A Multi-Agent Optimization Method for Preemptive Resource-Constrained Project Scheduling

Yongyi Shou, Zhejiang University, Hangzhou, China
Wenjin Hu, Zhejiang University, Hangzhou, China
Changtao Lai, Zhejiang University, Hangzhou, China
Ying Ying, Zhejiang University, Hangzhou, China

ABSTRACT

A multi-agent optimization method is proposed to solve the preemptive resource-constrained project scheduling problem in which activities are allowed to be preempted no more than once. The proposed method involves a multi-agent system, a negotiation process, and two types of agents (activity agents and schedule agent). The activity agents and the schedule agent negotiate with each other to allocate resources and optimize the project schedule. Computational experiments were conducted using the standard project scheduling problem sets. Compared with prior studies, results of the proposed method are competitive in terms of project makespan. The method can be extended to other preemptive resource-constrained project scheduling problems.

KEYWORDS

Multi-Agent Optimization, Preemption, Project Scheduling, Resource Constraint

1. INTRODUCTION

The preemptive resource-constrained project scheduling problem (PRCPSP) is an important sub-problem of the resource-constrained project scheduling problem (RCPSP). The classic RCPSP is based on some assumptions, one of which is that activities are non-preemptive (Moukrim, Quilliot, & Toussaint, 2015; Cheng, Fowler, Kempf, & Mason, 2015). However, in project management practices, some activities may be preempted for the reason that resources are not in place in time or have to be re-allocated for other high-priority activities. Therefore, the PRCPSP has been proposed to relax the non-preemption assumption (Zhu, Li, & Shen, 2011). The PRCPSP allows temporary release of resources in the ongoing process, i.e., an activity can be suspended in order to release its resources for other activities, which makes project scheduling more flexible. In this paper, we mainly focus on the so called 1_PRCPSP, in which activities cannot be interrupted more than once (Ballestín, Valls, & Quintanilla, 2008).

DOI: 10.4018/IJITPM.2019010102

The PRCPSP is a typical NP-hard problem. Exact methods (e.g., branch-and-bound procedure and linear programming-based algorithms) can achieve the optimal solutions of the problem but are not efficient in solving large scale problems (Buddhakulsomsiri & Kim, 2006). Hence, swarm intelligent optimization (SIO) methods (e.g., particle swarm optimization and ant colony optimization) have been adopted to solve the PRCPSP (Demeulemeester & Herroelen, 1996; Damay, Quilliot, & Sanlaville, 2007; Shou, Li, & Lai, 2015). Swarm intelligent optimization methods find the best solutions through interaction and evolution of a population of simple agents. To some extent, swarm intelligent optimization methods can be regarded as agent-based methods, and have been demonstrated to be effective for project scheduling (Ballestín, Valls, & Quintanilla, 2009; Van Peteghem & Vanhoucke, 2010; Fink & Homberger, 2013; Han et al., 2013; Shou et al., 2015). An agent in these methods represents a self-contained problem-solving entity, and is characterized with autonomy, social ability, responsiveness, self-learning and proactiveness (Shou, Xiang, Li, & Yao, 2014). The interactions such as cooperation, coordination and negotiation among these agents contribute to the solution diversity and rapid convergence (Jennings, Sycara, & Wooldridge, 1998; Xie & Liu, 2009).

We propose a multi-agent optimization (MAO) method to solve the 1_PRCPSP. A multi-agent system is established to provide a framework which includes an agent interaction system and a basic service platform. In the multi-agent system, the MAO method consists of two types of agents: activity agents that are responsible for the resource requests and activity implementation, and a schedule agent that is responsible for the resource allocation and project scheduling. Activity agents and the schedule agent negotiate with each other to do resource allocation and project scheduling, and conduct scheduling optimization through an iterative improvement process.

The remainder of this paper is organized as follows. In Section 2, we present the preemptive resource-constrained project scheduling problem and its mathematical model. Section 3 develops a multi-agent optimization method for the 1_PRCPSP. Computational experiments and results are given in Section 4. Section 5 concludes this paper.

2. PROBLEM DESCRIPTIONS

A project can be described as an activity-on-node (AON) network $G = (V, A)$, where V is the node set representing a set of activities and A is the set of arcs describing the precedence relations between activities. The precedence relations between activities are of "start-finish" type (Kolisch & Hartmann, 2006), which means that for activities j_1 and j_2 where $(j_1,j_2)\in A$, activity j_2 cannot start until activity j_1 finishes. The project consists of $n+2$ activities with activity index $j = 0,1,\ldots, n, n+1$. Activity $j = 0$ and activity $j = n+1$ are dummy source and dummy sink activities respectively. The start time of activity j is s_j. The duration of activity j is a non-negative integer d_j, and $d_0 = d_{n+1} = 0$. The capacity of resource k is denoted by R_k, $k = 1, 2,\ldots, m$. The demand of resource k by activity j is r_{jk}, where $j = 0,1,2,\ldots,n,n+1$, $k = 1,2,\ldots,m$, and r_{jk} is a non-negative integer. Resources of the project are assumed to be renewable.

Activities in progress can be preempted only once, and the preemption happens at an integer time instant. The activity that is preempted will restart after a period of time with no additional cost (Afshar-Nadjafi & Majlesi, 2014). It will not be recognized as preemption if preemption happens exactly at the beginning or the end of an activity or if an activity restarts immediately after its preemption. The start, preemption, restart, and finish of activities do not consume additional time (Vanhoucke, 2008; Afshar-Nadjafi, Yazdani, & Majlesi, 2017).

The goal of the 1_PRCPSP is to minimize the project makespan under the conditions of precedence relations and resource constraints. Therefore, it should be decided whether an activity is preempted or not, when the corresponding preemption occurs, and when the rest part of the preempted activity restarts.