ph.D. in computer science (logic programming systems) from the Aristotle University in 1988. He specializes in logic programming and knowledge-based and AI systems. He teaches graduate- and postgraduate-level courses in logic programming, AI, expert systems, and decision support systems. He has been involved in more than 10 research projects, leading most of them. He is a member of the Greek Physics and Computer Societies, IEEE, and the Association for Logic Programming. His e-mail address is vlahavas@csd.auth.gr.




Edited by Hillol Kargupta and Philip Chan

Knowledge discovery and data mining (KDD) deals with the problem of extracting interesting associations, classifiers, clusters, and other patterns from data. The emergence of network-based distributed computing environments has introduced an important new dimension to this problem—distributed sources of data. Distributed knowledge discovery (DKD) works with the merger of communication and computation by analyzing data in a distributed fashion. This technology is particularly useful for large heterogeneous distributed environments such as the Internet, intranets, mobile computing environments, and sensor networks. When the datasets are large, scaling up the speed of the KDD process is crucial. Parallel knowledge discovery (PKD) techniques addresses this problem by using high-performance multi-processor machines. This book presents introductions to DKD and PKD, extensive reviews of the field, and state-of-the-art techniques.

**PUBLISHED BY THE AAAI PRESS / COPUBLISHED BY THE MIT PRESS**
**Five Cambridge Center, Cambridge, Massachusetts 02142 USA**
**http://mitpress.edu/ • 617-625-8569 • 800-356-0343**

**ISBN 0-262-61155-4, 472 pp., illus., bibliography, index**