Finding Dominant Plans Using Plan Evaluation Criteria

Usbeck, K., Regli, W. C., Wickler, G., & Tate, A. (2009). Finding Dominant Plans Using Plan Evaluation Criteria. In Proceedings of the Fifth International Conference on Knowledge Systems for Coalition Operations (KSCO-2009).

Finding Dominant Plans Using Plan Evaluation Criteria

Kyle Usbeck and William C. Regli
{kfu22,regli}@drexel.edu
ACIN, Department of Computer Science
Drexel University
Philadelphia, PA, *USA*

Gerhard Wickler and Austin Tate
{g.wickler,a.tate}@ed.ac.uk
AIAI, School of Informatics
The University of Edinburgh
Edinburgh EH8 9LE, *Scotland*

Abstract — Qualitatively different plan generation is one aspect of planners that is useful to execution agents that operate under uncertainty, such as those in coalition operations. The purpose of this research is to improve plan construction and selection. To do so, we define the statistics that make plan evaluations most useful to planning and task assignment agents. Furthermore, we develop a method of optimizing and visualizing plans based on evaluation criteria. Experiments show that guiding the planner's search strategy based on plan evaluation criteria improves the diversity of qualitatively different plans.

1 Introduction

Often, plan execution agents lack the logic to operate robustly in highly-dynamic environments, such as coalition operations. Execution agents that perform under uncertainty can utilize multiple plans to improve their robustness. In such cases, the diversity of the plans affects the speed and efficiency of plan execution.

The concept of generating qualitatively different plans has been explored by many [4, 5, 9, 2]. Some, such as [3], even explore the possibility of using domain-dependent plan evaluations as a basis of plan differentiation. The purpose of this research is to improve the ability of agents to construct and choose between plans. To do so, we develop a new method of optimizing and visualizing plans based on evaluation criteria.

By specifying plan evaluation criteria, a task assignment agent is capable of finding plans that are most relevant to the current situation. We claim that plan evaluations can be used as an effective mechanism for finding dominant plans and visualizing plan results.

1.1 Approach

This paper presents a technique for improving mixed-initiative planning using domain-dependent and domain-independent evaluation criteria. We hypothesize that our method will improve plan execution in dynamic, multi-agent coalition operation environments.

To prove our hypothesis, we have developed an example scenario using an Improvised Explosive Device (IED) detection problem described in Section 3. Our approach to using plan evaluation criteria for guiding a planner is as follows:

1. Model the IED detection scenario as an HTN planning domain.
2. Modify the planner's searching algorithm to find qualitatively different plans based on (multiple) network-related evaluation criteria.
3. Compare the efficiency of the searching al-

gorithm to the improvement in plan cost.

The technical approach is described in Section 4.1. Prior to our approach, we note some relevant background material.

2 Background

2.1 Classical Planning

Classical Planning models the planning domain, Σ, as a state transition system such that $\Sigma = (S, A, E, \gamma)$ where S is the set of states, A is the set of actions, E is the set of events, and $\gamma = S(A \cup E) \rightarrow 2^S$. The planning problem, $\mathcal{P}$, can in turn be expressed as the triple (Σ, s_0, S_g) where s_0 is the initial state and S_g is a set of goal states.

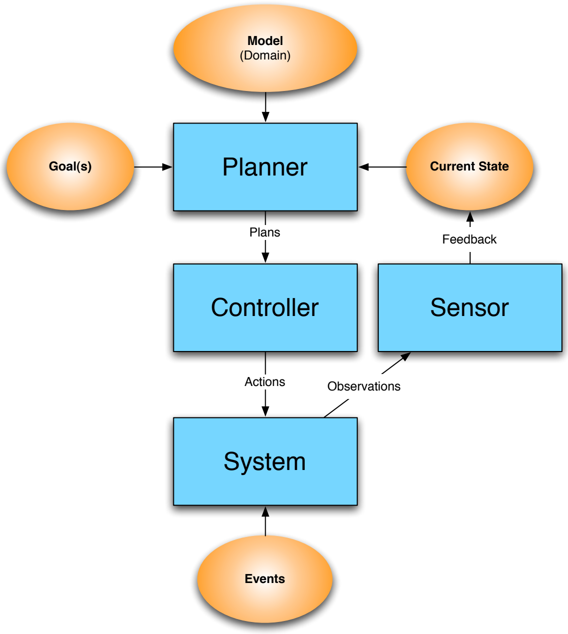

Figure 1: This figure illustrates the interactions between agents in a planning architecture.

2.2 HTN Planning

As described in [6], HTN planning differs from classical planning in that classical planning's objective is to achieve a set of goals whereas HTN planning's objective is to perform a set of tasks. Furthermore, tasks can be compositions of subtasks, subtasks can be decomposed into smaller subtasks, and so on until primitive tasks are reached and the planning operators can be performed directly.

2.3 Role of Agents in Planning

In [8], Tate discusses the roles of agents in the planning process. He defines three key agent roles: *Task Assignment*, *Planning*, and *Execution*. In this notion, the planning agent is responsible for solving a static planning problem described in Section 2.2 and passing the plan to the execution agent. The execution agent interacts with the real system, and in some situations, can react to some action execution failures. The task assignment agent communicates with the planning and execution agents to trigger plan creation and execution respectively.

The agent interactions in [8] form a Control Theory feedback loop. A control loop contains a *controller* which initially accepts a plan and then gives input to some *system*. A *sensor* component determines the current state of the system to help guide the controller in the remaining execution steps. In this terminology the controller functions as the task assignment and execution agent — it should be noted that these operations can be separated. Figure 1 combines Tate's agent roles with the control loop of agents in planning systems.

2.4 Comparing Plans

There are two high-level techniques of finding qualitatively different plans: domain-independent, and domain-dependent.

2.4.1 Domain-independent

One advantage to using domain-independent methods of finding qualitatively different plans is that they require no information, other than the domain, to derive different plans. The most popular method of finding qualitatively different plans without additional information gathers high-level preferences from the user. This approach is labeled, *mixed-initiative.*

Although TRIPS and TRAINS are considered mixed-initiative planning assistants, the authors explicitly say, "traditional planning technology does not play a major role in the system" [2]. The systems are mixed-initiative in that they help to repair initial tasks, however their approach aims to perform plan repair on an existing plan rather than generate unique plans.

In [7], Srivastava *et al.* investigate methods to find inter-related plans. They use a function, $distance(plan1, plan2)$, to represent the similarity/diversity of two plans. The function could use any combination of the following three domain-independent criteria:

- the actions present in the plan,
- the set of stages (or states) that execution takes, and
- the causal chains that support plan goals.

Tate *et al.*'s mixed-initiative approach, described in [9], is largely driven by the task assigner agent selecting the assumptions on the top-level activities. The planner is then responsible for refining the lower-level plan activities.

2.4.2 Domain-dependent

The main advantage of using domain-dependent methods of finding qualitatively different plans is that they incorporate domain information into the inter-related plan measurements. This is accomplished by adding new (domain-dependent) criteria to definition of the *distance* function. In [4] and [5], Myers *et al.* use the concept of domain metatheory to evaluate plans. In addition to providing a mechanism for comparing sets of plans, metatheory also provides capabilities to summarize plans. The conceptual components of metatheory are as follows:

- template features (which allow us to differentiate between functionally equivalent alternatives),
- task features (which is a typing system), and
- roles (which describes the capacity to which an individual resource is used).

Myers uses these components to direct planners towards solutions with distinct semantic traits in [4].

2.5 Dominant Plans

A goal of the project is to help agents choose between multiple plan options. One way we accomplish this is by distinguishing dominant plans from those that are dominated. To define the notion of *dominance*, we look to the field of game theory. Game theory's Iterated Elimination of Strongly Dominated Strategies (IESDS) technique for solving games states that *A strictly dominates B when choosing A always gives a better outcome than choosing B.* Applying this strategy to solving games often helps to reduce the size (and complexity) of the game. Our notion of dominance is analogous to that of IESDS in a cooperative game where each agent is a player whose interests are represented by a different plan evaluator.

3 Motivating Scenario

IED Change Detection is being developed by the US Army Commu-

> nications Electronics Research, Development and Engineering Center (CERDEC), to detect IEDs along travel routes using high resolution aerial/overhead imagery... This system helps an operator to identify and locate new environmental changes on a route which could indicate the presence of IEDs or landmines. [1]

The scenario we examine for this project provides a way to balance the tradeoffs between manual (human) IED detection and performing environmental change detection. There are advantages and disadvantages to each method of IED detection. For example, manual human searching is more accurate (likely to discover any IEDs present) than change detection, but it requires more time than change detection and requires a human resource that could otherwise be performing another task.

Furthermore, the tasks are constrained by the resources that are currently available. For example, manual human searching cannot be accomplished if the only human is already performing some other task. Additionally, properties of the resources can effect the evaluations of the plans. A picture taken with a high-resolution thermal camera mounted on an Unmanned Aerial Vehicle (UAV), for instance, will increase the speed of the plan execution and accuracy of the IED change detection process, but requires a larger amount of data to be transported over the network.

In this scenario, there are several locations that must be monitored for IEDs. Monitoring can be accomplished using any of the methods and resources discussed above. Each combination yields different plan evaluations, and the goal is to exploit the tradeoffs between the evaluation criteria to find qualitatively different plans to return to the user.

[1] `http://www.defense-update.com/features/du-4-04/IED-Early-Warning.htm`

The following are planning actions for the IED detection domain:

- sweepForIEDs: defines the list of locations to be searched.
- checkForIEDAt: satisfied by manualSearch or photographicSearch.
- manualSearch: complex task for a human search of a location.
- photographicSearch: complex task for a change detection search of a location.
- conductScan: a human scans a location for an IED.
- acquireCamera: a resource acquires a camera (a requirement for a change detection task).
- takePhoto: take a photo of a location using a camera.
- getOldPhoto: get the last photo taken of a location (a requirement for comparing photos).
- comparePhotos: compares two or more photos for a change that would indicate an IED is present at a location.
- physicalMove: move a resource from one location to another.
- reportResults: the results of a scan are reported to a central authority.

Also, we have high-level resources with type net-nodes, locations, and cameras. Net-nodes here represent anything that moves in physical space or communicates over the network. Thus, both humans and UAVs are represented by net-nodes. To distinguish between these types of net-nodes, we use net-node properties. For example, a UAV net-node has an average speed around 230mph versus a human net-node which is about 3mph.

4 Approach

Each IED detection method has costs and accuracy-levels. For example, costs would include the amount of gasoline or bandwidth consumed by a method. Another cost could be the amount of time necessary to execute a action. Accuracy is slightly different in that we seek to maximize the accuracy of the joint methods. For example, manual search has a higher probability of discovering an IED than change detection.

In addition to the methods that make-up a plan, several other factors can influence the costs and accuracy of a plan. Available resources yield variable costs and accuracy (e.g. higher-resolution camera is on a remote UAV and requires more network bandwidth, but has greater IED detection accuracy). Also the ordering of plan actions can effect the overall evaluation. For example, taking an aerial photograph of a location that has just recently been manually search does not increase the accuracy of the IED detection.

4.1 Biasing the Planner: Qualitatively Different Plans

This section discusses the techniques used throughout the rest of the paper to bias the planner using plan evaluation criteria.

Section 2.4 discusses two high-level techniques for finding qualitatively different plans: domain-independent, and domain-dependent. We use plan evaluation criteria to represent both techniques. The reasoning for using a single mechanism for both techniques is that plan evaluators are sufficiently capable of recognizing domain-independent as well as domain-dependent information about a plan. In fact, domain metatheoretic roles are implicitly rooted in plan evaluation criteria.

The idea for biasing the planner is to identify qualitatively different plans using the plan evaluation criteria produced by a plan evaluator. A plan evaluator contains the following:

- a complete/fully-ground plan evaluation function, EVALPLAN(P);
- a partial plan evaluation function, HANDLEPARTIALPLAN(p); and
- evaluation criterion statistics, discussed in Section 4.2.

Our method for biasing the planner's search strategy based on plan evaluations is to maintain a set of priority queues $\{Q_1, Q_2, \ldots, Q_n\}$ — one for each plan evaluator, where n is the number of plan evaluators. Every time a new partial-plan/backtrack-point is generated, its viability is assessed and it is inserted into each priority queue according to the partial plan evaluation of the priority queue's plan evaluator. Psuedocode for the algorithm is shown in Algorithm 1.

Algorithm 1 HANDLEPARTIALPLAN(p)

Require: p is the partial plan accepted as input. $\mathcal{E}$ is the set of plan evaluators. $\mathcal{L}$ is a list of priority queues containing plan evaluations.

Ensure: EVALPARTIALPLAN($evaluator, p$) is a function that evaluates partial plan, p, using the partial plan evaluator, *evaluator*. INSERT(Q, e, p) is a function that inserts a partial plan, p, into the priority queue, Q, according to the evaluation, e.

for all $evaluator \in \mathcal{E}$ **do**
  $Q \leftarrow \mathcal{L}[evaluator]$
  $e \leftarrow$ EVALPARTIALPLAN($evaluator, p$)
  INSERT(Q, e, p)
end for

4.2 Plan Evaluation Criteria Statistics

Alone, plan evaluators can distinguish only relative distances between plans. By adding a concept of plan evaluation criteria statistics to plan evaluators, we can position plans along an absolute continuum of evaluation values. The aspects of plan evaluation criteria statistics are

- range (effective and theoretic),

- direction (minimize or maximize), and
- statistics (e.g. mean, median, mode, standard deviation).

Plan evaluation statistics specify a theoretic range and keep track of the effective range of plan evaluation values. By specifying and tracking the evaluation statistics in this manner, we can implement metatheory within our plan evaluators using the theoretic values, or we can dynamically create our metatheoretic categories based on the effective statistics of the plan evaluations we have performed up to any point in the search.

As discussed in Section 4.2.1, domain metatheory features are rooted in discrete evaluation values. It is implied that the final task assigning agent understand what certain qualities of evaluation criteria are desired. For example, the task assigner must know what level of affordability the end-user seeks.

The notion of plan evaluators requires that the user specify one evaluator for each concern in the plan. An analogous plan evaluator for the "affordability" feature would be a "monetary cost" plan evaluator which seeks to minimize the overall cost. By specifying that we aim to minimize the monetary cost of a plan, we are able to eliminate strictly dominated plans, discussed in Section 2.5 and 4.3.

Our approach differs from previous approaches to generating qualitatively different plans in that it improves mixed-initiative planning by using a combination of domain-dependent and domain-independent plan evaluations.

4.2.1 Comparison to Related Work

Some preference-based planners, such as [1], filter plans in a post-processing phase, whereas our method biases the planner's search algorithm to find qualitatively different plans. Other work in this area, such as [9], uses mixed-initiative approaches to gather plan preferences.

Our method of finding qualitatively different plans most closely resembles Myers's concept of domain metatheory [4]. Domain metatheory, however, is rooted in discrete evaluation ranges. In Myers's example, transportation methods are distinguished by affordability and time-efficiency (among others). The feature, affordability takes the values "extravagant, expensive, moderate, inexpensive, or cheap" defined as categories of features. Time-efficiency is similarly broken into discrete categories.

By eliminating the predefined categorizations of the evaluations, we allow for a higher level of granularity in evaluating partial plans. Using partial plan evaluators as a basis for guiding the planner's search strategy has a greater expressivity than domain metatheory. Domain metatheoretic values can be expressed as partial plan evaluators, but complex interactions of methods and resources are more accurately modeled as partial plan evaluators. For example, filling a ground vehicle with expensive UAV fuel results in wasted money. Where domain metatheory might explore this option because it offers significantly different features in the plan, plan evaluation will guide the planner away from this result.

4.3 Plan Evaluation Visualization

While a task assigning agent might be interested in any number of plan evaluators, the plans whose evaluations dominate other plans in every criteria should certainly be considered. The plan evaluation visualization user interface (viewed by the task assigner) makes a distinction between dominant plans and their dominated counterparts. Dominant plans are defined in Algorithm 2.

The purpose of finding dominant plans is to present them to the execution agent as the most likely candidates for plan execution. A

Algorithm 2 TESTFORDOMINANCE(p, Γ)

Require: Γ is the set of plan evaluation criteria on which to test plan, p, for dominance (modified to seek minimization if necessary). Ψ is the set of all plans (other than the plan we are testing, p).
Ensure: Applying a plan to a plan evaluator yields a quantitative evaluation.

```
for all plan ∈ Ψ do
    for all e ∈ Γ do
        return (e(plan) ≥ e(p))
    end for
end for
```

visualization has been designed to plot plans along the continuum of possible evaluation values. Furthermore, the plan evaluation criteria statistics are maintained in the visualization to show general performance of the planner and indicate relative evaluation values for each criterion.

5 Experiment

The goal of the experiment is to show that plan evaluation criteria helps the task assignment agent in finding the most pertinent plans. To do this, we show that identifying dominant plans can improve the task assignment agent's coarse of action options.

Plan evaluators for each evaluation criteria described in Section 4 were implemented for the experiments discussed in this section. The domain-independent plan evaluators include:

- **Issue Count Evaluator** — minimizes the remaining implied constraints.
- **Node Count Evaluator** — minimizes the number of plan activities.
- **Longest Path Length Evaluator** — minimizes the length of the path along temporal ordering constraints.
- **Object Use Evaluator** — minimizes the number of resources used by the plan.
- **Object Count Evaluator** — minimizes the number of resources added to the world state (to facilitate variable grounding).

The IED-detection domain-dependent evaluators include:

- **Bandwidth Evaluator** — minimizes the total bandwidth consumed by a plan.
- **Hop Count Evaluator** — minimizes the number of network hops over which data travels through the network.
- **Monetary Cost Evaluator** — minimizes the total monetary cost incurred by a plan.
- **Plan Execution Time Evaluator** — minimizes the minimum amount of time necessary to execute a plan.
- **IED Detection Accuracy Evaluator** — maximizes accuracy of the IED detection techniques in a plan.

All experiments presented in this paper were implemented in Java using I-X/I-Plan[2] (version 4.5 build 10-Mar-08) with the IED detection scenario described in Section 3. They were all compiled and run on the same 2 GHz Intel Core Duo MacBook Pro with 2GB of RAM running MacOS version 10.5.5 with Apple's JVM (build 1.5.0_16-b06-284).

5.1 Search Strategies

The dominant plan experiment compares the optimization level of dominant plans for each search strategy. This section discusses the search strategies used in the experiments.

[2] I-X is a framework developed by the Artificial Intelligence Application Institute at the University of Edinburgh that allows humans and computer systems to cooperate in the creation or modification of some product such as a design, physical entity or plan. Within the I-X framework is an architecture, I-Plan, in which situated agents, such as planning agents, can be created. More information can be found at `http://www.aiai.ed.ac.uk/project/ix/`

The guided search algorithm is described in Section 4.1 and it is fairly intuitive that a random search strategy randomly selects from a list of backtrack points in the search space.

I-Plan's default search strategy uses a combination of exploration and optimization to return different plans. As the planner traverses the search space, it switches between a depth-first exploration strategy and an A* optimization strategy using the number of activities in the partial plan as its admissible heuristic. The planner starts by traversing the space in a depth-first manner and when it encounters an alternative whose constraints cannot be satisfied, it backtracks using the A* search. During the process, the planner monitors statistics about its search. These statistics include the following:

- the number of steps the planner took to generate a plan,
- the number of alternatives the planner uncovered along its way,
- the number of search options below the revealed alternatives,
- the number of alternatives left unexplored,
- the longest path along temporal ordering constraints, and
- the number of duplicate plans found before returning a new plan.

6 Results

This section of the paper discusses the results of the experiment described in Section 5.

6.1 Plan Evaluation Visualization

The plan evaluation user interface offers visualization of current evaluation values and the statistics of each plan evaluator. The goal of this visualization is to help the task assigning agent to quickly and efficiently understand the plan options explored by the automated planner in respect to their evaluations. Figure 2 shows a screen capture of the modified I-Plan Option Tool displaying plan evaluation visualization information.

6.2 Dominant Plans

See Figure 3 for a chart of the dominant plans for each search strategy between two evaluation criteria. In this case, we chose plan execution time and IED detection accuracy. Keep in mind that we seek to maximize IED detection accuracy and minimize plan execution time. This chart shows that, while I-Plan was able to find some plans with better plan evaluations in one criteria, the guided search uncovered a completely new search area that neither I-Plan's default search nor random search uncovered (the bottom-right-most point in Figure 3). This data is supported by the fact that I-Plan strives to do some local optimization in its search strategy, which in turn causes it to perform broad tree exploration less-often.

7 Conclusions

This paper explains the usage of plan evaluation criteria to improve the ability of agents to construct and choose between plans. In doing so, we describe the contents of plan evaluation statistics and how the evaluations can be used to find dominant plans.

By specifying plan evaluation criteria statistics for each plan evaluator, we exploit the notion of dominant plans, described in Section 2.5. A limited set of dominant plans are presented to the task assigning agent for careful consideration since these are most likely to be the best plan options.

I-Plan uses an *option comparison matrix* (see the top-right panel of Figure 2) to inform the

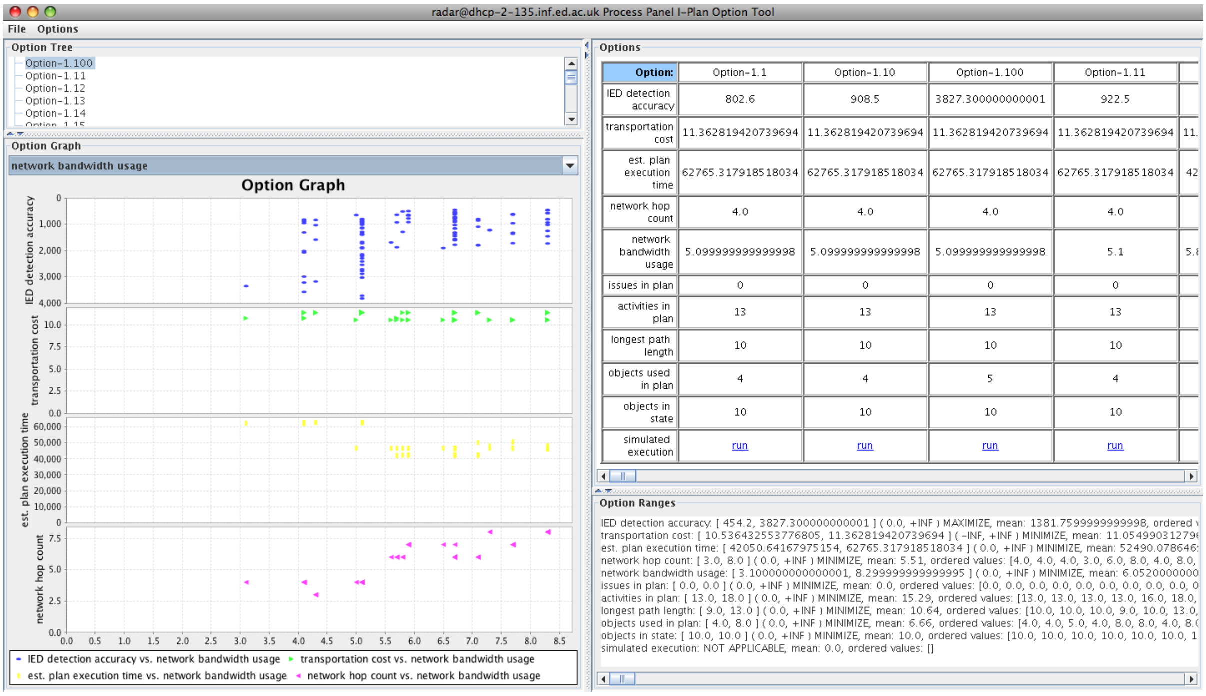

Figure 2: A screen capture of the modified I-Plan Option Tool displaying plan evaluation comparisons and statistics. The top-right panel is the *option comparison matrix* which displays textual information about each of the plan options. The bottom-right panel is the *option statistics visualization* which shows the theoretic ranges and direction of the plan evaluators as well as current statistics for each plan evaluator. The bottom-left panel is the *option comparison graph* which shows the effective ranges of the plan evaluators and plots the dominant and dominated plans along these ranges.

user of plan evaluation information once a solution is found. The matrix has been augmented with plan evaluation statistics and comparative evaluations to plan evaluation criteria.

Experiments show that guiding the planner's search strategy based on plan evaluation criteria improves the diversity of qualitatively different plans.

7.1 Continued Work

The proper guidance of the search strategy relies on the correlation of the partial and full plan evaluators. Future experiments will explore the correlation between these parts of the plan evaluators.

Other future work will explore plan execution, monitoring, and repairing using plan evaluation criteria.

References cited

[1] Mark S. Boddy, Johnathan Gohde, Thomas Haigh, and Steven A. Harp. Course of action generation for cyber security using classical planning. In *ICAPS*, pages 12–21, 2005.

[2] George Ferguson, James F. Allen, and

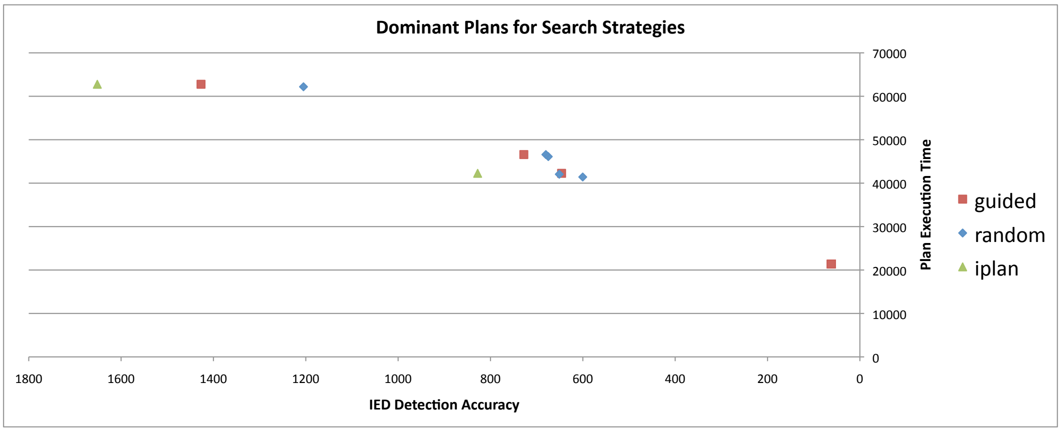

Figure 3: This chart shows the comparison of dominant plans (with respect to plan execution time and IED detection accuracy plan evaluators) discovered within the first 40 plans of each search strategy.

Brad Miller. Trains-95: Towards a mixed-initiative planning assistant. In *Proceedings of the National Conference on Artificial Intelligence*, pages 70–77. AAAI Press, 1996.

[3] Yolanda Gil, Mark Hoffman, and Austin Tate. Domain-specific criteria to direct and evaluate planning systems. In *Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop*. Morgan Kaufmann, 1994.

[4] K. Myers and T. Lee. Generating qualitatively different plans through metatheoretic biases. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. AAAI Press, 1999.

[5] K. L. Myers. Metatheoretic plan summarization and comparison. In *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS-06)*. AAAI Press, June 2006.

[6] Dana Nau, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[7] B. Srivastava, S. Kambhampati, M. H. Do, and T. Nguyen. Finding inter-related plans. In *ICAPS 2006 Workshop on Plan Analysis and Management*, 2006.

[8] Austin Tate. Authority management - Coordination between task assignment, planning and execution. In *IJCAI-93 workshop on Knowledge-based Production Planning, Scheduling and Control*, 1993.

[9] Austin Tate, Jeff Dalton, and John Levine. Generation of multiple qualitatively different plan options. In *Fourth International Conference on AI Planning Systems (AIPS-98)*, pages 27–35, 1998.