Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

A Heuristic Learning Algorithm and its Application to Project Scheduling Problems

Reza Zamani
University of Wollongong

Louie Athanasiadis
University of Wollongong

Li-Yen Shue
University of Wollongong

A Heuristic Learning Algorithm and its Application to Project Scheduling Problems

Reza Zamani
Li-Yen Shue
Louie Athanasiadis
Department of Business Systems
University of Wollongong

Abstract

We present a real time heuristic learning algorithm. This algorithm is characterized by the complete heuristic learning process, which consists of state selection, heuristic learning, and search path review. The execution of the search path review is controlled by the user specified heuristic learning threshold. The algorithm will return an optimal solution with zero threshold, and near-optimal solutions with non-zero thresholds. We base on the dynamic nature of the resources of a project scheduling problem to present an application approach, which includes definition of states, state transition operator, and the cost of transition between states. Important results with the Patterson's110 problems are presented.

Introduction

The project scheduling problems with resource constraints (PSRC) have been shown by Blazewicz[1] to be NP-Complete. Its complexity has led to the development of many solution methods since early 1960s. Among them the enumerative branch-and-bound procedures appears to be more effective than others. Some of the major works and reviews in this area include Elmaghraby[2], Demeulemeester et al[3], Sampson and Weiss[4], Demeulemeester and Herroelen[5], and the earlier works by Christofides et al[6] and Stinson et al[7]. The works by Patterson et al[8], who apply a backtracking algorithm to solve the PSRC problems, and Bell and Park[9], who apply the A* algorithm to solve the PSRC problems, represent another approach in implementing techniques of Artificial Intelligence to address this problem.

In this research, we present a heuristic learning algorithm SLAT*, and an approach to facilitate its applications to solve the PSRC problems. The development of SLAT* algorithm is based on the real time heuristic search algorithm LRTA*[10], which is a much improved version of the original A* algorithm.

The LRTA* Algorithm and The SLAT* Algorithm

The Learning Real Time A* (LRTA*) algorithm assumes non-overestimating initial heuristic estimates from every state to the goals state, and positive edge costs between states. At a front state i, a neighboring state j with the minimum heuristic function f(j) = k(i,j) + h(j) is selected, where k(i,j) is the edge cost from i to j, and h(j) is the heuristic estimate of j. Based on the rationale that the farther away a state is from the goal state the larger its solution estimate should be, the heuristic learning for state i is for the algorithm to update h(i) to the value of f(j) should the latter be greater than the former, and then the search continues from j. Otherwise, there will be no heuristic learning, and from j the algorithm continues the search. It has been shown that repetitive application of the algorithm to a problem will eventually find an optimal[10].

We add a path review component to LRTA* to form the SLAT* algorithm (Search and Learning with Threshold), and control its execution with the user specified heuristic learning threshold. The path review is carried out only when the accumulated heuristic learning has exceeded the given threshold. When that happens, from the front state i, the algorithm backtracks to its previous state i-1 and applies the same rationale to see if h(i-1) can be improved as a result of the improved h(i). The improvement of h(i-1) will lead to further backtracking to state i-2, and this backtracking procedure will continue until a state is reached where its heuristic remains unchanged or the root state, then the forward search resumes. This review procedure will ensure that, before the occurrence of the subsequent heuristic learning, the search path remains a minimum path. Hence when the goal state is found, the solution will be a near-optimal one within the range of the specified threshold. When zero threshold is provided, the algorithm, called SLA* algorithm, backtracks every time a front state experiencing heuristic learning, and hence the solution will be an optimal one.

The Approach for Solving Project Scheduling Problems

Definition of states : Based on the status of all activities of a project, we define a state as a partial schedule, which consists of all activities of a project in three sets - completed, in-progress, and unscheduled. The completed set consists of all activities which have been completed at the time a state is being considered. The in-progress set consists of activities to be scheduled at that moment and those activities which have been scheduled but not yet completed. The unscheduled set

consists of all activities which have never been scheduled. In this way, the transition from one state to the next corresponds to the changes of the contents of the three sets.

State transition operator : With the starting and the ending activities as the two added dummy activities to a project, the starting activity can be scheduled at the outset. Upon the completion of an activity and subject to the satisfactory compliance of the precedence relationships, we define the transition operator as : moving the completed activities from the in-progress set to the completed set, and from the unscheduled set moving those activities whose resource requirements can be met to the in-progress set, including no new activity.

State transition costs : The transition cost between two neighbouring states is the time interval between the two. Since the completion of an activity from the in-progress set will lead to a new scheduling decision and a new state, hence the transition cost between two successive states is the time interval between two successive activity completion events.

Approaches for estimating initial heuristic : The heuristic estimate of a state for the PRSC problem is a lower bound estimate for completing the remainder of a given project from that state. To obtain initial non-overestimating heuristic estimates, we ignore the resource constraint, and apply the conventional CPM technique to find the heuristic of a state as the duration of the longest remaining path to completion. Alternatively, we can ignore the precedence constraint, and divide the total resource requirements of all remaining uncompleted activities with the maximum available resource levels.

Performances with Patterson's 110 Problems

In order to demonstrate the performance of SLAT* algorithm, we have implemented the algorithm to solve the 110 problems designed by Jim Patterson[11]. With zero threshold, SLA* is implemented first to find optimal solutions, then SLAT* with thresholds is applied to problems where excessive search is required for SLA*. Table 1 and 2 summarise the performances for SLA*. We have noticed that problem 12 and 72 require the highest rounds of heuristic learning, and hence the largest cpu time 159 sec and 74 sec respectively, to find their optimal solutions. Table 3 and 4 show the results when non-zero thresholds are applied with the SLAT* algorithm.

Table 1 cpu time with SLA*

cpu (sec)	number solved	cumulative
0	44	44
0 < & <= 10	60	104
10 < & <= 30	3	107
30 <	3	110

Table 2 heuristic learning with SLA*

rounds of heuristic learning	number solved	cumulative
< = 1000	56	56
1000 < & <= 10000	47	103
10000 <	7	110

Table 3 Problem 12 with SLAT*

threshold	rounds of heuristic learning	solution	cpu (sec)
0	36856	13#	159
1	2433	13	12
2	5	13	0
3	0	13	0
4	0	13	0

Table 4 Problem 72 with SLAT*

threshold	rounds of heuristic learning	solution	cpu (sec)
0	42946	41#	74
3	5079	42	19
5	2123	42	8
11	681	42	4
13	38	44	0

optimal solution

Conclusion

We have developed the SLAT* algorithm and proposed a method to implement it to solve the project scheduling problems with resource constraints. As shown with the results with Patterson's 110 problems, the non-zero threshold SLAT* may be applied to problems where excessive search is required, and the solutions are either near-optimal or even optimal with much improved computation times.

References

1. J. Blazewicz, J. K. Lenstra and A. H. G. Rinnooy Kan (1983) Scheduling Projects to Resource Constraints: Classification and Complexity, *Discrete Appl Math.* 5, 11-24.
2. S. E. Elmaghraby (1995) Activity nets: A guided tour through some recent developments. *European J. Oper Res.* 82, 383-408.

3. E. Demeulemeester, W. Herroelen, W. Simpson, S. Baroum, J. Patterson, and K.Yang (1994). On a Paper by Christofides et al. for Solving the Multiple-Resource Constrained Single Project Scheduling Problem, *European J. Oper Res.* 76, 218-228.
4. S. Sampson and E. Weiss (1993) Local search Techniques for the Resource Constrained Project Scheduling Problem, *Naval Res. Logistics*, 40, 665-675.
5. E. Demeulemeester and W. Herroelen (1992) A Branch-and -Bound Procedure for the Multiple Resource-Constrained Project Scheduling problems, *Mgmt. Sci.* 38, 1803-1818.
6. N. Christofides, R. Alvarez-Valdes and J. M. Taramit (1987) Project Scheduling with resource Constraints : A Branch and Bound Approach, *Euro. J. Opl. Res.* 29, 262-273.
7. J. P. Stinson, E. W. David and B. M. Kuumawala (1978) Multiple Resource-Constrained Scheduling Using Branch and Bound, *AIIE Trans.* 3, 91-98.
8. J. H. Patterson, F. Talbot, R. Slowinski, and J. Weglarz (1990) Computational Experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems, *European J. Oper Res.* 49, 68-79.
9. C. E. Bell and K. Park (1990) Solving Resource-Contrainted Project Scheduling Problems by A* Search, *Naval Research Logistics*, 37, 280-318.
10. R. Korf (1990) Real Time Heuristic Search, *Journal Of Artificial Intelligence*, 42, 189-211.
11. J. H. Patterson (1984) A Comparison of Exact Procedures for Solving the Multiple Constrained Resource Project Scheduling Problem, *Management Sci.*, 30, 854-867.