Automation in Construction 18 (2009) 434–443

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints

Po-Han Chen *, Seyed Mohsen Shahandashti

School of Civil and Environmental Engineering, Nanyang Technological University, 639798 Singapore

A R T I C L E I N F O

Article history:
Accepted 24 October 2008

Keywords:
Multi-project scheduling
Metaheuristic
Genetic algorithm (GA)
Simulated annealing (SA)
Heuristic rules
Multiple resource constraints

A B S T R A C T

Since scheduling of multiple projects is a complex and time-consuming task, a large number of heuristic rules have been proposed by researchers for such problems. However, each of these rules is usually appropriate for only one specific type of problem. In view of this, a hybrid of genetic algorithm and simulated annealing (GA-SA Hybrid) is proposed in this paper for generic multi-project scheduling problems with multiple resource constraints. The proposed GA-SA Hybrid is compared to the modified simulated annealing method (MSA), which is more powerful than genetic algorithm (GA) and simulated annealing (SA). As both GA and SA are generic search methods, the GA-SA Hybrid is also a generic search method. The random-search feature of GA, SA and GA-SA Hybrid makes them applicable to almost all kinds of optimization problems. In general, these methods are more effective than most heuristic rules. Three test projects and three real projects are presented to show the advantage of the proposed GA-SA Hybrid method. It can be seen that GA-SA Hybrid has better performance than GA, SA, MSA, and some most popular heuristic methods.

1. Introduction

Companies usually manage multiple projects simultaneously which makes scheduling and decision-making tough and time-consuming. A lot of heuristics have been proposed for scheduling of multiple projects with or without resource constraints [1–7]. However, their effectiveness is not consistent with the type of problem and sometimes none of them is effective. Some exact mathematical techniques have also been proposed which are not effective because of the combinatorial nature of the problems. These methods are often time-consuming, too.

Metaheuristics like genetic algorithm, simulated annealing, tabu search, artificial neural networks and their hybrids are used in various engineering fields. Nevertheless, their applications in multi-project scheduling are rare. A model hybridizing genetic algorithm and a heuristic are recently proposed for multi-project scheduling [8]. In this model, genetic algorithm only identifies the priority of projects over one another, and the heuristic identifies the priority of activities over one another. This combination makes the model dependent on the heuristic. The objective is to minimize the makespan of projects. It is among the first attempts to use metaheuristics and non-traditional techniques for the optimization of multi-project scheduling problems. Activity preemption is not allowable in this model. The heuristic is based on Less Slack Time. A numerical example of five projects, each of which has a maximum of 12 activities, is analyzed using Less Slack Time and compared to the results of using other heuristic approaches as the priority rules. Latest Come First Served (LCFS), Shortest Processing Time (SPT), First Come First Served (FCFS) and Earliest Due Date (EDD) are the heuristic approaches considered in the comparison with Less Slack Time. The result of the hybrid genetic algorithm and proposed heuristic (i.e., Less Slack Time) is better than all the abovementioned heuristics.

In this paper, a new method that hybridizes genetic algorithm and simulated annealing (GA-SA Hybrid) is proposed for scheduling of multiple projects with multiple resource constraints. The proposed GA-SA Hybrid method will be compared to various most popular models and methods.

1.1. Multiple resources allocation algorithm

An algorithm based on next time frame (NTF) is used for multiple resource allocation [9]. It is a least impact algorithm and only assigns resources to activities that are allowed to start. The assignment is done through some time frames in which resource assignment remains constant.

Suppose that CT and NT represent the current time and next time respectively and CT is known. NT is defined as the following:

$$\mathrm{NT} = \mathrm{Min}[f(a), f(b), s(c) \,|\, \forall a \in As, \forall b \in Ap, \forall c \in Af] \tag{1}$$

where As is the set of activities which can start at CT, Ap is the set of activities which are in progress, and Af is the set of activities, whose

* Corresponding author.
E-mail address: cphchen@ntu.edu.sg (P.-H. Chen).

0926-5805/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2008.10.007

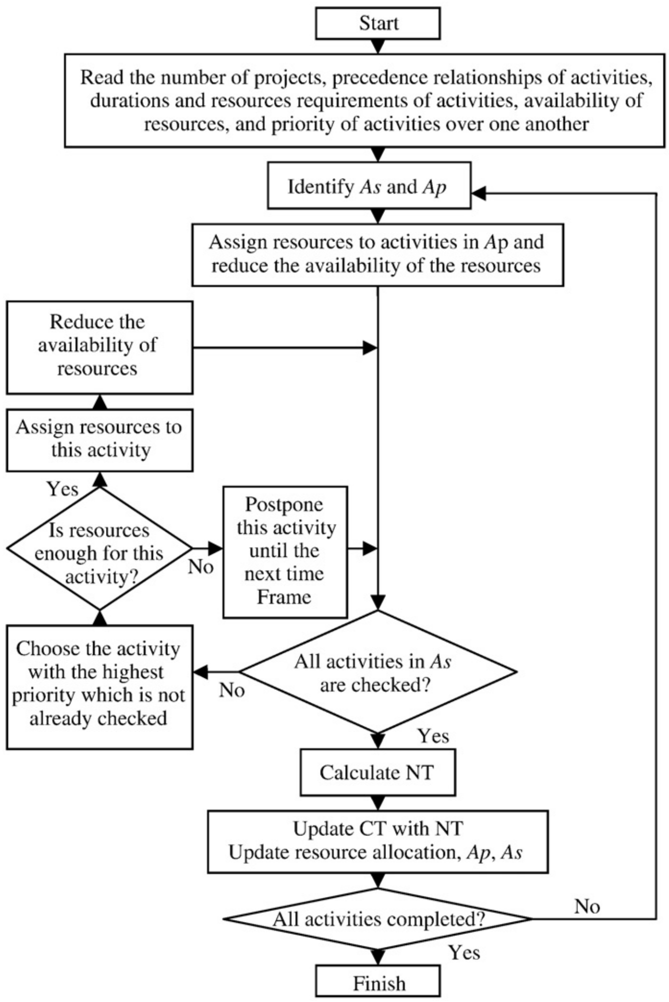

Fig. 1. Flow of multiple resource allocation.

predecessors are completed, that should wait till the end of their lead times. $f(i)$ and $s(i)$ represent the finish time and start time of activity i, respectively.

The next time frame (NTF) is the time frame between CT and NT. First, resources are assigned to activities which are in progress. Afterwards, the remaining resources are assigned to activities which could start according to their priorities. Then, CT and NT could be updated for the next time frame (NTF).

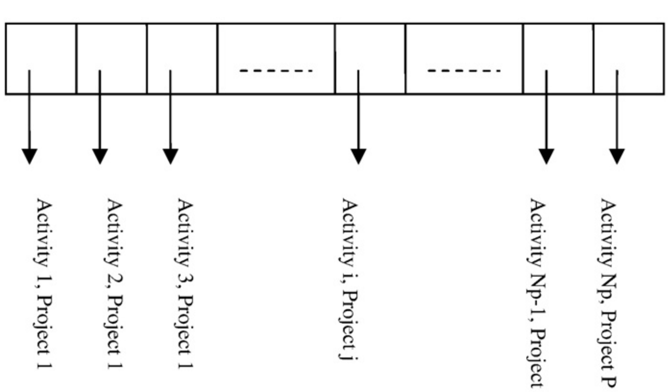

Fig. 2. Presenting the priority of activities from various projects as a string.

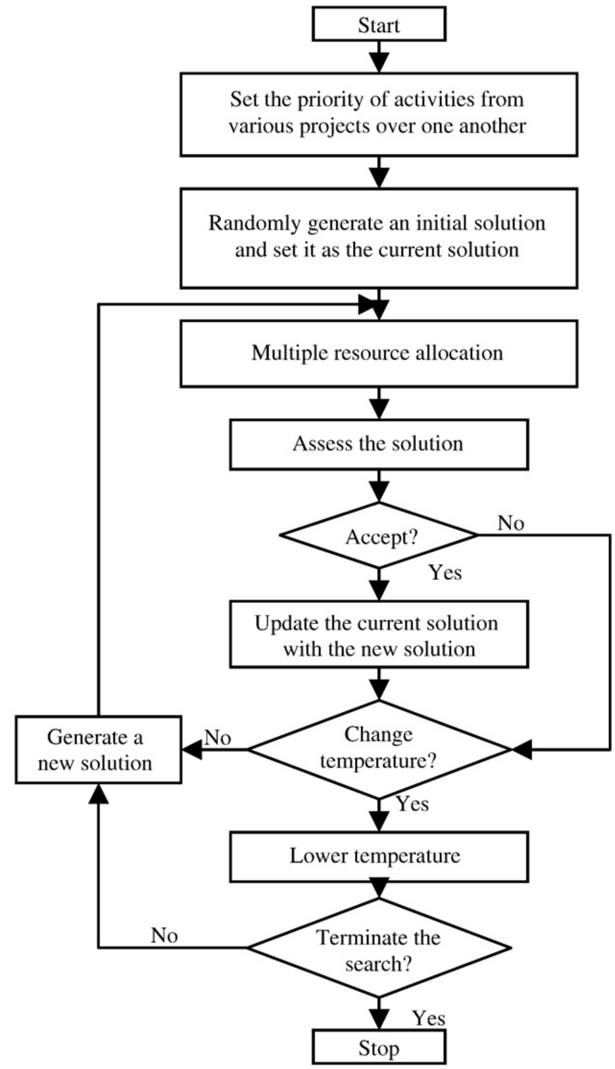

Fig. 3. Flow of simulated annealing (SA).

The following assumptions are considered for the multiple resource allocation process.

1. Resources are positive integers.
2. Preemption is not allowed. Activities cannot be split.
3. Precedence relationships among activities should be identified. Precedence relationships could be obtained using the critical path method (CPM).
4. The priority of activities that could start at the same time should be determined. (This is explained in detail in the “System of Coding” section.)

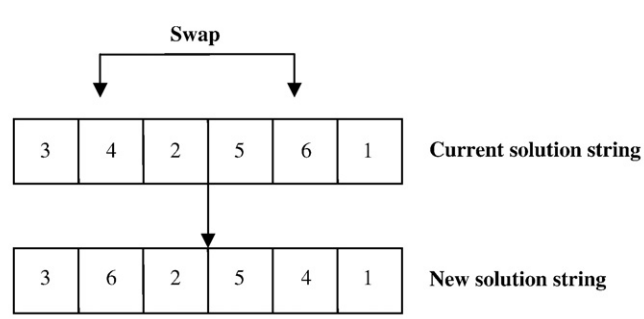

Fig. 4. Generation of a new solution.