A HEURISTIC RESOURCE ALLOCATION ALGORITHM FOR SYSTEMS WITH UNCERTAINTY IN REQUEST TIMING; A BACKTRACKING STRATEGY

Antti Huusko & Boris Segerståhl

Research Institute of Northern Finland, University of Oulu,
Linnanmaa SF-90570 OULU, FINLAND

ABSTRACT

This report describes an object-oriented approach to model representation. An object's behaviour is defined by sets of rules. The system provides control of the model's dynamic behaviour by providing simulation monitoring facilities. The main performance assessment is based on a heuristic analysis of simulation data which creates a specification of refinements to the model in order to better satisfy the predefined goals. The simulation life cycle is managed by means of a goal-directed rule system which examines the performance of a scenario. Based on the performance assessment, resources are reallocated by means of a backtracking strategy where the key features are the use of heuristic and domain-dependent search algorithms, and the concept of parallelism is used in the structuring of the optimal strategy.

INTRODUCTION

To find the optimal course of action is in many real world problems a difficult task, even when the target system does not seem extraordinarily complex. The difficulty is further increased when a decision maker has to make choices between several conflicting objectives, the system has many decision and output variables, and the structure of the system is not well known. In this situation an optimal decision is one that somehow establishes the best mix of outcomes. In some cases a lowest acceptable level is defined for the quality of a solution. Everything better than this minimum is acceptable. Optimality is replaced by "good enough" (or "acceptability").

A system contends with its environment to achieve its objectives. A problem solver must decide which operations to perform to achieve a given goal. Another difficulty arises from that a decision maker does not act in an isolated world. He must consider the risks of the alternative decisions.

The determination of the optimal policy usually requires that the optimum decision must be specified for the known time span. One realistic view is to consider this period to be divided into several time segments or stages during which a particular policy is followed. The solution can then be thought of as a plan where actions follow each other. One well known example is resource allocation in production scheduling. The importance of scheduling and operational control of production processes evolves from the need to realize a given task in an optimal way. Actually, the theory of scheduling as an area of operation research is well known but still many problems arise in practical implementation. The schedule generation does not allow for a simple solution. That is, there is no efficient algorithm for solving the general resource-constrained scheduling problem, and there is no hope that such an algorithm will ever be developed due to the high computational complexity of the problems. Therefore, the solutions are often obtained by means of heuristic methods and sub-optimal solutions are considered as sufficient.

The planning domain can be classified in terms of 1) size, 2) complexity, and 3) structure. Size of the planning problem refers to the number of elements that must be considered. Complexity relates to the number of interrelationships between elements. Structuredness means the degree of uncertainty about the precise nature of the relationships between elements. Resource allocation problems tend to be, in any reasonable abstraction, NP- hard. This implies that these problem types are probably inherently intractable in a well defined sense (Garey and Johnson 1979). By a more classical classification allocation problems are semi- or unstructured. Semi- and unstructured problems, however, do not usually fit a standard problem solving mold, and are generally solved by examining different scenarios, and asking 'what if' type of questions (Simon 1960). Examples of the structural complexities are multiple feedback loops in continuous systems described by differential equations. Predictability of the behaviour of the system is extremely poor if the time dependency of the system components is non-linear.

In this paper we describe a solution based on simulation of the resource allocation problem. It is known that a simulation approach to resource allocation perfectly fits the framework of discrete-event simulation (Baker 1974). The interest of this study is to find means for solving reasonably complex, i.e., too complex for analytical solution but not for simulation, resource allocation problems. This kind of system has inputs and outputs which are connected to each other by a dynamic behaviour. Another objective is to improve the efficiency and effectiveness of extracting information from simulation models. One of the main advantages of simulation is its ability to experiment with a simulated environment. The end-user is engaged in a process of "what if" analysis based on the simulation model. The result of this process is either the design parameters or the modifications of the systems being simulated. Both the process of selecting design parameters and the process of making recommendations for modifying a system based on a simulation model can be very lengthy and may involve many simulation runs.

Combining artificial intelligence concepts with traditional simulation methodologies yields a powerful design support tool known as knowledge based simulation. This approach turns a descriptive simulation tool into a prescriptive tool which orders goals according to predefined criteria (Reddy 1987, 162-166).

DOMAIN CHARACTERISTICS

In this section the characteristics of the resource allocation domain is defined and the generic problem is shown which is solvable by the method described in this paper. The allocation is considered to be a process which is controllable to the predefined state or states during the plan using predefined policy functions. These policy functions are constrained like control variables in control engineering problems. The system consist of an object process, a disturbance process and a regulator process. The disturbance process describes the environment with which the system interacts. The object process represents operation and dynamics of the system. The regulator process represents decision making system which selects from among all possible system behaviour the action it considers potentially most effective.

The resource allocation process described above differs from the common scheduling problems in some aspects. For instance in manufacturing scheduling problems the resource allocation does not affect the dynamics of the other tasks which are coming later into the production. Only the overall dynamics of the system is affected. In this system the scheduling of activities interacts with the dynamics of the process. This is due to the time varying dynamics of the scheduled processes.

OPTIMALITY OF THE RESULT

For the complex problems typical of systems and policy analysis, true optimization is a myth (Miser and Quade 1988). If there were no other reason, the gaps that are always present between the real-world problem situation and the formulated problem and the models chosen to represent it preclude any true optimization of the real-world situation.

There is often more than one possible solution. This means that an acceptable rather than an optimal solution is usually the "right" answer. The optimality of the solution may be measured in different ways. For instance the final state, speed of transition from initial state to final state, cost of transition, and value of the predefined performance index are usually included in quality measures.

The quality of the plan is not only the quality of the final state but it is a function of the system's state vector and the value of descriptive parameters integrated over the planning horizon. In the context of the plan of actions which constitute the optimal policy, actions must satisfy predefined constraints during the evolvement of the action chain. This kind of acceptable solution has a trajectory behaviour. This behaviour may be reduced to a figure of merit via numerical or statistical techniques. Also the structural behaviour of the system has its own value when the structure is only partially known beforehand or the structure fluctuates during the analysis process. In a structural simulation, the structural behaviour of a system is generated to depict a time-indexed sequence of structures of the system (Ören and Zeigler 1987, 131-134).

In order for an optimization analysis to be feasible, there must exist a prescriptive mathematical model that will permit an optimal alternative to be derived, either by a mathematical algorithm or by a controlled trial-and-error procedure. Simulation is a trial-and-error procedure where true optimization is not possible. Therefore simulation is considered to be a black box experiment where the structure and causal relations of the system cannot be utilized in optimization. In the planning context the black box view has its limitations. Therefore we introduce an interactive optimization approach where the solution is controlled by setting initial parameters and adjusting solutions. Jones (Anthonisse et. al. 1988, 413-419) introduces the term grey box for this type of optimization; the traditional single black box is replaced by a network of black boxes with user intervention required whenever one of them completes execution. In this way, the human planner guides the computer towards promising parts of the solution space.

RELATED WORK

There is no standard analytic strategy for solving resource allocation problems. Operations research uses a wide variety of approaches, each adapted to the problem situation being dealt with. The classic analytic strategy is to use a modelling approach that replicates the essential features of the problem situation as closely as possible. This heritage from operations research work served well in early systems analyses in which the structures were perceived as clear and the complexities and uncertainties modest (Miser and Quade 1988). The traditional operations research techniques for scheduling applications are simulation, network methods, combinatorial procedures, and heuristic approaches (Eilon 1979). The choice of technique usually depends on the problem complexity, the type of the model, the choice of objective, and other factors (such as wether several alternative solutions are required).

Network methods are inapplicable to dynamic scheduling because the precedence network is constantly changing. Combinatorial procedures (i.e. the use of blind search) can be ruled out, because of the complexity of the problem. The only remaining traditional computer-based techniques that are applicable are simulation techniques and heuristic approaches. However, simulations have to be interpreted by skilled scientists before a naive user can readily understand them. The 'cycle time' for simulation (from user's query to answer) is likely to be too long. Existing heuristic-scheduling programs have limited intelligence, because of the very simple knowledge representation used (Eilon 1979).

On the other hand the achieving the optimal course of actions has the same characteristics as the planning problem known in the artificial intelligence community. In planning a planner tries to construct a course of action to achieve a set of goals. A problem of reasoning about actions (Simon 1966) is given in terms of an initial situation, a terminal situation, a set of feasible actions, and a set of constraints that restrict the applicability of actions. The task of the problem solver is to find the best sequence of permissible actions that can transform the initial situation into the terminal situation.

Although significant advanced were made during the 1970s, the rate of progress has levelled off over the last decade, and major technical obstacles remain (Fiksel and Hayes-Roth 1989). The knowledge required for solving realistic planning problems has great temporal and conceptual complexity, as well as inherent instability and uncertainty. Consequently, existing algorithms for plan construction and optimization have had only limited success, even in relatively narrow problem areas (Fiksel and Hayes-Roth 1989, 16-23). A number of problems in current planning techniques have been identified in (Doran 1984). However, current AI planning research is just beginning to tackle the problem in temporal and resource-restricted domains (Bell et. al. 1987).

For solving resource allocation problem of the type described above we have to resort to simulation with domain dependent knowledge functions and approximation algorithms, that deliver acceptable solutions within an acceptable amount of time. It is just one step further to embed such algorithms in a heuristic setting. The solution is found by means of a directed trial-and-error procedure, in which man and machine divide tasks in accordance with their respective capabilities.

FRAMEWORK OF THE SOLUTION

The problem solving method presented here is focused on representation of the two different modeling aspects: 1) the modeling (representation) of the physical system, and 2) the modeling of the problem solving itself, i.e., the model of the problem solving strategy and tactics.

The basic modeling of the resource allocation problem is based on the combined discrete-continuous simulation pradigm. Differentiation into one of the two classes is based on the way the descriptive variables of the system change, e.g., how the system advances with time. Continuous dynamics is modelled by continuous-change models, which can be described by ordinary or partial differential equations, and the resources are modelled by sequencing time events. The system is partially structured using parallel processes. A parallel process is defined in a conventional way as the collection of all concurrent independent sequences of system states and events. The applicability of these prarallel processes is discussed more thoroughly below.

The physical system is modeled using an object-oriented approach. In an object oriented simulation model each object has a state (some sort of structured content), behaviour (a set of methods that operate on its state) and communication capabilities with other objects (sends and receives messages).

The problem solving strategy is divided into three hierarchical layers: 1) simulation and generation of the plan network, 2) analysis of the constructed network, and 3) design of the next simulation if the constraint satisfaction of the network cannot be found so that the acceptability criteria is satisfied. These steps generate a partially hierarchical plan which is revised according to the acceptability criteria of the plan.

Simulation is a good alternative when the system dynamics is poorly known. The problems encountered in AI planning can be avoided in simulation, e.g., temporal complexity in simulations is dealt with using a monotonically increasing system clock. Structural complexity which is the main difficulty in turning the simulation from a descriptive tool to a prescriptive tool is maintained both in the goal seeking capabilities of the parallel processes and the analysis and the possible relaxation of the generated project network. The project network is used only in acceptability analysis and for guiding the next simulation towards a satisficing solution. The concept of parallel processes is important when the processes is guided towards the subgoals independently. This strategy for seeking optimality is similar to the grey box idea presented above.

The network analysis (output analysis) is based on the backtracking strategy. The foundations of this strategy are in state space representation. The result of both resource allocation processes can be described as a state transition diagram, with transitions corresponding to activities. The diagram represents a sequence of states (or activities) from a given start state to a certain goal state. A simulation model may also be described as a state transition diagram, again with activities corresponding to states. In a simulation, however, there is a transition back from the goal state to the start state (Lee and Miller 1986, 15-25), so that the processing of the model can be backtracked through repeated iterations.

STRUCTURE OF THE KNOWLEDGE BASED SIMULATOR

The simulator design presented here is based on the knowledge based (intelligent) simulation paradigm. The coordination of symbolic reasoning and numeric computation is a difficult task in intelligent simulation. It has been realized that if applied separately, neither symbolic reasoning nor numeric computing can successfully address all problems. Therefore complex problems cannot be solved by purely symbolic or numeric techniques and coordination between them is needed (Kitzmiller and Kowalik 1987, 85-90). We are implementing a knowledge based methodology in our simulator two ways: 1) Embedding an expert systems within a simulation to improve the behaviour of the objects. Another possibility is to increase capabilities of the simulator with symbolic reasoning, e.g., the next event scheduling in the simulator is important for the optimality criteria selected, and 2) using expert systems methodology for post simulation analysis. In our case the post simulation expert system is used for network analysis.

The aim of the first approach is to insert look-ahead schemes into the simulation for minimize the trial and error procedures (search) in the simulation study. This is comparable to some extent with conditional events in traditional simulation (Kreutzer 1986). Implementation of this strategy plan of action is developed in a fragmentary manner during the simulation run adding actions into the plan. A complete plan is synthesized in this strategy as opportunities arise. This kind of opportunistic systems is constructed using a "black board" architecture

(Hayes-Roth 1985, 251-321) in which multiple knowledge sources and intelligent objects contribute to the development of alternative plans. The knowledge functions are used in this "intelligent simulation" phase in a strictly monotonic manner. This approach is a good alternative when the inference engine does not support temporal reasoning or it is not possible to use the truth maintenance schemes known in the more sophisticated systems.

The second stage in the analysis is more fundamental for simulation studies. In this model the expert system is a tool that helps the user with the analysis of a finished simulation model or the development of a model which converges towards an acceptable solution. In output analysis the expert system backtracks to the simulation time point where the deviation does not yet violate the acceptable solution. This kind of analysis adds a strategic component to the problem solving.

We are using a micro computer as a platform for the simulator and therefore it is realistic to base the development of the "intelligent simulator" on the upgrading of an existing one. The structure of the simulator is object oriented running in the Smalltalk programming environment. The base structure is presented in (Goldberg and Robson 1983). In the object oriented simulation the world view is process oriented. Parallel processes are implemented through processes in a Smalltalk programming environment (Goldberg and Robson 1983). Process orientedness provides locality of object: each process routine in a model specification describes the action sequence of a particular model object or objects. This kind of world view is well suited for the optimality strategy we are implementing because the system's object class descriptions specify how to deal with objects in their class definitions. The knowledge based components are easily embedded in this class hierarchy.

The primary difference between discrete event simulation models developed in the AI environment and models developed in conventional programming environments is the ability to use a rule system to model decision making (Egdorf and Roberts 1988). This leads to a more comprehensive view of an activity in the simulation produced in the AI environment. In conventional environment, emphasis is placed on modeling the exercise of physical capabilities. This leads to a model of sequences of activities as a sequence of events: begin activity - end activity.

With the ability to model the exercise of cognitive cabability, each activity is modeled in our simulator as the event sequence presented in (Egdorf and Roberts 1988):

- Begin Cognitive activity to determine next action.
- End Cognitive activity.
- Begin Physical activity
- End Physical activity.
- Begin assessment activity to determine result of physical activity.
- End assessment activity.

The new sub-activities that represent the cognitive and assessment portions allow modeling of command and control systems, and allow a model to be built that performs analysis of the decision making process itself, rather than just performance analysis of a physical system (Egdorf and Roberts 1988).

CONCLUSIONS

Object oriented programming is a powerful tool for the design of intelligent simulation. This approach allows for the inclusion of knowledge based methods into the simulation in ways that are not easily realizable using classical techniques. In this paper the structure of the intelligent simulator is presented. The project is in the development stage of the simulator. Next step in this development phase is to increase the system's knowledge based capabilities. Because the intelligence is a goal-directed and adaptive knowledge processing ability. Two important extensions will be explored: Adding knowledge-based intelligence which has an ability to perform simulation studies defined several scenarios within which they can simulate a system to increase their knowledge about: 1) behaviour of the model or 2) sensitivity of the behaviour of a model to parameters or operating conditions.

REFERENCES

Anthonisse, J.M., Lenstra, J.K., Savelsbergh, M.W.P. Behind the Screen: DSS from an OR Point of View. *Decision Support Systems.* 4(1988): 413-419.

Baker, K.R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley and Sons.

Bell, C., Currie, K., Tate, A. 1987. *Time Window and Resource Usage in O-Plan.* AIAI-TR-32, University of Edinburgh.

Doran, J. 1984. *Planning systems and expert systems*. Planning session report, Alvey IKBS research theme: IKBS, Workshop number 1. Abingdon, 5 - 6 May.

Egdorf, H.W., Roberts, D.J. 1988. *Discrete event simulation in the artificial intelligence environment.* In Ai papers by the Society for Computer Simulation International.

Eilon, S. 1979. *Production Scheduling*. In OR'78 (K.B. Haley, Ed.). Amsterdam: Noth-Holland.

Fiksel, J., Hayes-Roth, F. Knowledge Systems for Planning Support. *IEEE Expert.* Fall(1989): 16-23.

Garey, M.R., Johnson, D.S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.

Goldberg, A., Robson, D. 1983. *Smalltalk-80: The language and its implementation.* Reading: Addison Wesley.

Hayes-Roth, B. A Blackboard Architecture for Control. *Artificial Intelligence*. 26(1985): 251- 321.

Kitzmiller, C.T., Kowalic, J.S. Coupling Symbolic and Numerical Computing in Knowledge-Based Systems. *AI Magasine*, Summer(1987): 85-90.

Kreutzer, W. 1986. *System simulation programming styles and languages*. Reading: Addison Wesley.

Lee, R.M., Miller, L.W. A Logic Programming Framework for Planning and Simulation. *Decision Support Systems*. 2(1986): 15-25.

Miser, H.J., Quade, E.S. 1988. *Analytic Strategies and Their Components*. In Handbook of Systems Analysis: Craft Issues and Procedural Choices (Miser and Quade eds.). United States: Elsevier Publishing Co., Inc.

Reddy, R. 1987. Epistemology of knowledge based simulation. *Simulation*. 48(4): 162-166.

Simon, H.A. 1960. *The New Science of Management Decision*. New York: Harper and Row.

Simon, H.A. 1966. *On reasoning about actions*. CIT #87. Carnegie Institute of Technology

Ören, T.I., Zeigler, B.Z. Artificial intelligence in modelling and simulation: Directions to explore. *Simulation*. 48(4): 131-134.