An overview of the project: Planning: A joint AI/OR approach

AI and OR approaches to planning

The general problem of planning a task is one of very broad scope Tasks may be as varied as house building, cooking, or robot assembly and the agent may be a person, a robot or an industrial organization. Nevertheless, when the planning process is formalized, problems from different domains are found to have much in common. Current work in Operational Research (OR) and Artificial Intelligence (AI) has concentrated on different aspects of the problem and, in the "Planning: a joint AI/OR approach" project we hope that an interdisciplinary approach will lead to a development of both these aspects.

In the ORapproach, the planning process falls into two stages.

1. The constituent "jobs" of a plan are specified together with their precedence relationships (i.e. requirements of the form that one job precede another). This information is represented in a graph with jobs as nodes and precedence relationships as arcs. This graph is termed a project network.

2. Various operations are performed on the project network to establish schedules and allocate resources.

OR work has been concerned with the second stage of computational operations on a given project network. No work has been performed on automating the preliminary stage of generating the project network. This has to be performed in an intuitive, not well understood and probably haphazard way.

It can be argued that the generation of a project network is important because of the structure it imposes on the task in hand. It forces component jobs to be isolated and necessary orderings between them considered. The project network can be used not only for predications of how the project will be done, but also as a tool to aid in monitoring its progress and allowing bottlenecks to be identified. Estimates of the time spent in drawing up a project network vary: the type of work, the amount of detail and the extent of previous knowledge are the major factors which affect the work load per job. Several examples which reflect this wide variation are given here:

A network of 120 activities for a new product line took 2 man weeks

A 400-event network occupied five men for three weeks

British Rail have quoted a case of 8 men spending 7 hours each on a 134-job.

So we see that a considerable amount of effort is expanded in the construction of project networks. Once a network is constructed and is in use on a project, much effort can also go into modifications to keep it up to date with actual progress on a task.

Steps towards automating the process of specifying constituent jobs for some task and for giving the precedence relationships between jobs, have been made in the AI work on plan formation. Formalisms have been developed for representing the data to the planning process, i.e. a description of the goals of the plan and the operations (jobs) of which it might consist (notably the representation of operator schemas to STRIPS - a problem solver developed at Stanford Research Institute, California). Such formalisms are completely general purpose and planners have been designed which can use such data to generate, fairly efficiently, plans to solve simple tasks. A major restriction on the applicability of the AI work on plan formation to the generation of project networks is that most of the work has concentrated on the production of plans with totally ordered sequences of primitive jobs rather than the networks familiar to OR workers. In effect, arbitrary precedence relationships will have been added to a solution network to render the constituent jobs totally ordered. Avoiding as much unnecessary sequencing in the project network as possible is important to permit effective scheduling by OR techniques. Some recent work at SRI has explored the use of a planner able to generate a plan as a partially ordered network of actions. This work forms the basis of our approach to aiding a user to generate a project network for some task.

Overview of the project

The Planning: a joint AI/OR approach project has initially been concerned with aiding a user in the process of constructing a project network. To do this, as in the SRI work, we have been investigating the use of a partially ordered network of actions to represent a plan at any stage of development. Our planning program proceeds by stages in progressively greater levels of detail where the current network is refined by expanding a particular node into more detailed actions and adjusting the orderings accordingly. Any orderings in the network result from the fact that either

(i) an action achieves a condition for a subsequent action.

or (ii) an action interferes with an important effect of another action and must be removed outside its range.

Our intention has been not only to write a program which will perform well within a particular problem domain but to structure the program in such a way that the various aspects of the planning process can be easily identified. In particular, the domain dependent aspects should be clearly distinguished from the remainder.

Accordingly we have attacked the problem under several headings:

1. Task formalism

A formalism has been specified to enable a task to be described in a hierarchic fashion. Task descriptions can be written independently of their use at higher levels and the formalism is intended to encourage the writing of job descriptions by experts at the highest level, by middle-management and by tradesmen allowing each to describe their tasks in a modular way and in their own terminology.

Within the specification will be information about

a) when to introduce an action into the plan

b) the effects of an action

c) what conditions must hold before an action can be performed

d) how to decompose an action

2. The Planner

We now have an operational program, NONLIN, which uses our current task formalism to generate plans. NONLIN operates on a partially ordered network of actions by expanding nodes and handling interactions between them. In general, an expansion will involve choosing between several alternatives and such choices will involve feasibility considerations. Some of the alternatives may generate unresolvable interactions with other parts of the plan and such failures may only become apparent at a much later stage in the planning process. A good planning system must not only provide good criteria for choosing between alternatives but also the ability to relate a failure to the choice point responsible and to modify the plan accordingly.

3. Optimization

For any real world task considerations of efficiency are important and the planner must find cost effective plans. Rather than keep alternative actions in the plan we propose that "good" choices are made at each level. For each alternative we must consider not only its individual costs and duration but also its contribution to the whole plan. Since we have a complete plan at each level of detail, critical path data can be used to guide such choices. Algorithms have been developed to calculate critical path data for a continuously altering network. The critical path package has been implemented and used with NONLIN to compute critical path data but has not yet been used to guide choices.

Of course, because the high level plans are inaccurate, such calculations will often point to the wrong choice and result in an inefficient solution. If at some more detailed stage we find that the solution is too bad (i.e. violates some predetermined constraints) the C.P. data will give a guide to suitable modifications. In general, such modifications will be to speed up critical jobs or find cheaper alternatives for jobs with large slack times. We require a modification procedure which allows the removal of actions from the plan without throwing away all the work done since they were introduced.

4. Modifying Plans

We have seen that recovery from a failure (either of feasibility or efficiency) involves removing certain actions while keeping the remainder of the plan intact. In order to do this we will build a structure which records the decisions made in generating the plan and the relationship between them. In particular, this structure (called a decision graph) records the relationship between the expansions made and the ordering introduced as a consequence. The program for modifying plans has now been designed but is not yet implemented.

Future work

The four aspects of our work described above reflect our preliminary thoughts about how a planning system should be implemented. We expect that many modifications must be made before a practically useful program can emerge. However, we feel that before there is any further development of the program it is important to gain experience of real-world problems and test out our current philosophy against practical examples.