Automation in Construction 75 (2017) 1–9

Contents lists available at ScienceDirect

Automation in Construction

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

An effective approach with feasible space decomposition to solve resource-constrained project scheduling problems

Zhenyuan Liu [a,b,*], Liu Yang [a], Raoyi Deng [a], Jing Tian [a]

[a] College of Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Key Laboratory of Education Ministry for Image Processing and Intelligent Control, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 8 May 2016
Received in revised form 14 November 2016
Accepted 24 November 2016
Available online xxxx

Keywords:
Project scheduling
Decomposition-based approach
Extended serial scheduling scheme
Double justification
Construction management

ABSTRACT

Resource-constrained project scheduling problem is a classic problem in construction project. Aimed at solving this problem, an effective approach with decomposition on time windows is proposed in this paper. This approach is to select one activity to do decomposition and to partition the feasible space of the original problem into some feasible subspaces, in which solutions are generated by using an extended serial scheduling scheme. Double justification is also performed in the process of searching in subspace. Four strategies for selecting activity to do decomposition, three strategies for decomposition and a strategy on sampling size in various subspaces are designed. The results of experiments on two real construction projects show that the strategy based on degree for selecting activity and the strategy based on initial schedule for decomposition can obtain the best results. When compared with some other exiting algorithms, it is proven that the decomposition-based approach is effective and competitive.

1. Introduction

The resource-constrained project scheduling problem (RCPSP) is a classic NP-hard optimization problem [1] which can be found frequently in construction project, production development etc. Various procedures for solving the RCPSP including exact and heuristic ones have been developed in the last several decades years [2,3]. Among them, branch-and-bound method is a typical exact procedure, which is able to find optimal solutions in feasible time [4,5]. However, exact solution procedures are restricted to small or medium-scale RCPSPs. In order to solve large-scale problems, heuristic methods have been proposed which could generate optimal or suboptimal solutions within acceptable time. Serial Scheduling Scheme (SSS) and Parallel Scheduling Scheme are two popular schedule generation schemes (SGSs) in this area, which can generate feasible schedules with priority rules-based selection of activities stage by stage [6]. In recent decades, these two schemes have been widely combined with lots of meta-heuristic algorithms to solve the large-scale RCPSPs. Besides, priority rules were proposed to construct schedules based on the SGS. The most popular priority rules include latest finish time (LFT), shortest process time, minimum slack etc. The priority rules can make different influences on the performance of heuristics [7].

* Corresponding author at: College of Automation, Huazhong University of Science and Technology, Wuhan 430074, China.
E-mail address: zyliu@mail.hust.edu.cn (Z. Liu).

http://dx.doi.org/10.1016/j.autcon.2016.11.012
0926-5805/© 2016 Published by Elsevier B.V.

During the past 30 years, kinds of meta-heuristic methods have been put forward to find more accurate solutions within shorter computing time. In their procedures, solutions are usually encoded firstly. Then the codes are visited by using some meta-heuristic strategy and finally decoded to generate schedules. Random-key and activity-list are two typical encoding methods [8]. Also choosing a decoding process according to characteristics of the problem reflects the flexibility of the algorithm [9]. The widely used meta-heuristic algorithms include Genetic Algorithms (GAs) [10–12], particle swarm optimization [13,14], simulated annealing algorithm [15] and ant system algorithm [16], etc. For example, Zamani presented a Genetic Algorithm with a magnet-based crossover operator [10], Kadam and Mane combined GA with local search algorithm [11], and Chen and Weng presented a two-phase GA model for RCPSP [12]. Georgios Koulinas et al. proposed a particle swarm optimization based hyper-heuristic algorithm by using random keys as the solution representation [13]. Similarly, Anantathanvit and Munlin extended the particle swarm optimization by regrouping the agent particle within the appropriate radius of the circle to solve the problem [14]. Anagnostopoulos and Koulinas proposed simulated annealing hyper-heuristics [15] and Li and Zhang put forward an ant colony optimization-based multi-mode scheduling algorithm [16].

As for the new approaches in recent years, such as bee algorithm [17], analogous immune algorithm [18], shuffled frog-leaping algorithm [19], and neurogenic algorithms [20] have also been applied to the RCPSP. For other heuristic algorithms, Wang and Fang developed a hybrid estimation of distribution algorithm [21] and Liu et al. designed an activity-list-based nested partitions algorithm recently [22]. Cheng et al. proposed a novel fuzzy clustering chaotic-based differential evolution with serial method [23].

Evaluation on the performance of these heuristic algorithms shows that the meta-heuristic often costs far more time than the heuristic. In order to saving computing time, some researchers reduce solution space by decomposing the RCPSP into sub-problem. Benders decomposition is an effective method to solve mixed-integer liner programming problem [24]. Recently, a hybrid bender decomposition algorithm outperforms pure constrain programming in solution quality and speed for solving project scheduling problem with multi-purpose resources, through decomposing the mixed-integer linear programming into a relaxed master assignment problem and a feasibility scheduling sub-problem [25]. These two sub-problems are connected by benders cuts to exclude infeasible solutions.

When searching feasible schedules in solution space, we often need to rank activities by checking their earliest finish (start) time and latest finish (start) time which can be called finish (start) time windows of activities to satisfy precedence relation constraints and resource constraints. These time windows are combined to form the whole feasible space without consideration of resource constraints. RCPSP has been solved with various algorithms in previous work, but there is still no research trying to decompose these time windows into sub-windows which may reduce searching space. In one of our work, a rough approach with the idea of decomposition has been developed and some rules for decomposition have been designed [26]. However, decomposition on feasible space has not been defined clearly. The SSS embedded in the procedure just operates directly on the original sub-windows without considering that the feasible start times of activities will change during the scheduling procedure. There are only some simple rules for decomposition and sampling. The numerical studies are not done on real cases and the results are not satisfying.

This work is an extension of our previous work in [26] where the procedure will be defined in a systematic way. The remainder of the paper is organized as follows. Section 2 provides the description of the RCPSP. And then the decomposition methods on time windows are proposed in Section 3. Section 4 introduces the decomposition method-based approaches and Section 5 shows several case studies about experimental design and result comparisons on the PSPLIB. Finally, some conclusions are presented in Section 6.

2. Problem description

In RCPSP, it is assumed that a project consists of J activities. Without loss of generalization, we can assume that activity 1 is the only start activity and activity J is the only end activity. The duration of activity j is d_j periods.

In the project, an activity $j \in \{2, \cdots, J\}$ may not be started before any of its immediate predecessors $i \in P_j$ (P_j is the set of immediate predecessors of activity j) has been finished. There are K-type of renewable resources. R_k indicates the capacity of resource k. $r_{jk}(\forall k)$ units of resource k are required to process activity j during every period of its duration.

The time parameters in the problem are all integer valued. We use D to denote the deadline of the project. The earliest finish time EF_j and the latest finish time LF_j of activity j can be obtained with this D. We use a set of integer decision variables $x_j \in [EF_j, LF_j]$ ($j = 1, \ldots, J$) to be the finish time of activity j. Another variable is A_t, the set of activities being in progress in period t.

Table 1
Earliest finish times and latest finish times.

Finish times	1	2	3	4	5	6
EF	3	4	5	6	6	10
LF	17	14	19	16	20	20

The model of RCPSP can be formulated as the following:

$$\min x_J \tag{1}$$

$$s.t.\, x_i \le x_j - d_j, \forall i \in P_j, \forall j = 1, \ldots, J \tag{2}$$

$$\sum_{j \in A_t} r_{jk} \le R_k, \forall k = 1, \ldots, K, t = 1, \ldots, D \tag{3}$$

$$x_j \ge 0, \forall j = 1, \ldots, J \tag{4}$$

The formula (1) shows the objective function which minimizes the completion time of the project. (2), (3), (4) are constraints. Constraint (2) considers immediate predecessor and successor relationship of all activities and ensures that start time of every activity is not earlier than the finish times of its immediate predecessors. (3) indicates that the total resource usage per period is in the range of availability. (4) ensures that finish time of an activity is not negative.

3. Decomposition on feasible space

3.1. A notation of feasible space

Actually, the feasible space of RCPSP can be derived from the constraints (1) to (4). But it is not obvious because of the resource constraints. If this type of constraints is not considered, the feasible space can be denoted as a J-dimension-space Ω which is actually the combination of finish time windows of activities.

$$\Omega = [EF_1, LF_1] \times [EF_2, LF_2] \times \cdots \times [EF_J, LF_J] \tag{5}$$

For example, Fig. 1 is the activity on node network of a project example. The deadline is set as 20 and the finish times obtained with CPM are shown in Table 1.

Feasible space of this example can be defined as:

$$\Omega = [3, 17] \times [4, 14] \times [5, 19] \times [6, 16] \times [6, 20] \times [10, 20]$$

In this case, solving the RCPSP can be regarded as a procedure of finding a schedule in Ω to satisfy resource constraints and precedence relation constraints with minimizing the makespan of the project.

3.2. Decomposition operator

With the above notation of feasible space, we can employ a decomposition operator to partition Ω into several subspaces, as Fig. 2 shows.

The finish time window of activity i is $[EF_i, LF_i]$. It is assumed that the time window will be partitioned into M feasible sub-windows: $[EF_i^1, LF_i^1]$,

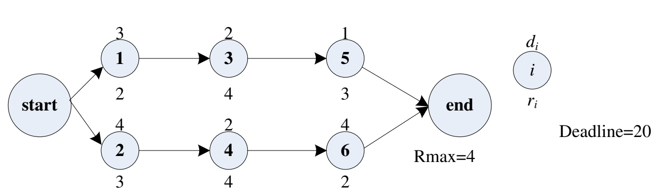

Fig. 1. Activity on node network of project example.

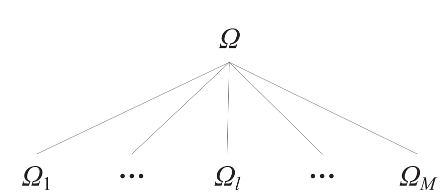

Fig. 2. Decomposition Operation on Feasible Space.