Expert Systems with Applications 42 (2015) 6039–6049

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A multi-agent optimization algorithm for resource constrained project scheduling problem

Xiao-long Zheng, Ling Wang *

Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Available online 10 April 2015

Keywords:
Resource constrained project scheduling problem
Multi-agent system
Optimization algorithm

ABSTRACT

In this paper, a multi-agent optimization algorithm (MAOA) is proposed for solving the resource-constrained project scheduling problem (RCPSP). In the MAOA, multiple agents work in a grouped environment where each agent represents a feasible solution. The evolution of agents is achieved by using four main elements in the MAOA, including social behavior, autonomous behavior, self-learning, and environment adjustment. The social behavior includes the global one and the local one for performing exploration. Through the global social behavior, the leader agent in every group is guided by the global best leader. Through the local social behavior, each agent is guided by its own leader agent. Through the autonomous behavior, each agent exploits its own neighborhood. Through the self-learning, the best agent performs an intensified search to further exploit the promising region. Meanwhile, some agents perform migration among groups to adjust the environment dynamically for information sharing. The implementation of the MAOA for solving the RCPSP is presented in detail, and the effect of key parameters of the MAOA is investigated based on the Taguchi method of design of experiment. Numerical testing results are provided by using three sets of benchmarking instances. The comparisons to the existing algorithms demonstrate the effectiveness of the proposed MAOA for solving the RCPSP.

1. Introduction

The resource constrained project scheduling problem (RCPSP) is to schedule project activities over time under a limitation of available resources (Brucker, Drexl, Mohring, Neumann, & Pesch, 1999). The RCPSP is one of the most intractable NP-hard problems in the field of operation research and management science (Mohring, Schulz, Stork, & Uetz, 2003). The RCPSP is very common in a variety of engineering fields, such as medical research (Hartmann, 1997), and software development (Alba & Chicano, 2007). Many scheduling problems can be formulated as the special cases of the RCPSP, such as job-shop scheduling, flow-shop scheduling, open-shop scheduling and project scheduling (Leung, 2004). It is very important for both academic research and engineering applications to develop effective solution algorithms for solving the RCPSP.

Over the past few decades, the RCPSP has attracted increasing attention due to its challenge and extensive engineering background. Early solution methods mainly focus on heuristic priority rules, e.g., minimum slack (MSLK), latest start time (LST) (Davis & Patterson, 1975), earliest start time (EST), shortest processing time (SPT), latest finish time (LFT), most total successors (MTS) (Cooper, 1977), worst case slack (WCS), and resource scheduling method (RSM) (Kolisch, 1996). To obtain the solutions with a higher quality, meta-heuristics have been widely adopted during recent years. Relevant research work in this area includes the following. Hartmann (1998) proposed a permutation-based genetic algorithm (GA) by employing a regret-based sampling method and LFT rule for initializing the population, where the proposed single point crossover that can always generate feasible offspring. Hartmann (2002) further developed a self-adapting GA by adaptively adopting a parallel and serial scheduling generating scheme (SGS). Bouleimen and Lecocq (2003) proposed an activity list based simulating annealing (SA) algorithm based on an alternated activity and time incrementing process. Zhang, Li, and Li (2005) proposed a particle swarm optimization (PSO) algorithm with both the priority-based and the permutation-based representation, showing that the permutation-based PSO outperformed the priority-based PSO. Debels and Vanhoucke (2007) presented an effective decomposition-based GA with a resource-based crossover. Chen, Shi, Teng, Lan, and Hu (2010) proposed an efficient hybrid algorithm by combining ant colony optimization (ACO) and scatter search (SS), and Lam & Li, 2012 proposed a chemical reaction optimization (CRO). Shi, Qu, Chen, et al. (2010) proposed an artificial bee colony (ABC) algorithm, where each solution was represented by a

* Corresponding author. Tel.: +86 10 62783125; fax: +86 10 62786911.
E-mail addresses: zhengxl11@mails.tsinghua.edu.cn (X.-l. Zheng), wangling@mail.tsinghua.edu.cn (L. Wang).

http://dx.doi.org/10.1016/j.eswa.2015.04.009
0957-4174/

random key. Wu, Wan, Shukla, and Li (2011) proposed a chaos-based improved immune algorithm. Fang and Wang (2012) proposed a shuffled frog-leaping algorithm (SFLA) by using an extended activity list representation, and they presented a speed-up method for evaluating new solutions based on some theoretical analysis. Paraskevopoulos, Tarantilis, and Ioannou (2012) proposed an event list-based evolutionary algorithm. More recently, Zamani (2013) proposed a competitive GA with a magnet-based crossover operator, Yannibelli and Amandi (2013) proposed a hybrid method by integrating the simulated annealing into evolutionary algorithm, Faghihi, Reinschmidt, and Kang (2014) proposed a novel approach of construction project application of the GA based on building information model, and Fahmy, Hassan, and Bassioni (2014) proposed a PSO with stacking justification for solving the RCPSP. Although many methods have been developed, it is still a challenge to solve the large-scale RCPSP effectively.

Multi-agent system (MAS) is an active research topic in artificial intelligence and expert systems. In a reasonable way, multiple agents collaborate with potential advantages for solving complex problems. The MAS based methods have been used to solve the RCPSP (Ren & Wang, 2011) and showed good performance, where an agent represented a resource and the behaviors of agents like collaboration, learning were adopted to handle the RCPSP. Moreover, motivated by swarm intelligence, several population-based algorithms (Liu, Wang, Liu, & Wang, 2011) have been widely applied to solve optimization problems. The intention of this research is to design a new multi-agent optimization algorithm (MAOA) by adopting the key behaviors of agent and the population-based mechanism of swarm intelligence to solve the RCPSP effectively. To be specific, each agent represents a feasible solution, which collaborates with others in a grouped environment via social behavior, autonomous behavior, self-learning, and environment adjustment. Through these elements, agents perform exploration and exploitation in a collaborative way among search space to obtain the promising solutions. We present the detailed implementation of the MAOA and the numerical comparisons with the existing algorithms for solving the RCPSP. The results show that the proposed MAOA is very competitive, especially for solving the large-scale problems.

The rest of the paper is organized as follows: In Section 2 the RCPSP is described; In Section 3, the framework of the multi-agent optimization algorithm is provided after introducing the concepts of agent and multi-agent system. In Section 4, the detailed design of the MAOA for solving the RCPSP is presented. Numerical tests and comparisons as well as the investigation on the effect of parameter setting are given in Section 5. Finally, the paper is ended with some conclusions and future work in Section 6.

2. Resource constrained project scheduling problem

Generally, the RCPSP can be described as follows: A project consists of J activities, where each activity j ($j = 1,2,\ldots,J$) can be processed with a duration d_j without preemption. There exists precedence relationship between several activities, which implies that activity j cannot be started until its predecessors are all finished. Moreover, there are K types of renewable resources available for the project. For each resource k ($k = 1,2,\ldots,K$), its availability at any time is R_k. The activity j requires r_{jk} units of resource k during its period of duration. The dummy activities 0 and $J+1$ represent the beginning and the end of the project, respectively. It is assumed that $d_0 = d_{J+1} = 0$ and such dummy activities consume no resource. The objective is to determine the start time of each activity to minimize the maximum complete time (makespan) of the project. For mathematical models of the RCPSP, please refer to Leung (2004).

In Fig. 1, a simple RCPSP with 6 activities is illustrated. For simplicity, only one type of resource is considered with an available amount of 2. In Fig. 2, a feasible schedule for the project in Fig. 1 with a makespan value of 12 is shown.

3. Multi-agent optimization algorithm

3.1. Agent and multi-agent system

Agent is a concept in the field of artificial intelligence (Minsky, 1988). An agent can be viewed as a computer system situated in a certain environment, such as a physical system or internet, with flexible autonomous action (Wooldridge & Jennings, 1995). Agents perceive input information though sensors and then perform resultant actions to affect the environment (Jennings, Sycara, & Wooldridge, 1998), as shown in Fig. 3 (Wooldridge, 2009). Agents are also capable of making decisions of choosing suitable actions to realize the goal independently without external intervention from other agents or human interference. Meanwhile, agent may have a variety of possible actions, including responsiveness, pro-activeness and social behaviors (Wooldridge & Jennings, 1995). Each agent can receive the information from the external

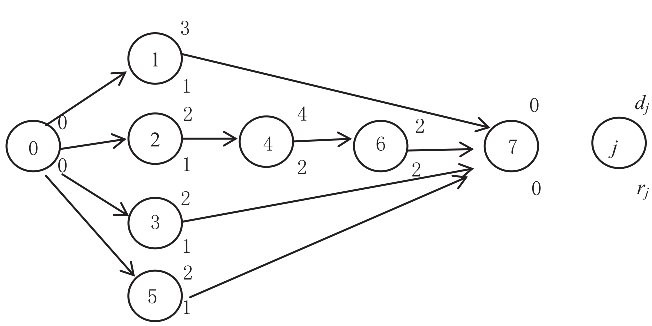

Fig. 1. An example of a project.

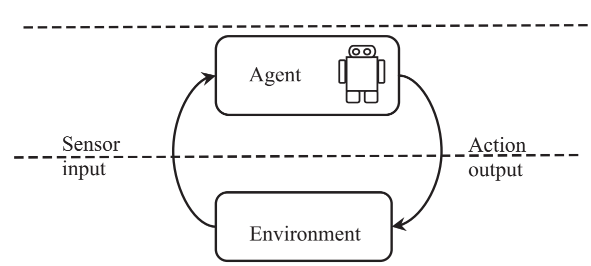

Fig. 3. An abstract view of an agent.

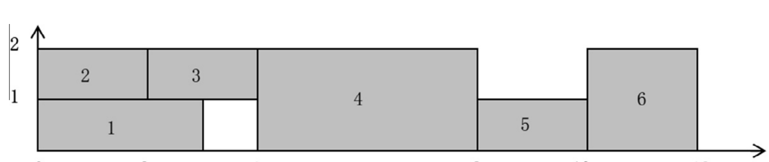

Fig. 2. A feasible schedule.