Expert Systems with Applications 38 (2011) 7102–7111

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem

Ruey-Maw Chen

Department of Computer Science and Information Engineering, National Chin-yi University of Technology, Taichung 411, Taiwan, ROC

ARTICLE INFO

Keywords:
Scheduling
Particle swarm optimization
Justification
Resource-constrained project scheduling problem

ABSTRACT

The studied resource-constrained project scheduling problem (RCPSP) is a classical well-known problem which involves resource, precedence, and temporal constraints and has been applied to many applications. However, the RCPSP is confirmed to be an *NP-hard* combinatorial problem. Restated, it is hard to be solved in a reasonable time. Therefore, there are many metaheuristics-based schemes for finding near optima of RCPSP were proposed. The particle swarm optimization (PSO) is one of the metaheuristics, and has been verified being an efficient nature-inspired algorithm for many optimization problems. For enhancing the PSO efficiency in solving RCPSP, an effective scheme is suggested. The justification technique is combined with PSO as the proposed justification particle swarm optimization (JPSO), which includes other designed mechanisms. The justification technique adjusts the start time of each activity of the yielded schedule to further shorten the makespan. Moreover, schedules are generated by both forward scheduling particle swarm and backward scheduling particle swarm in this work. Additionally, a mapping scheme and a modified communication mechanism among particles with a designed *gbest ratio* (GR) are also proposed to further improve the efficiency of the proposed JPSO. Simulation results demonstrate that the proposed JPSO provides an effective and efficient approach for solving RCPSP.

1. Introduction

Many applications involve scheduling notion, such as generating units planning of power plants (Saksornchai, Lee, Methaprayoon, Liao, & Ross, 2005), grid computing (Hou, Zhou, & Wang, 2006; Liu, Yang, Shi, Lin, & Li, 2005), control system (Park, Kim, Kim, & Kwon, 2002), food industrial (Simonov & Simonovov'a, 2002), network packet switching (Symington, Waddie, Taghizadeh, & Snowdon, 2003), classroom arrangement (Vejzovic & Humo, 2007) and manpower scheduling (Ohki, Morimoto, & Miyake, 2008). Generally, these problems commonly accompany the cost considerations related to certain constraints. A scheduling algorithm determines a schedule for a set of processes, satisfying the prerequisite constraints and minimizing cost. Scheduling problems differ markedly from case to case. One of the well studied scheduling problems is the resource-constrained project scheduling problem (RCPSP) (Hartmann, 2002); a variety of applications are part of RCPSP. RCPSP is a combinatorial optimization problem to schedule the activities such that the makespan (total completion time) of the schedule can be minimized, while satisfying given precedence constraint between the activities and resource constraint. The resource requirements of the scheduled activities per time unit do not exceed the given capacity limit of different types resources. However, the minimum makespan is hard to obtain since the inestimable situation of constraints. And RCPSP has been confirmed to be an *NP-hard* combinatorial problem (Blazewicz, Lenstra, & Rinooy Kan, 1983); it is hard to solve RCPSP in a reasonable time especially for large-scale scheduling problems. Restated, solving RCPSP requires considerable computation times for large instances.

Although there are some exactly algorithms such as branch-and-bound method (Brucker, Knust, Schoo, & Thiele, 1998; Jalilvand et al., 2005) is able to find optimal solutions of RCPSP. However, the execution time required is impractical when the number of activities increases. Comparatively, several priority-based heuristics (Buddhakulsomsiri & Kim, 2007; Li, Bettati, & Zhao, 1997) such as the latest finish time (LFT) and minimum slack (MSLK) (Edward & James, 1975), can solve RCPSP with shorter time, but they are hard to adapt to the constraints of problems dynamically. Hence, the sound solution is seldom obtained via heuristics.

Many studies solve the RCPSP by applying the metaheuristics-based schemes, such as genetic algorithm (GA) (Hartmann, 2002), simulated annealing algorithm (SA) (Bouleimen & Lecocq, 2003; Rutenbar, 1989), tabu search (TS) (Glover, 1989, 1990; Thomas & Salhi, 1998), ant colony optimization (ACO) (Lo, Chen, Huang, & Wu, 2008; Merkle, Middendorf, & Schmeck, 2002) and

E-mail address: raymond@mail.ncut.edu.tw

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.12.059

the particle swarm optimization (PSO) (Zhang, Li, & Tam, 2006), etc. The GA mimics the mechanism of natural selection as global evolution (Holland, 1987); then part of more superior solution is inherited via crossover operation, and increasing the diversity of solution via mutation process. Originally, simulated annealing was investigated by Kirkpatrick, Gelatt, and Vecchi (1983) as a stochastic method for combinatorial optimization problem. The optimal solution is a stable state when the thermal energy of the system minimized. The thermal energy is decreased by cooling down temperature parameter. Noteworthy, the SA applies a mechanism to avoid trapped on the local optimum by a probability during cooling down procedure. Tabu search is an approach proposed to prevent the search from sinking into the local minimum by recording the solutions which have been ever obtained. Therefore, the already obtained solutions in the following search can be avoided (Glover, 1989, 1990).

The ACO emulates the foraging behavior of ants (Dorigo & Gambardella, 1997). The ant left pheromone on the trail of the searched path from nest to the food source. The pheromone deposited on the way is for other ants to identify and communicate with each other. Additionally, the amount of pheromone is inverse proportional to the length of path; a large amount of pheromone is accumulated on the shorter path. The maximum amount of pheromone on the path can be regarded as an ant notification signal indicating where the shorter path is located at.

The particles swarm optimization (PSO) is first proposed by Kennedy and Eberhart (1995). In PSO, a swarm of particles spreads in the space and the position of a particle represents a solution of a dedicated problem. Each particle would move to a new position for the global optimal solution based on the global experience of the swarm and the individual experience of the particle. The PSO has been widely applied to solve the scheduling problems. Liu and Wang (2006) and Zhang, Sun, Zhu, and Yang (2008) solved flowshop scheduling problem (FSP) by means of the PSO, and Chen, Zhang, Hao, and Dai (2006) solved task scheduling in grid based on PSO. Zhang et al. (2006) used PSO to solve RCPSP; they showed that the PSO is applicable to various combinatorial problems and scheduling problems.

Besides the algorithm itself, some other schemes are combined with the algorithm to enhance the effectiveness and efficiency. There is a scheme named "justification" proposed by Valls, Ballest, and Quintanilla (2005), which is effective for improving the solution quality of the scheduling problems. The justification technique adjusts the start time of each activity in scheduling, and guarantees that the scheduling after justification is not worse even possible better than before one. Moreover, the efficiency of justification technique has been verified, it can apparently improve population-based algorithms such as GA while applying for RCPSP. In Valls et al. (2005), the justification implemented by double justify (DJ) applied to population-based algorithms, GA and SA have been tested, respectively, and the DJGA (GA applying DJ) and DJSA (SA applying DJ) outperform than all the state-of-the-art algorithms (such as GA, ACO). The performance evaluation comparison was also listed in Valls et al. (2005). Restated, the justification is able to promote the performance of population-based algorithms. Nevertheless, relatively few PSO studies with the combination of justification were devoted to solve RCPSP (no related literature was found). Hence, this study focuses on improving PSO algorithm based on the combination of PSO and justification for RCPSP, this proposed scheme is named justification particle swarm optimization (JPSO) herein.

Moreover, the suggested JPSO integrates two other designed mechanisms to further improve the efficiency, one is the mapping technique for enhancing the exploitation efficiency of justification, and the other is the adjusting ratio of communication topology of PSO for trade-off between exploration and exploitation. The simulation results demonstrate that both of these two schemes have significant improvement for solving RCPSP.

This article is organized as follows. Section 2 introduces the RCPSP. Section 3 presents the PSO. Section 4 presents the schemes of JPSO and how to solve RCPSP by JPSO. The simulated cases and results of experiments are displayed in Section 5. In Section 5, a complete comparative evaluation of the effectiveness and efficiency of the proposed JPSO algorithm as well as a comparison to other state-of-the-art approaches were presented. Finally, Section 6 presents the conclusions and discussions.

2. Resource-constrained project scheduling problem (RCPSP)

The scheduling problems have been applied in various fields. Among them, the resource-constrained project scheduling problem (RCPSP) is a general scheduling problem which involving activities need to be scheduled. Moreover, the RCPSP is confined to meet various constraints and achieves a certain objective. The studied RCPSP in this investigation is defined as follows:

1. The objective is to find the minimal *makespan* schedule.
2. There're $N+2$ activities, and each activity j has processing duration d_j ($j = 0, \ldots, N+1$). Meanwhile, activities are non-preemptive in the schedule. The activity 0 and activity $N+1$ are pseudo activities for indicating the start and end of schedule, respectively.
3. Activities have precedence constraint, let P_j be the set of immediate predecessors of activity j; the activity j cannot start to work until all of its immediate predecessors finished. Activity 0 is the source (start activity) that has no predecessors.
4. There are various renewable resources, constant amount renewable resources are provided at each time or period. Let

Table 1
30 activities case (j301_6) with precedence and resource requirement constraints.

Activity#	Successors			Activity#	Duration	Required resources			
						R 1	R 2	R 3	R 4
1	2	3	4	1	0	0	0	0	0
2	5	7	8	2	10	0	0	0	4
3	11			3	1	0	0	0	10
4	6	16		4	9	4	0	0	0
5	15	23		5	3	6	0	0	0
6	10	12		6	1	3	0	0	0
7	9	14	25	7	7	0	4	0	0
8	13			8	1	0	0	0	2
9	24			9	4	10	0	0	0
10	22			10	10	0	0	0	2
11	14	16	24	11	6	0	0	10	0
12	13	21		12	2	0	0	0	6
13	17	24	30	13	3	0	7	0	0
14	18			14	1	0	0	3	0
15	16	29		15	3	0	0	0	6
16	19			16	1	0	0	10	0
17	18			17	3	0	0	0	7
18	20	31		18	10	0	0	0	9
19	28			19	1	0	6	0	0
20	26			20	3	5	0	0	0
21	28			21	4	0	3	0	0
22	28			22	2	8	0	0	0
23	27			23	4	1	0	0	0
24	26	31		24	2	3	0	0	0
25	30			25	4	0	9	0	0
26	29			26	6	0	0	0	7
27	30			27	9	0	0	0	7
28	31			28	2	0	0	0	5
29	32			29	1	0	0	9	0
30	32			30	1	0	0	9	0
31	32			31	9	0	0	4	0
32				32	0	0	0	0	0
				Available resources		12	10	10	12