IDA Assistance for Mixed-Initiative Planning

Robert St. Amant

Department of Computer Science
North Carolina State University
EGRC-CSC Box 7534
Raleigh, NC 27695-7534
stamant@csc.ncsu.edu
http://www.csc.ncsu.edu/eos/users/s/stamant/www/index.html

Abstract. In mixed-initiative planning, a human planner collaborates with an automated system to build a plan of action. In existing approaches, the participants incrementally add constraints or modifications to a single plan structure until an acceptable solution is found. We propose instead to treat the process as a form of data exploration, in which sets of plans and their relationships are the objects of interest, rather than an individual abstract plan. The system assists the human planner in this process by autonomously constructing simple statistical descriptions of patterns in sets of plans. Its analysis is supported by heuristic knowledge about the physics of a planning environment, knowledge that contributes to a limited form of strategic statistical reasoning.

1 Introduction

In mixed-initiative planning, a human planner and an automated system contribute to a problem solution—formulation, development, analysis, repair—without the need for a constant exchange of explicit instructions [5]. The goal is to integrate human judgment into an automated search for solutions to complex problems. The difficulty in mixed-initiative planning lies in formalizing and managing an interactive, cognitively intensive process in which control is shared between the participants. Researchers have attacked the problem from different perspectives, treating the human-computer interaction as a peer-to-peer dialog [8, 9], as a type of visual programming [4], as an advice-exchanging relationship [17], and as a navigation problem [22, 24], among other possibilities.

These approaches to mixed-initiative planning all share a common theme. The user and the system enter into an exchange (in various systems via command entry, direct manipulation, even spoken input) about an initially abstract, relatively unstructured plan. As each participant contributes modifications, constraints, and other forms of guidance to the construction process, the plan is gradually revised and refined until it meets its goals. This style of interactive search has led to systems that contribute significantly to problem-solving while still interacting with users in a relatively natural way [8, 19, 24].

One limitation of this approach, however, is that while the system may search extensively for plans, the user generally views only one plan at a time, considering questions such as "Which actions should the plan contain?" and "How should their execution be constrained?" The planning process moves forward incrementally through local changes to an abstract plan, at a cost very familiar to data analysts [7]. We propose to change the terms of the mixed-initiative search to ameliorate this problem. We treat the planning process as a form of data exploration, in which sets of plans and their relationships are the objects of interest, rather than a single abstract plan. We have built a prototype data analysis planning system, called DAPS, to support the user in three areas of plan space exploration: a representation that focuses on sets of plans rather than individuals; data analysis methods appropriate for identifying patterns in this space; and heuristics to guide the selection and application of these methods. These methods and heuristics are based on a representation of knowledge about interaction with the physical world. This knowledge, applied to a dataset describing the behavior of agents in a planning environment, helps guide the decisions involved in the analysis of the dataset. In this regard our work has implications for statistical strategy as well as planning.

2 Background

This paper integrates work in three areas: mixed-initiative planning, intelligent data analysis, and reasoning with physical schemas.

Mixed-initiative planning is one approach to building an intelligent decision support system. A mixed-initiative system shares decision-making responsibility with the user such that it acts sometimes as a tool, to be directly applied to a specific task, and other times as an autonomous problem-solver. In the best case, the user can delegate the details of a task to the automated system without giving up the ability to guide and review the decision-making process. James Allen [1] distinguishes mixed-initiative planning from conventional planning by three characteristics: mixed-initiative planners allow problem-solving initiative to change hands flexibly and opportunistically between the user and the system; they are able to shift focus of attention to meet changes in user needs; they contain mechanisms for maintaining shared, implicit knowledge. Burstein and McDermott expand on these and related issues in a summary of the state of the art [5]. From an IDA perspective, this area gives us problems with large amounts of data, rich structure to be extracted, and complex decisions that depend on our interpretation of the data.

Our work builds on two earlier statistical systems, CLASP and AIDE. CLASP, the Common Lisp analytical statistics package, is an interactive environment for data analysis and exploration [2]. CLASP gives interactive and programmatic access to a variety of statistical tools: graphical displays, data manipulation operations, model-fitting algorithms, and statistical tests. AIDE, a descendant of CLASP, is an assistant for intelligent data exploration [23, 24]. AIDE focuses on techniques for representing and applying *statistical strategies*, formal descriptions of the actions and decisions involved in applying statistical tools to a problem [13, 11]. One of the most difficult problems faced by AIDE (as well as earlier systems such as REX [10], TESS [16], and others [11, 13, 14, 18, 21]) is that of context. Understanding the meaning of data can strongly influence our decisions about appropriate analysis [12]. Handling contextual knowledge in general is difficult for an automated system, because it usually requires knowledge about the real world and some degree of common sense. AIDE addresses the problem through various techniques for "accommodating" user expertise into its search for descriptions and models. In our current work, CLASP acts as a statistical and numerical processing substrate, while AIDE provides the knowledge structures for higher-level manipulation and evaluation.

The third area we draw on is reasoning about physical relationships. Some researchers in human cognition see a strong relationship between our experiences in interacting with the physical world and our understanding of physical and even entirely abstract concepts [15]. This correspondence applies to data analysis as well. Data analysis methods are closely tied to our understanding of the physical world. If we explain the notion of a measure of central tendency to beginning students of statistics, for example, we appeal to their intuitions about objects in a one or two-dimensional space, often bringing mass into the picture. The notion of variance, or spread, easily follows. In fact, for a great many statistical concepts, from counting to modeling techniques for regression, clustering, and so forth, we find a natural mapping to real-world physical concepts. While other interpretations (e.g., probabilistic accounts) are common as well, physical explanations have a natural intuitive appeal.

To summarize, our work seeks to integrate these areas, to show that knowledge about physical relationships, behaviors, and tasks can give a statistical system the ability to reason more effectively—more strategically—about patterns relevant to interactive plan construction.

3 Problem environment

Our work takes place in the context of AFS, an abstract force simulator provided by Paul Cohen's lab at the University of Massachusetts [3]. AFS is a general-purpose simulation system that supports experimentation with interactive, distributed planning techniques and their relationship to physical processes. AFS provides a physical domain in which agents can interact, based generally on Newtonian physics. Agents and objects have mass, size, and shape; they may be solid or permeable; they move with variable friction over terrain; they apply force to one another, causing damage/mass reduction.

In AFS's Capture the Flag domain, two teams of agents move over a terrain, their travel constrained by mountains, water, and forests. Each team is responsible for defending a set of stationary flags, and successfully completes a scenario by destroying the members of the opposing team or capturing all of its flags. A sample scenario is given in Figure 1. In this domain, as in all AFS domains, agents rely on a small set of primitive physical actions: they may `move` from one

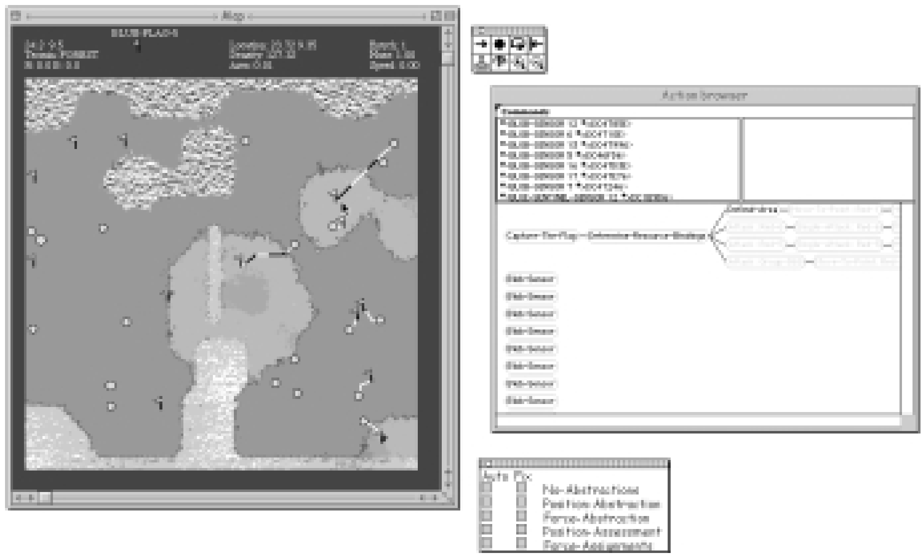

Fig. 1. A Capture the Flag scenario in AFS

location to another and `apply-force` to other agents and objects such as flags. These actions can be specialized and combined in various ways to form higher level strategies for action, such as blocking a pass, encircling a flag, attacking an opponent in a group, and so forth. Plan execution and monitoring is provided by HAC, a hierarchical planner embedded in AFS.

We have taken steps toward conventional mixed-initiative planning in AFS. The user can direct the low-level actions of agents, and can view visualizations of decisions the planner makes, such as the tasks a team has taken on and how its members are assigned. Figure 1 shows a visualization of a partial plan, in which some agents are assigned to defend their flags. The figure also shows a plan browser that displays a more abstract view of the planning process. As discussed earlier, the focus of the planning process is on modifications and refinements to a single active plan. One reason the task is difficult for the user is the need to work either with the details of directing individual agents, or with the much more abstract representation of the plan as a directed graph.

The new approach we will describe combines some of the advantages of both views of planning. The key is to consider the execution of plans embedded in the physical world. We simulate a plan forward in time, recording time-stamped values for the location and activity of each agent. A plan then becomes the "paths" agents take through a spatial, temporal, and action space, and the information derived from their passage. At this level of description, distinct plans can be aggregated, a much more difficult proposition with the action- or graph-based representations above. We do lose some flexibility, in that plans must be elab-

Evaluation level:	Exploration, evaluation, presentation
Task level:	Context-dependent interpretation
Physical level:	Physically meaninful derivations
Statistical level:	Conventional statistical results

Fig. 2. Data analysis levels in DAPS

orated to the point that they can be simulated, but in return we gain a great deal of concrete detail.

We also gain an element of abstraction that is sometimes lacking in mixed-initiative planning. Instead of the interaction concerning actions that should be added to a plan, their ordering, their timing, and so forth, it now treats questions such as "What properties should candidate plans have, regardless of their internal structure?" and "How do these plans compare with those plans?" and "Do other plans exist that differ from the current plan but have these properties...?" Because the interaction deals with sets of plans, rather than individual plans, users gain a different perspective on the process.

The next section describes how we explore the space of plans as sets of paths. The source data for the exploration is provided by a restricted version of the planner HAC, rather than the complete planner, which allows for flexibility in the current early stage of our development. HAC generates a plan in which agents are assigned to various roles: attacking opponent flags, attacking opponent agents, defending flags, defending regions, blocking strategic passes, or aggregating with other agents. A plan involves a temporal component as well as a spatial component: some actions, such as attacks on distant targets, may be synchronized. An initial attack plan consists of each agent on a team simply traveling to its destination and there executing its assigned task. Our IDA task is much simplified: we consider only initial attack plans, rather than complete plans, we ignore the uncertainty from interaction with opponent agents, and we leave aside the iterative human interaction element, to concentrate on the IDA part of the picture. As should be clear in our discussion below, however, the simplified problem retains much of the complexity of the full planning problem.

4 Data analysis for planning

Data analysis in DAPS can be divided into four distinct levels by the kind of data processed and the results generated, as shown in Figure 2. Acting at the lowest level we have the statistical operators provided by CLASP. These operate on a simple representation of paths as numerical and symbolic sequences. Also at this level we have abstract operators provided by AIDE, which take functions and relationships as input: reductions, transformations, and decompositions [23]. A reduction summarizes a relationship by mapping it to a scalar value. Computing the mean of a sequence of numbers is an example of a reduction of a relationship; DAPS computes the aggregate mass of a team via a reduction. A transformation maps a function over the tuples of a relationship. A familiar transformation is a log transform; another is the generation of residuals from a linear fit. DAPS uses transformations to filter the spatial path an agent follows. For example, "penetration into opponent territory," measured simply as forward distance, is one useful transformation of a sequence of Cartesian location coordinates. Finally, a decomposition operation breaks a relationship down into smaller relationships. Separating a relationship into clusters is a decomposition, as is isolating outliers, both of which DAPS relies on to impose structure on sets of plans. This lowest level provides general-purpose manipulation of the planning data via what we will term *statistical* operators.

One level above, we have heuristic operators that limit the lowest-level functions to those reductions, transformations, and decompositions that generate data with a useful physical interpretation. We call these *physical* operators. For example, one simple operator aggregates all the agents on a team and computes their total mass. Another operator specifies a combination of transformations and reductions that creates a sequence of planned locations (comparable to one of those in Figure 1) for the aggregation, based on the individual paths of the member agents. These operators generate information about the physical behavior of the agents in the plan. These are some examples of their output:

Agent/object properties:
- Agent/object aggregation: members and membership.
- Mass (which represents strength.)
- Mass distribution, plus temporal dependence.
- Location/center of mass, plus temporal dependence.
- Spatial distribution, plus temporal dependence.
- Incremental movement, plus temporal dependence.

Plan properties:
- Duration.
- Event sequences.
- Event ordering precedence.
- Event coordination.

This information feeds the third level, in which *task* operators act. Task-level operators apply and parameterize physical and statistical operators to construct descriptions that are relevant in the context of the planning domain. For example, at the physical level we can construct a vector that describes the movement of an aggregation of agents, and apply other physical and statistical operators to analyze the movement. For an AFS scenario, one useful application of these operators is to extract the component of the movement vector that lies in the direction of the opponent forces: does a plan call for an attack or a retreat with respect to the aggregated agents? In other words, we use the general, implicit goals of the planning simulation to impose a task-level interpretation on physical data. Examples of the results of operators at this level include the following:

- *Aggregate agents:* By aggregating agents we create new, composite agents with properties that overlap with those of individual agents. These can be generated for an entire team, or based on properties of the individual agents, such as spatial location or mass.
- *Aggregate agent movement:* From a planning perspective, aggregated agent activity such as movement is often much more informative of a plan as a whole than individual agent movement. Movement of an aggregate is modeled as change in the location of the center of mass.
- *Force allocation:* A plan assigns agents to match opponent agents and objects, by their position and mass. A simple reduction summarizes the "balance" of the assignment.
- *Penetration:* The aggregate of agents moves into opponent territory along a path at some specific speed, which can be measured via a combination of transformation and decomposition.
- *Offense versus defense:* One simple indication of this balance-related assessment is the aggregate mass of agents forward versus those to the rear.
- *Dispersion:* Some plans may call for an increase in the spread of forces over time, others for a concentration. This can affect the choices made for later actions during a scenario. It can be measured in a way similar to aggregate movement.

Note that although these terms have a domain-specific flavor to them, they also map clearly to the data produced by general-purpose operators at lower levels, and they are implemented in similar ways. That is, the data representation remains largely the same, sequences of numbers and symbols, but their interpretation is richer at the higher levels. We also observe that the terms above correspond roughly to our understanding of specific data analysis concepts, such as correlation (for allocation), skewness (for offense versus defense), variance (for dispersion), and so forth.

Finally, at the highest level, we have procedures for exploration and evaluation. These take the information produced by the task-level operators to identify similarities and differences between sets of plans, in the form of comparisons, clusters, and complete partitions. The results of task-level operators can be used to define numerical similarity measures between plans, which in turn lets us apply conventional data analysis techniques. Further, task-level operators can be organized into an importance hierarchy. Force allocation and penetration are more significant for a plan, for example, than dispersion.

Processing in Daps at the evaluation level relies mainly on the SLINK clustering algorithm [20]. Although some strategies for cluster analysis exist, none quite matches our needs. Daps thus adopts the approach used in Aide, supporting the user's evaluation with a systematic search via weak heuristics, but

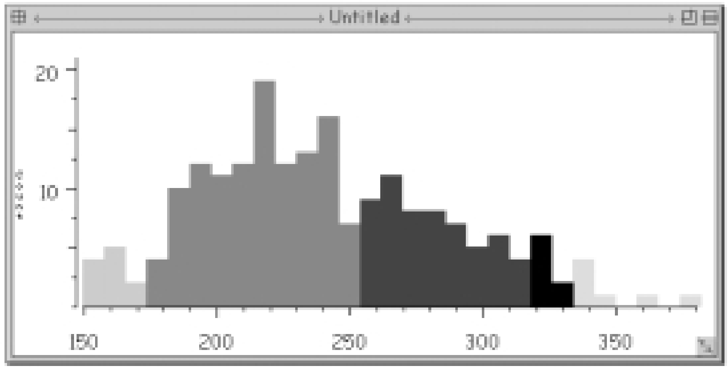

Fig. 3. Clusters in plan duration for 200 plans

also relying significantly on user guidance of the process. Everitt's discussion of cluster analysis also motivates our informal approach [6]:

> It is generally impossible *a priori* to anticipate what combination of variables, similarity measures and clustering techniques are likely to lead to interesting and informative classifications. Consequently the analysis proceeds through several stages with the researcher intervening if necessary to alter variables, choose a different similarity measure, concentrate on a particular subset of individuals, etc.

DAPS uses the hierarchy of task-level operators to impose an ordering on the data variables generated by the planner. It then carries out a systematic search, biased toward low-dimensionality clusters in data produced by highly ranked operators. The search space consists of plans that pass a quality threshold (or planning deadline) internal to the planner. The kinds of results DAPS generates can be displayed as conventional plots in CLASP; for example, Figure 3 shows four clusters in a set of 200 plans evaluated by the plan duration operator. (Not all operators and combinations lead to such clear-cut patterns; considerable work remains in refining DAPS's evaluation capabilities.) Duration is a simple static evaluation of similarities between plans. A more dynamic, opportunistic evaluation is possible in DAPS as well. We are currently experimenting with operators at the task and evaluation levels that monitor changes in dispersion, for example, and identify spatial clusters of agents that can be relevant for planning future stages of extended plans (i.e., beyond initial attack plans.) We are also developing visualizations, icon-based and otherwise, so that the results of DAPS can be integrated into the AFS interface, but these are not yet complete.

5 Discussion

The work we report here is preliminary. A number of limitations apply. In the area of planning, we have only considered a simple class of plans, those that involve multiple agents in coordinated movement over terrain, taking specific actions at the end. One of the next issues we must address is incorporating agent interactions, especially opponent interactions, into our analysis. Also, some aspects of planning are not captured by this approach. For more abstract problems that concern complex ordering relationships and object configurations, there may be no clear mapping to spatial and temporal sequences as in AFS. In the area of data analysis, we rely on a simple set of analysis techniques, and the number of plans we explore with them is not large; they are also relatively inefficient, which limits the number of plans that can be effectively analyzed. The simplicity of our approach was partly by design, to explore the potential difficulties in applying physical reasoning heuristics to data analysis in this context. To solve practical problems, however, we expect that we will need to extend the work to include more sophisticated methods and more complex heuristics to guide their application. Finally, in the area of reasoning about the physical aspects of plans, again we have used very simple heuristics, sufficient only for weak interpretation of patterns and guidance for DAPS.

Despite these points, we believe this work has significant promise. Our application is mixed-initiative planning in an environment that simulates physical object interactions, an area of significant interest in AI from both a practical and theoretical viewpoint. Our approach has three components. We have proposed a novel reformulation of the planning problem, as a search through sets of plan execution paths. We have developed a layered physical/task/evaluation IDA architecture, on top of a set of existing IDA components, to perform a search through this space. Finally, we have argued that this representation provides a kind of contextual knowledge that can support limited but useful guidance of the data analysis process. We are currently extending the range of DAPS; in the future we expect to carry out an empirical study of the effectiveness of the system in comparison with a more conventional approach to mixed-initiative planning.

References

[1] James F. Allen. Mixed initiative planning: Position paper. [WWW document]. Presented at the ARPA/Rome Labs Planning Initiative Workshop. URL http://www.cs.rochester.edu/research/trains/mip/, 1994.

[2] Scott D. Anderson, Adam Carlson, David L. Westbrook, David M. Hart, and Paul R. Cohen. CLASP/CLIP: Common Lisp Analytical Statistics Package/Common Lisp Instrumentation Package. Technical Report 93-55, University of Massachusetts at Amherst, Computer Science Department, 1993.

[3] Marc Atkin, David L. Westbrook, Paul R. Cohen, and Gregory D. Jorstad. Afs and hac: Domain-general agent simulation and control. In *AAAI-98 Workshop on Software Tools for Developing Agents*, pages 89–95, 1998.

[4] Jeffery Bonar and Blaise W. Liffick. Communicating with high-level plans. In Joseph W. Sullivan and Sherman W. Tyler, editors, *Intelligent User Interfaces*. ACM Press, 1991.

[5] Mark H. Burstein and Drew V. McDermott. Issues in the development of human-computer mixed initiative planning. In B. Gorayska and J. L. Mey, editors, *Cogni-*

tive Technology: In Search of a Humane Interface, pages 285–303. Elsevier Science, 1996.
[6] Brian Everitt. *Cluster Analysis*. John Wiley & Sons, Inc., 1993.
[7] Julian J. Faraway. On the cost of data analysis. *Journal of Computational and Graphical Statistics*, 1(3):213–229, 1992.
[8] George Ferguson and James Allen. TRIPS: An intelligent integrated problem-solving assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.
[9] George Ferguson, James Allen, and Brad Miller. TRAINS-95: Towards a mised-initiative planning assistant. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems*, pages 70–77, 1996.
[10] W. A. Gale. REX review. In W. A. Gale, editor, *Artificial Intelligence and Statistics*. Addison-Wesley Publishing Company, 1986.
[11] William A. Gale, David J. Hand, and Anthony E. Kelly. Statistical applications of artificial intelligence. In C. R. Rao, editor, *Handbook of Statistics*, volume 9, chapter 16, pages 535–576. Elsevier Science, 1993.
[12] David J. Hand. Intelligent data analysis: Issues and opportunities. In *Advances in Intelligent Data Analysis: Reasoning about Data*. Springer, 1997. 1–14.
[13] D.J. Hand. Patterns in statistical strategy. In W.A. Gale, editor, *Artificial Intelligence and Statistics*, pages 355–387. Addison-Wesley Publishing Company, 1986.
[14] Nira Herrmann, Abraham Silvers, Katherine Godfrey, Bruce Roberts, and Daniel Cerys. A prototype statistical advisory system for biomedical researchers II: Development of a statistical strategy. *Computational Statistics and Data Analysis*, 18:357–369, 1994.
[15] George Lakoff. *Women, Fire, and Dangerous Things*. University of Chicago Press, 1984.
[16] David Lubinsky and Daryl Pregibon. Data analysis as search. *Journal of Econometrics*, 38:247–268, 1988.
[17] Karen L. Myers. Advisable planning systems. In Austin Tate, editor, *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, pages 206–209. AAAI Press, 1996.
[18] R. Wayne Oldford and Stephen C. Peters. Implementation and study of statistical strategy. In W.A. Gale, editor, *Artificial Intelligence and Statistics*, pages 335–349. Addison-Wesley Publishing Company, 1986.
[19] Charles Rich and Candace L Sidner. COLLAGEN: a collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3/4), 1998.
[20] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *Computer Journal*, 16(1):30–34, 1973.
[21] Abraham Silvers, Nira Herrmann, Katherine Godfrey, Bruce Roberts, and Daniel Cerys. A prototype statistical advisory system for biomedical researchers I: Overview. *Computational Statistics and Data Analysis*, 18, 1994.
[22] Robert St. Amant. Navigation and planning in a mixed-initiative user interface. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. AAAI Press, 1997. 64–69.
[23] Robert St. Amant and Paul R. Cohen. Intelligent support for exploratory data analysis. *Journal of Computational and Graphical Statistics*, 7(4):545–558, 1998.
[24] Robert St. Amant and Paul R. Cohen. Interaction with a mixed-initiative system for exploratory data analysis. *Knowledge-Based Systems*, 10(5):265–273, 1998.