Research on a resource-constrained project scheduling problem in a hazardous environment and its staffing strategies based on PSO algorithm

Shuai Li, Zhicong Zhang, Xiaohui Yan, Kaishun Hu, Shaoyong Zhao

Department of Industrial Engineering, Dongguan University of Technology, Dongguan, People’s Republic of China

Abstract
We study a resource-constrained project scheduling problem in a hazardous environment considering some different strategies of staffing. An overhaul project of a nuclear power plant is chosen as a typical example. The same as the conventional projects scheduling, the problem is constrained by the availability of resources. However, due to the unique working environment in this project, the availability of resources is constrained by the accumulated amount of harm that the workers could withstand. As this extremely increases the complexity of the problem. In order to address the investigated problem, we propose a novel particle swarm optimization algorithm: probable mechanism-based discrete particle swarm optimization algorithm (PMPSO). The PMPSO algorithm is a discretization form of the traditional particle swarm optimization (PSO) algorithm. We use the PMPSO algorithm to solve the problem thinking of nine combinations of staffing strategies respectively. Comparison experiments of the combination of staffing strategies show that strategy of ‘3’ outperforms the other strategies. Numerical experiments indicate the adaptability of the PMPSO algorithm and the validity of the conclusion.

Article History
Received 19 November 2017
Accepted 26 June 2018

Keywords
Hazardous environment; resource-constrained project; scheduling; strategies of staffing

1. Introduction
Resource-constrained project scheduling (RCPSP) is one of the most notorious problem types in the context of scheduling. Owing to its complexity, RCPSP became a very hotspot for researchers either in scheduling field or in operations research. It involves the determination of each activity start time which achieves the minimum project duration without violating any of the precedence or resource constraints. Numerous numbers of researchers have devoted themselves to solving RCPSP through three major methods: precise (or deterministic), heuristic, and materialistic. Both of them considered the limit of the number of resources, however, the environment in which the resources are located is rarely mentioned. In recent years, more and more projects are conducted in a hazardous environment. Hazardous environment causes a multitude of occupational health and safety issues, which could lead to serious injury or even death. Although many robots used in the hazardous environment have been developed, there is still a lot of work to be taken by manpower. Workers cannot work in the hazardous environment for a long time for the sake of irreversible damage. The project of a nuclear power plant (NPP) overhaul and the project of nuclear accident emergency response are illustrative examples. The constraint of working time of employees must be considered to ensure the safety of the workers when we conduct these projects. Employees cannot continuously work for too long time under the hazardous environment. In addition, environmental radiation safety must be considered when staff scheduling in the use of radioactive isotopes for NDT (Non-destructive testing), in order to work effectively and safely; the spallation neutron source construction projects must consider the radiation environment to effectively schedule the construction activities. Thus, the RCPSP in a hazardous environment has strong practical significance and value.

Generally speaking, a RCPSP has two kinds of restrictions, precedence relationships and resource constraints. For each activity, the requests of resource, the duration and precedence relations with other activities are all assumed to be deterministic and known in advance. Garey and Johnson (1975) showed that the RCPSP with a single resource is NP-complete problem in the strong sense by reduction from the 3-partition problem. RCPSP is one of the most crucial problems in project scheduling problem and it has been established many theoretical models by lots of scholars (Brucker et al., 1999). The researchers (Demeulemeester & Herroelen, 2002a; Herroelen, De Reyck, & Demeulemeester, 1998; Kelley, 1961) usually devoted to find a schedule of optimal objective.
developed. In the RCPSP model, two kinds of renewable resources consist of labour, machine equipment, and field. Demeulemeester and Herroelen (2002b) developed a model that the objective is minimal total resource cost. Herroelen (2005) solved a RCPSP model in order to maximize the project’s net present value. In recent years, some scholars focus on the multi-objective problem (Viana & De Sousa, 2000) and the multi-executive mode problem (Koliseh & Hartmann, 2006). Most of these studies thought the resource as the most important constraint, however, the environment in which the resources worked is seldom considered in these articles.

Hazardous environment is concerned in various fields of scheduling. Li and Zhao (2007) established the collaborative and dynamic location optimization model of storage network in a hazardous environment. Based on the radiation environment of the nuclear power plant, Fourcade and Johnson established a mixed-integer programming model optimizing plant refuelling (Fourcade, Johnson, Bara, & Cortey-Dumont, 1997). Gorge focus on stochastic scheduling of NPP outages (Gorge & Zorgati, 2012). Khemmoudj, Porcheron and Bennaceur (2006) present a new approach for modelling and solving the problem of scheduling of 58 nuclear reactors outages. In the bulk of researches, collective dose is taken as the main concern, however, the injury suffered by the specific worker is equally important. When a project is executed in a hazardous environment, we must keep the hurt under the upper limit for every human body by all kinds of methods.

In this paper, a RCPSP model in the hazardous environment was established, with the objective of minimizing the project duration and keeping the hurt suffered by each worker under the controllable limit. To solve the problem, a novel discrete particle swarm algorithm was developed. In the RCPSP model, two kinds of renewable resources are examined. Taking into account the effect of staffing strategies on the project, nine strategies combinations will be discussed in detail.

2. Problem description

NPP overhaul refuelling is a typical project in a hazardous environment. Administering the project implementation is a challenging task. Depending on statistics, 80% of the nuclear collective effective dose is from the maintenance refuelling. All kinds of events or accidents are most prone to happen in refuelling overhaul project. In literature, many researchers focus on radiation protection and dose reduction of the NPP. Under the scenario of normal operation, Nguyen and Do (2015) concentrated in assessing radiation dose caused by radioactive substances rejected by a NPP using software package NRC-Dose72 provided by the USNRC (United States Nuclear Regulatory Commission). Wakker and Verhagen (2003) used the method of optimizing core design and shuffling sequence to reduce refuelling outage duration. Ke, Wang, Zeng, and Liu (2015) proposed the adjustment of radiation zoning in the reactor building during the refuel outage, based on the analysis of the change of radiation level in Qinshan NPP of China. These studies concentrate on the environment monitor and the ordinary task schedule.

The RCPSP in a hazardous environment can be stated as follows (Mejia et al., 2017; Trautmann, 2017; Villafañez et al., 2018). A project consists of a set \( N = \{0, 1, \ldots, J + 1\} \) of activities, where activity 0 and activity \( J + 1 \) are two dummy activities which represent the events of the start time and the finish time of the project, respectively. The precedence relationships between activities must be specified in the project and each activity has to be processed without interruption during implementing. The duration of activity \( j \) is denoted \( d_j \), and the quantity of resource type of \( k \) requested by activity \( j \) is denoted \( r_{jk} \). For the dummy activities, for any resource type \( k \), we have \( d_0 = d_{J+1} = 0 \), and \( r_{0k} = r_{J+1,k} = 0 \). The availability of each unit of resource type of \( k \) is \( R_k \) (\( k = 1, \ldots, K \)). \( W_k \) is the dangerous quantity which is undertaken by the resource engaged in the activity \( j \) during time period \( t \). The dangerous quantity of each resource, which is denoted \( U_{jit} \), should be accumulated according to Equation (4) given as follows. If the resource type is \( k \), then the accumulated dangerous quantity suffered by the resource must not exceed \( W_k \). A project schedule can be represented by the start time of each activity denoted \( (s_0, s_1, \ldots, s_{J+1}) \) with \( s_0 = 0 \) and the finish time of each activity denoted by \( (f_0, f_1, \ldots f_{J+1}) \) with \( f_0 = 0 \). A schedule is feasible if it satisfies both precedence relationships and all these constraints at any execution time period. The objective of the RCPSP problem in hazardous environment is to determine a feasible schedule with minimum project duration. The notations and variables are shown as follows.

Notations:

| \( J \) | Maximal number of project activities |
| \( K \) | Maximal number of resource type |
| \( j \) | Activity index, \( j = 0,1, \ldots, J + 1 \) |
| \( k \) | Resource type index, \( k = 1,2, \ldots, K \) |
| \( i \) | Resource index, \( i = 1,2, \ldots, R_k \) |
| \( (k,i) \) | Specific resource which type is \( k \) and index is \( i \) |
| \( t \) | Time period, \( t = 1,2 \ldots \infty \) |
Variables:

\[
x_{jkit} = \begin{cases} 
1, & \text{the resource } i \text{ of type } k \text{ is engaged in activity } j \text{ during time period } t \\
0, & \text{otherwise} 
\end{cases} 
\]

\[
y_{jt} = \begin{cases} 
1, & \text{activity } j \text{ is being executed during time period } t \\
0, & \text{otherwise} 
\end{cases} 
\]

\[
U_{kit} \quad \text{In time period } t, \text{ accumulated dangerous quantity of the resource which index is } i \text{ in the type of } k 
\]

\[
s_j \quad \text{Start time of activity } j 
\]

\[
f_j \quad \text{Finish time of activity } j 
\]

\[
w_{jt} \quad \text{Dangerous quantity which is undertaken by the resource engaged in the activity } j \text{ during time period } t 
\]

The mathematical model of the problem can be formulated as follows.

\[
\min f_{j+1}, \quad (1) 
\]

s.t.:

\[
f_j - s_j \geq d_j, \quad \forall j; \forall i \in P_j, \quad (2) 
\]

\[
\sum_{j=0}^{j+1} x_{jkit} \leq 1, \quad \forall k, i, t, \quad (3) 
\]

\[
R_k \sum_{j=0}^\infty x_{jkit} y_{jt} = r_{jk} y_{jt}, \quad (4) 
\]

\[
f_j - s_j = d_j, \quad \forall j, \quad (5) 
\]

\[
\sum_{t=s_j}^{f_j} x_{jkit} = d_j \left| \sum_{t=s_j}^{f_j} x_{jkit} = 0, \quad (6) 
\]

\[
\sum_{j=0}^{J+1} r_{jk} y_{jt} \leq R_k, \quad \forall k, t, \quad (7) 
\]

\[
U_{kit} = \sum_{j=0}^{J+1} w_{jt} x_{jkit} + (1 - C_k) U_{kiz-1}, \quad (8) 
\]

\[
U_{k0} = 0, \quad t = 1, 2, \cdots, \infty, \quad (9) 
\]

Equation (1) is the objective function to minimize the finish time of the last activity of the project, namely minimizing the project make span. Constraint (2) means that each task’s start time must be subject to the finish time of its preceding activities. It also can be expressed as activity \( j \) cannot start until all of the predecessor activities have finished, where \( P_j \) stands for the set of preceding activities of activity \( j \). Constraint (3) states that each resource can be engaged only by one activity at each period. Constraint (4) formulates that the quantity requirement of each resource type must be needed if activity \( j \) is being executed during the time period \( t \). Constraint (5) describes that the activity is non-interruptible. Constraint (6) ensures that the resource \((k,i)\) engaged in activity \( j \) must keep working until the finish of activity \( j \). Constraint (7) ensures that the total number of used resources of type \( k \), denoted \( R_k \), cannot exceed its available maximum. Constraint (8) describes the recursion of accumulative dangerous quantity suffered by the special resource, where \( C_k (C_k \in (0, 1)) \) is defined as the recovery coefficient of resource of the type \( k \). The recovering coefficient is used to indicate that the amount of damage received by the resource decreases over time. Constraint (9) limits the cumulative damage of a particular resource to no more than the maximum value. Constraint (8) and Constraint (9) emphasize the risk characteristics of the project environment, which is different from the traditional RCPSP.

We take a NPP outage project as the RCPSP in a hazardous environment in this paper. Because the outage project is having a high concentration of work activities and personnel in hazardous conditions, safety is the most fundamental and important issue when scheduling the project. The project needs many workers off and on, two kinds of key resources are discussed in this paper, the mechanical repairman and the radiation supervisor. In this paper, we take them as the constraints of scheduling with the objective of minimizing the project duration and keeping the personal dose at the reasonable level. According to the statistical data of the past years, we set the upper limit of the radiation to 8 mSv. The maximum number of repairmen and supervisors is 9. We assume that the initial values of all the cumulative radiation amount of the staff are 0. Because radiation attenuation is too long, \( C_k \), the recovery coefficients of \( k \) resources, are all set to 0. The other relevant information of the project is shown in Table 1.
Table 1. Information of the nuclear power plant outage project.

| Activity no. | Description of the activity                                      | Predecessor | Duration (h) | Dose (mSv/h) | Number of special resources |
|--------------|------------------------------------------------------------------|-------------|--------------|--------------|-----------------------------|
| 1            | Disassemble the generating unit and test the hot shunt down system | –           | 18           | 0.01         | 3, 2                         |
| 2            | Cool the reactor to residual heat removal process                | 1           | 25           | 0.02         | 2, 3                         |
| 3            | Clean the reactor to maintenance cold shutdown                   | 2           | 30           | 0.02         | 4, 3                         |
| 4            | Transport the reactor tool                                       | 3           | 13           | 0.03         | 3, 2                         |
| 5            | Drain reactor’s water to 10.5 m                                  | 3, 4, 5     | 10           | 0.05         | 2, 4                         |
| 6            | Remove reactor head and fill the reactor’s water to 19.5 m       | 4, 5, 6     | 36           | 0.05         | 2, 2                         |
| 7            | Maintain the main transformer                                    | 6           | 240          | 0.005        | 2, 3                         |
| 8            | Overhaul the main pump and motor                                 | 6           | 190          | 0.005        | 2, 2                         |
| 9            | Unlock the control rod drive package and hang out of the upper device | 6           | 7            | 0.05         | 2, 2                         |
| 10           | Overhaul the emergency generating set                            | 6, 11       | 108          | 0.01         | 3, 2                         |
| 11           | Prepare the reactor emergency generating set                     | 10          | 48           | 0.01         | 2, 4                         |
| 12           | Overhaul the generating set                                      | 11          | 108          | 0.01         | 5, 2                         |
| 13           | Send out opening tools and send in unloading tools                | 6, 16       | 16           | 0.05         | 2, 3                         |
| 14           | Handle the fuel assembly and test the storage system              | 9, 11       | 11           | 0.05         | 5, 4                         |
| 15           | Unload the fuel                                                  | 13, 14, 16  | 40           | 0.05         | 2, 2                         |
| 16           | Drain the reactor’s water to flange face                         | 15          | 7            | 0.01         | 3, 5                         |
| 17           | Overhaul the pressure vessel                                     | 16          | 75           | 0.11         | 3, 3                         |
| 18           | Do the charge-discharge experiments and overhaul the dc power distribution unit | 16, 17, 18  | 56           | 0.02         | 2, 3                         |
| 19           | Fill the reactor’s water to 19.5 m                               | 17, 18, 24  | 24           | 0.05         | 3, 2                         |
| 20           | Load fuel                                                        | 19          | 56           | 0.03         | 3, 3                         |
| 21           | Drain reactor’s water to 10.5 m                                  | 7, 12, 20   | 17           | 0.01         | 2, 4                         |
| 22           | Assembly the reactor lid                                         | 8, 21       | 26           | 0.02         | 2, 2                         |
| 23           | Fill water to Reactor and exhaust                                | 22          | 24           | 0.02         | 3, 2                         |
| 24           | Transit the reactor to hot shutdown                               | 23          | 37           | 0.01         | 3, 3                         |
| 25           | Reappraise the auxiliary feed water system                        | 24          | 8            | 0.01         | 2, 2                         |
| 26           | Test the digital system                                          | 24          | 6            | 0.01         | 2, 2                         |
| 27           | Test the generator set with load                                  | 25          | 12           | 0.01         | 3, 3                         |
| 28           | Test the telecom system                                          | 26, 27      | 12           | 0.01         | 3, 3                         |
| 29           | Escalate the power and combine to the grid                        | 28          | 66           | 0.02         | 3, 3                         |

To solve this schedule problem in the radiation environment, not only the availability of resources but also the radiation dose of each worker should be considered. We must maintain the radiation dose within the limit to assure the health of the workers. Because we must consider every worker’s tolerance, the number of constraints increases with the quantity of the resources, and this increase complexity of the investigated problem. We develop a novel Probability Mechanism Based on Particle Swarm Optimization algorithm (PMPSO) to solve the problem.

3. Design of the algorithm

Since the PSO algorithm was proposed, it has received extensive attention and continuous improvement from many scholars. PSO algorithm has made remarkable achievements in solving problems in the continuous domain. In order to apply PSO algorithm to the investigated problem, we propose a novel and efficient discrete PSO algorithm-probability mechanism-based particle swarm optimization (PMPSO). The mechanism of the PMPSO algorithm is described in the following sections.

3.1. The representation of particles

The representation of particles is one of the most challenging issues when applying the PSO algorithm to the discrete domain. In other words, establishing a correspondence between the particles and the problem to be solved is a key process. This process can be called particle coding. To solve the problem in this article, we used priority-based encoding rules. In this method, the sequence position of task \( j \) represents its priority. When resource contention occurs, tasks with high priority are first scheduled. Then, the project scheduling problem in this paper can be regarded as sequencing of the activities. For example, we assume that a project consists of five activities which represented by the number of 1, 2, 3, 4 and 5, then the scheduling problem can be considered as sequencing of the five numbers. In some cases, a permutation, e.g. (2, 3, 1, 4, 5), may be more suitable for the objective of the project scheduling rather than another permutation, e.g. (5, 4, 3, 2, 1). The scheme is consists of some basic subarrays, for example, the permutation of (1, 2, 3, 4, 5) is formed by the fundamental subarrays of (1, 2), (2, 3), (3, 4) and (4, 5). The permutation can be described by a two-dimensional array called adjacent matrix, in which the elements correspond to the
son-permutations exist in the permutation are set to 1. The other elements are set to 0. Here are two examples of adjacent matrixes representing permutations: (1, 2, 3, 4, 5) can be represented by \[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\] and (2, 3, 1, 4, 5) can be represented by \[
\begin{pmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}.
\]
Hence, the adjacency matrix is a \((0, 1)\)-matrix with zeros on its diagonal, where the element \(a_{ij} = 1\) states that the activity ‘i’ is in front of the activity ‘j’ and they are adjacent in the scheduling solution.

3.2. Optimization mechanism of son-permutation based on probability

A better scheduling scheme is different from the other because of the component of the son-permutations. In other words, the better scheme includes better son-permutations that the others do not include. The optimization of a scheduling scheme is determined by the merits of its son-permutations. In this paper, we use the probability selection method to generate a scheduling scheme based on the idea that an excellent son-permutations should have a high probability of being selected. We try to find these excellent son-permutations through the PSO algorithm and set higher probability values to them. Moreover, a scheme formed by means of probability selection can also break the local optimum and do not lose the opportunity to find a better solution. From the above, we propose the optimization mechanism of son-permutations based on probabilities.

We define the position of a particle denoted \(x_{id}\) as the adjacent \((0, 1)\)-matrix corresponding to the scheduling scheme. Where ‘i’ denotes the particle and ‘d’ denotes the dimension search space. The velocity of a particle denoted \(v_{id}\) is defined as a probability matrix. \(v_{id}\) can be updated through recursion operation with the Equation (10), through which the probabilities of the excellent son-permutations are strengthened.

\[
v_{id}(t + 1) = w \times v_{id}(t) \oplus c_1 \otimes r_1 \cdot (p_{gd} \rightarrow x_{id}(t)) \\
\oplus c_2 \otimes r_2 \cdot (p_{gd} \rightarrow x_{id}(t)),
\]
where \(x_{id}(t)\) is the current position of the particle, \(p_{id}\) is the best position of the particle experienced, \(p_{gd}\) is best position of the group experienced, and they are all \((0, 1)\)-matrixes. ‘\(\rightarrow\)’ denotes a generalized subtraction operator. The items \((p_{id} \rightarrow x_{id})\) and \((p_{gd} \rightarrow x_{id})\) are used to find the unique son-permutations in the schemes \(p_{gd}\) and \(p_{gd}\) but not in the individual \(x_{id}\). The elements of the operation result less than zero are set to zero. We give an example of the ‘\(\rightarrow\)’ operator. The \((0, 1)\)-matrix \[
\begin{pmatrix}
0 & 0.1386 & 0.0008 & 0.0002 & 0.0003 \\
0 & 0.0003 & 0.0007 & 0.0006 & 0.0007 \\
0.4984 & 0.0008 & 0.0010 & 0.0008 & 0.0001 \\
0.0007 & 0.0003 & 0.2551 & 0.0005 & 0.0003 \\
0.0002 & 0.0006 & 0.0007 & 0.2435
\end{pmatrix}
\]
is \(x_{id}\) representing the permutation \((1, 2, 3, 4, 5)\). The result of \((p_{gd} \rightarrow x_{id})\) is \[
\begin{pmatrix}
0 & 0.1386 & 0.0008 & 0.0002 & 0.0003 \\
0 & 0.0003 & 0.0007 & 0.0006 & 0.0007 \\
0.4984 & 0.0008 & 0.0010 & 0.0008 & 0.0001 \\
0.0007 & 0.0003 & 0.2551 & 0.0005 & 0.0003 \\
0.0002 & 0.0006 & 0.0007 & 0.2435
\end{pmatrix}.
\]
This result denotes that the son-permutations \((3, 1)\) and \((1, 4)\) are unique for the \(p_{gd}\). The \(r_1\) and \(r_2\) are the random matrixes whose elements are in the interval \((0, 1)\). \(c_1\) is a positive constant called coefficient of the self-recognition component and \(c_2\) is a positive constant called coefficient of the social component. The variable \(w\) is called the inertia factor, whose value is typically set to vary linearly from 1 to near 0 during the iteration process. Operator \(\oplus\) is defined as element-wise generalized multiplication and operator \(\otimes\) is generalized addition. After the operation with the generalized addition operator \(\oplus\), the elements of calculated result are set to one if they exceed one. From Equation (10), a particle decides where to move next, considering its current state, its own best experience and the experience of its most successful particle in the swarm.

When we update \(x_{id}\) to \(x_{id}(t + 1)\), an intermediate variable \(g_{x_{id}}\) is used. The \(g_{x_{id}}\) can be obtained according equation (11).

\[
g_{x_{id}}(t) = g(x_{id}(t)),
\]
where \(g_{x_{id}}\) is defined as a probability matrix with zero elements on its diagonal. \(g_{x_{id}}\) can be calculated from \(x_{id}\) according to the ‘\(g\)’ algorithm. The \(g\) algorithm can be free chosen following the rule that make sure the elements in \(g_{x_{id}}\) correspond to the elements which are equal to one in the \(x_{id}\) have big probability. For example, we assume that \(x_{id}(t)\) is \[
\begin{pmatrix}
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
then \(g_{x_{id}}(t)\) could be \[
\begin{pmatrix}
0 & 0.4984 & 0.0008 & 0.0010 & 0.0008 \\
0.0007 & 0.2551 & 0.0005 & 0.0003 & 0.0002 \\
0.0002 & 0.0006 & 0.0007 & 0.2435 & 0.0001 \\
0.0001 & 0.0002 & 0.0009 & 0.0003 & 0
\end{pmatrix}.
\]

We firstly update \(g_{x_{id}}\) according to Equation (8), then we can update \(x_{id}\) to \(x_{id}(t + 1)\) according to Equation (12).

\[
g_{x_{id}}(t + 1) = g_{x_{id}} \oplus v_{id}(t + 1).
\]
Here the probability sum rule must be obeyed. We assume that \(v_{id}(t + 1)\) is \[
\begin{pmatrix}
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000
\end{pmatrix},
\]
then \(g_{x_{id}}(t + 1) =
\begin{pmatrix}
0 & 0.4984 & 0.0008 & 0.7010 & 0.0008 \\
0.0007 & 0.2551 & 0.0005 & 0.0003 & 0.0002 \\
0.0001 & 0.0002 & 0.0006 & 0.0007 & 0.2435 \\
0.0001 & 0.0002 & 0.0009 & 0.0003 & 0
\end{pmatrix}.
\]

Then, we use Equation (13) to generate the 0–1 matrix of \(x_{id}(t + 1)\).

\[
x_{id}(t + 1) = f(g_{x_{id}}(t + 1)),
\]

where \( f(gx_{id}(t+1)) \) was defined as probability selecting operation, which select the right elements and set them to 1 in \( x_{id}(t+1) \) according to the given probabilities in \( gx_{id}(t+1) \). Then we can have the new adjacent matrix and the new scheduling scheme is obtained by decoding it.

The following is the process to generate the adjacent \((0, 1)\)-matrix with obtained \( gx_{id}(t+1) \) above.

**Process: Adjacent (0, 1)-matrix generation process**

**Step 1.** Select the row where the biggest value of the matrix exists, e.g. \((0.8007, 0.0003, 0.0005, 0.1386, 0.0008)\).

**Step 2.** The elements in step 1 are used as weight coefficients, and the column numbers are randomly selected based on them. Assume that the first column is selected.

**Step 3.** Let the corresponding element in \( x_{id}(t+1) \) equal to 1 and set the contradictory elements in \( gx_{id}(t+1) \) to zero. Then \( x_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\)
and \( gx_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0.4984 & 0.0008 & 0.7010 & 0.0008 \\
0 & 0 & 0.2551 & 0.0006 & 0.0003 \\
0 & 0.0006 & 0.0007 & 0.2435 & 0 \\
0 & 0.0002 & 0.0009 & 0.0003 & 0
\end{pmatrix}
\).

**Step 4.** The first row \((0, 0.4984, 0, 0.7010, 0.0008)\) is selected and the elements are used as weights to randomly select a row. Assume that the fourth column is selected. Then \( x_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\)
and \( gx_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\).

**Step 5.** The second row \((0, 0, 0.2551, 0, 0.0003)\) is selected and the elements are used as weights to randomly select a column. Assume that the third column is selected. Then \( x_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\)
and \( gx_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\).

**Step 6.** The third row \((0, 0.0006, 0, 0, 0.2435)\) is selected and the elements are used as weights to randomly select a column. Assume that the fifth column is selected. Then \( x_{id}(t+1) \) is \(
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\)
and the 0-1 adjacent matrix is finish and the new permutation is \((2, 3, 1, 4, 5)\).

The logic diagram of \((0, 1)\)-matrix generation operation is showed as Figure 1.

4. Solution and discussions

About the RCPSP in a hazardous environment, it is hard to be solved on account of coupling of many factors. We
develop the PMPSO algorithm to solve it and get good results. The logic diagram of solving process is showed as Figure 2. Except for the algorithm, scheduling the project according to a given sequence is also a difficult point because the coupling of resources requirements, availability and the dangerous quantity accumulated. Individual task usually cannot be interrupted, so the use of the resources is continued throughout the task processing. All of the above make the project scheduling according to a given sequence to become complex. In addition, the project’s environment is hazardous, then, the staffing strategy is important. Different strategies make resources suffer different harm.

4.1. Staffing strategies and scheduling

When we assign resources to an activity, the rule of selecting resources, which we call it staffing strategy, is necessary. Generally, we can construction many kinds of strategies according many methods of selecting resources. In this paper, three basic strategies are considered:

Strategy 1: Use resources as much as possible. In this strategy, we allocate resources to an activity in the order of its index. Only when the resources are unavailable, we allocate the next resource to the activity in the order.

Table 2. Combinations of strategies.

| Combination 1 | Combination 2 | Combination 3 | Combination 4 | Combination 5 | Combination 6 | Combination 7 | Combination 8 | Combination 9 |
|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Repairman     | Strategy 1    | Strategy 2    | Strategy 3    | Strategy 1    | Strategy 2    | Strategy 3    | Strategy 1    | Strategy 2    |
| Supervisor    | Strategy 1    | Strategy 2    | Strategy 3    | Strategy 2    | Strategy 3    | Strategy 1    | Strategy 2    | Strategy 3    |

Figure 3. Logic diagram of solving process of project duration.

Table 3. Experiments results of the algorithm performance.

| Instance     | Results in PSPLIB | Results by PMPSO |
|--------------|-------------------|------------------|
|              | Best known solution | Literature      | Best solution | Worst solution | Average of solutions | Deviation (%) |
| J301-1       | 43                | Brucker et al. (1999) | 43            | /              | 43                | 0             |
| J302-4       | 43                | Brucker et al. (1999) | 43            | /              | 43                | 0             |
| J303-9       | 65                | Brucker et al. (1999) | 65            | /              | 65                | 0             |
| J305-10      | 70                | Herroelen et al. (1998) | 70            | 72             | 70.04             | 0.06          |
| J308-1       | 44                | Brucker et al. (1999) | 44            | /              | 44                | 0             |
| J3010-4      | 58                | Brucker et al. (1999) | 58            | /              | 58                | 0             |
| J3015-2      | 47                | Brucker et al. (1999) | 47            | 48             | 47                | 0             |
| J3026-6      | 53                | Demeulemeester and Herroelen (2002a) | 53           | /              | 53                | 0             |
| J3030-1      | 47                | Kelley (1963)       | 47            | 49             | 47.08             | 0.17          |
| J3040-8      | 57                | Slowinski (1980)    | 57            | /              | 57                | 0             |
Strategy 2: Use resources randomly. In this strategy, we assign a resource randomly. If the resource is not available, we randomly select another resource.

Strategy 3: Use a resource firstly when its accumulated dangerous quantity is smallest. In this strategy, we first allocate resources that have the least cumulative dangerous quantity to activities.

In the project we studied, we considered two key resources, so when planning a project, there are three strategies for each resource. Therefore, we have 9 combinations of strategies, as showed in Table 2.

When we scheduling the investigated project according to feasible sequence, we adopt these combinations of strategies, respectively.

The logic diagram of project scheduling according feasible sequence is shown in Figure 3.

4.2. Computational results and discussions

First we conducted some experiments to verify the algorithm performance. The test set comes from J30
of the PSPLIB (Project Scheduling Problems Library), provided on the website (http://129.187.106.231/psplib). The currently best known solution in PSPLIB as reported on 17 June 2015. For the J30 set, these solutions are all optimal. We set the size of the particle swarm to 20, the largest number of iterations to 500, c1 to 2, c2 to 1.02 and w to 0.95. We run the program 20 times for each case. The statistical results of the experiments are shown as the ‘Results by PMPSO’ component in Table 3, where Deviation is defined as (Average of solutions-best known solution)/ best known solution.
Next, we use the PMPSO algorithm to solve the project scheduling problem under the dangerous environment studied in this paper. We use the same parameter settings as when solving the J30 set. The program terminates and outputs the result when no optimal value is updated for 500 consecutive iterations.

We schedule the project according to the strategies combine 1, which is illustrated in Table 2. The results are showed in Figures 4–7. Figure 4 is the evolution curve of the algorithm; Figure 5 is the optimal schedule, which form is the Gantt chart; Figure 6 is the diagram of resources use and Figure 7 is the total radiation dose of each resource.

From Figure 4, we note that the evolution curve drops sharply at first. This shows that the PMPSO algorithm has advantage of rapid convergence being the most important advantage of original PSO algorithm. Before the curve levels off to the minimal value, it experiences several platforms. This shows that the PMPSO algorithm can effectively avoid converging on local optimum and remarkably reduce solution time. Figure 5 shows the Gantt chart of the optimal schedule. The duration of the project is 500 hours and the length of the critical path is 490 hours, which states that the final duration of the project is comparatively good. However, from Figure 6, we observed that resources usage exits imperfection. The use of some resources is ineffective and inefficient. Figure 7 shows the total radiation dose of each resource of each kind. Due to the staffing strategies combination, only after the resources which number is smaller are all not available, can a resource be assigned to a task. This leads to the total radiation dose of these resources whose number is smaller is larger, which is showed in Figure 7.

We conduct the programme adopting the strategies combination 2, which is illustrated in Table 2. The final results are shown in Figures 8–10.

In Figure 8, the evolution curve is similar to Figure 4, and the curve also converges when the project completion time is 500. This shows that the strategy of randomly allocating resources does not substantially affect the duration of the project. However, the resources usage and the total radiation dose of each resource are different from combination 1, which are illustrated in Figures 9 and 10 separately. Due to the random assignment, the usage and radiation dose are all random distributions.

We employ the combination 3 to conduct the programme and Figures 11–13 show the results. From Figure 11, we observed that the evolution curve eventually levels off to 500 too, and yet it experiences fewer iterations. Figure 12 shows that the optimization of resource levelling is preferable to the other two combinations of strategies. A resource is selected to be assigned to a task only if its radiation dose is smallest, thus balancing the

From Table 3 we can see that for the instances we chose in the study, we can use the PMPSO algorithm to find the optimal solution. Thus, the PMPSO algorithm proposed in this paper is suitable for solving the RCPSP.
Table 4. Results of 9 combinations (Dose range = Max dose — Min dose).

| Items                        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|------------------------------|----|----|----|----|----|----|----|----|----|
| Project duration (h)         | 500| 500| 500| 512| 502| 512| 502| 512| 502|
| Max dose of repairmen (mSv)  | 7.79| 7.95| 5.95 | 8 | 7.69| 6.45| 7.78| 7.81| 6.14|
| Min dose of repairmen (mSv)  | 0.37| 3.27| 4.81| 0.48| 3.68| 4.91| 1.12| 3.08| 4.84|
| Dose range of repairmen (mSv)| 7.42| 4.68| 1.14| 7.52| 4.01| 1.54| 6.66| 4.73| 1.3 |
| Max dose of supervisors (mSv)| 7.16| 6.7 | 5.5 | 7.19| 5.25| 7.98| 5.19| 7.92| 7.66|
| Min dose of supervisors (mSv)| 2.04| 2.6 | 4.26| 2.25| 4.05| 1.27| 4.11| 0.07| 2.44|
| Dose range of supervisors (mSv)| 5.12| 4.1 | 1.24| 4.94| 1.2 | 6.71| 1.08| 7.85| 5.22|

Table 5. Duration of each activity based on three-point estimating method.

| Activity | $t_o$ | $t_m$ | $t_p$ | Activity | $t_o$ | $t_m$ | $t_p$ |
|----------|-------|-------|-------|----------|-------|-------|-------|
| 1        | 15    | 18    | 25    | 16       | 5     | 7     | 12    |
| 2        | 20    | 25    | 35    | 17       | 68    | 75    | 85    |
| 3        | 25    | 30    | 50    | 18       | 45    | 56    | 76    |
| 4        | 10    | 13    | 20    | 19       | 16    | 24    | 34    |
| 5        | 7     | 10    | 15    | 20       | 45    | 56    | 69    |
| 6        | 30    | 36    | 56    | 21       | 12    | 17    | 25    |
| 7        | 210   | 240   | 300   | 22       | 20    | 26    | 36    |
| 8        | 150   | 190   | 290   | 23       | 19    | 24    | 32    |
| 9        | 5     | 7     | 15    | 24       | 30    | 37    | 57    |
| 10       | 90    | 108   | 120   | 25       | 6     | 8     | 12    |
| 11       | 40    | 48    | 78    | 26       | 5     | 6     | 10    |
| 12       | 96    | 108   | 120   | 27       | 10    | 12    | 18    |
| 13       | 12    | 16    | 23    | 28       | 9     | 12    | 16    |
| 14       | 9     | 11    | 15    | 29       | 60    | 66    | 79    |
| 15       | 30    | 40    | 55    |          |       |       |       |

use of resources and the radiation dose. From Figure 13, we observed that the dose of each resource is not much different and each dose is under 6mSv, which is an absolutely safe dose of radiation.

We conduct the programme according to 9 combinations, respectively. The statics results are shown in Table 4. From Table 4, we observed that the combination 3 outperforms the other combinations in almost all of the indicators. It also shows that the first three combinations are better than the others in the indicator of duration.

4.3. Discussion on the adaptability of the scheduling schemes

In the project case showed in Table 1, the task duration is given by a single time estimate. However, the exact duration in reality often varies with the estimating. Thus, the better scheme calculated above is in doubt when the difference exists. Thus, the scheduling scheme calculated above based on PMPSO compares with the random scheduling scheme to examine the performance of it in our numerical experiments.

During the numerical experiment, for the project case showed in Table 1, we use three estimates to define an approximate range for every activity’s duration of the project: most likely ($t_m$), optimistic ($t_o$) and pessimistic ($t_p$). We assume that every activity’s duration is subjected
to beta distribution with the parameters of the three estimates mentioned above. Table 5 shows the results of the three-point assessment method for the duration of each activity.

According to the time estimates in Table 5, we generate 100 instances for the problem. For each instance, the duration of each activity is randomly generated according to its parameters of beta distribution. Then, for each instance, we randomly generate a scheduling priority sequence and schedule the project under the condition of satisfying all constraints to the project. At the same time, we use the optimal scheduling scheme to schedule the project of each instance. The optimal scheduling Priority sequence is (1, 2, 4, 7, 8, 3, 6, 10, 13, 14, 5, 9, 15, 20, 11, 12, 19, 25, 16, 17, 18, 21, 22, 24, 27, 23, 26, 28, 29). The results are shown in Figures 14 and 15. In the above experimental schemes, the personnel use strategy is based on the strategies combination 3.

It can be observed from the Figure 14 that the optimal scheduling scheme is always superior to randomly determine scheduling scheme in duration indicator. With the random fluctuation of the duration of the project task, the duration of the project scheduling according to the optimal scheduling scheme fluctuates less. Figure 15 shows that the max dose of all kinds of resources varies with the instances. We observed that the optimal scheduling scheme is almost superior to randomly determine scheduling scheme for all instances. Thus, the scheduling scheme in Section 4.1 has good adaptability.

Based on the optimal scheduling priority sequence obtained in Section 4.2, we compare the strategy combination 3 with the other 8 different combinations. For each comparison, we randomly generate 100 different project instances based on the time estimates in Table 5. Different combinations of strategies have little effect on the duration of the project, but they have a great impact on the maximum amount of radiation that the workers are exposed to. Figure 16 shows the comparison results of the maximum radiation dose of workers for the combination of strategy 3 and other combinations.

From Figure 16, we can see that using the strategy combination 3 can effectively reduce the maximum amount of radiation that workers can bear when forming a final dispatching plan, and ensure the physical safety of workers.

5. Conclusions and future work

RCPSP is a NP-hard problem. In this study, the hazardous environment received special attention, which increases the complexity of the problem. The discrete PSO algorithm based on probability optimization mechanism is designed to solve the problem efficiently. We schedule the project respectively according to unique combination of staffing strategies. A comparison of combinations of staffing strategies shows that Strategy 3 outperforms the other strategies. The numerical experiments prove that the solution still maintains better performance compared with the random scheduling even though project activities’ duration fluctuates. Therefore, in a hazardous environment, adopting PMPSO algorithm to solve project scheduling problems can effectively schedule project tasks and control extent of damage of the resources. The
scheduling solution not only helps to ensure the implementation of the project, but also guarantees the physical and mental health of workers.

In general, the theoretical contribution of this paper has two points: First, the RCPSP problem is extended to a dangerous environment and a corresponding problem.
model is established; secondly, a corresponding solution algorithm PMPSO is proposed and applied to successfully solve the problem. In addition, the experiment proves that the strategy combination 3 is better than other strategy combinations. This result can provide guidance for the staffing strategy in a dangerous environment, which has a general practical significance.

For future work, we intend to concentrate on non-deterministic project scheduling in a hazardous environment. In a hazardous environment with exponential probability distribution or Gaussian probability distribution (Wei, Qiu, & Fu, 2015; Wei, Qiu, Karimi, & Quantized, 2015), the project’s activity duration is stochastic. However, The PMPSO algorithm will still have a strong adaptability. Because the distribution of duration does not affect the performance of the algorithm. We can establish a new objective function based on the mean square framework, then, the PMPSO algorithm can be used to obtain a robust schedule scheme. Sliding mode technique has a strong robustness for uncertain parts of the systems (Wang, Shen et al., 2017; Wang, Gao et al., 2017b; Wang, Xia, Shen, & Zhou, 2018). In view of this, we can combine the SMC technique to improve the robustness of the algorithm in the future.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Natural Science Foundation of China [61703102]; Social Science and Technology Development Program of Dongguan City [2014106101014]; Science and Technology Planning Project of Guangdong Province, China [2015A010103021]; Education Scientific Planning Project of Guangdong Province, China [14JKX051]; Natural Science Foundation of Guangdong Province, China [2015A030310274,2015A030313649]; Key Platforms and Scientific Research Program of Guangdong Province, China [No.2015KTSCX137]; Institutions of Higher Learning Innovation Strong School of Project of Guangdong Province, China [2015GJXK144].

ORCID

Shuai Li http://orcid.org/0000-0003-0558-6795

References

Brucker, Peter, Drexl, Andreas, Mohring, Rolf, Neumann, Klaus, & Pesch, Erwin. (1999). Resource-constrained project scheduling: Notation, classification, models and methods. European Journal of Operational Research, 112(1), 3–41.

Demeulemeester, E., & Herroelen, W. (2002a). Project scheduling: A research handbook. Boston, MA: Kluwer Academic.

Demeulemeester, E., & Herroelen, W. (2002b). Project scheduling: A research handbook. New York, NY: Kluwer Academic.

Fourcade, F., Johnson, E., Bara, M., & Cortey-Dumont, P. (1997). Optimizing nuclear power plant refueling with mixed-integer programming. European Journal of Operational Research, 97(2), 269–280.

Garey, M. R., & Johnson, D. S. (1975). Complexity results for multiprocessor scheduling under resource constraints. SIAM Journal on Computing, 4, 397–411.

Gorge, A. L., & Zorgati, R. (2012). Stochastic nuclear outages semidefinite relaxations. Computational Management Science, 9(3), 363–379.

Herroelen, W. S. (2005). Project scheduling-theory and practice. Production and Operations Management, 14(4), 413–432.

Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource-constrained project scheduling: A survey of recent developments. Computers and Operations Research, 25(4), 279–302.

Ke, H., Wang, C., Zeng, J., & Liu, J. (2015). Study on radiation zone adjustment of reactor plant during refueling outage in Qinshan phase II NPP. Nuclear Power Engineering, 36(S1), 75–77.

Kelley, J. E. (1961). Critical-path planning and scheduling: Mathematical basis. Operations Research, 9(3), 296–320.

Kelley, J. E. (1963). The critical path method: Resources planning and scheduling in industrial scheduling (pp. 347–365). New Jersey: Prentice Hall.

Kemmoudj, M., Porcheron, M., & Bennaceur, H. (2006). When constraint programming and local search solve the scheduling problem of electricité de france nuclear power plant outages[C]. International Conference on Principles and Practice of Constraint Programming, Springer, Berlin, Heidelberg, 271–283.

Koliseh, R., & Hartmann, S. (2006). Experimental in ventilation of heuristics for resource constrained project scheduling: An update. European Journal of Operational Research, 174(1), 23–37.

Li, S., & Zhao, L. (2007). Collaborative allocation problem of rescue supplies under danger source diffusion. Journal of Southeast University: Natural Science Edition, 37(1), 367–373.

Mejia, O. P., Anselmet, M. C., Artigues, C., & Lopez, Pierre. (2017). A new RCPA new RCPSP variant for scheduling research activities in a nuclear laboratory[C]/47th International Conference on Computers & Industrial Engineering (CIE47), Lisbon, 8p.

Nguyen, T. K., & Do, X. A. (2015). Assessment of radiation dose caused by radioactive gaseous effluent released from nuclear power plant Ninh Thuan 1 under scenario of normal working conditions. World Journal of Nuclear Science and Technology, 5, 183–191.

Slowinski, R. (1980). Two approaches to the problem of resource constrained project scheduling. European Journal of Operational Research, 25(4), 413–432.

TRAUtmann, N. (2017). A mixed-integer linear programming formulation of the RCPSP based on explicit assignment and sequencing decisions[C]. INFORMS Annual Meeting, Houston, 22–25.

Viana, A., & De Sousa, J. P. (2000). Using metaheuristics in multi objective resource constrained project scheduling. European Journal of Operational Research, 120(2), 359–374.

Villafáñez, F. A., Poza, D., López-Paredes, A., & Pajares, J. (2018). A unified nomenclature for project scheduling problems (RCPSP and RCMPSP). Dirección y Organización, 64, 56–60.

Wakker, P. H., & Verhagen, F. C. M. (2003). Reducing refueling outage duration by optimizing core design and shuffling
sequence. Advances in nuclear fuel management III (ANFM), Hilton Head Island, 16p.

Wang, Y., Gao, Y., Karimi, H. R., Shen, Hao, & Fang, Zhijun. (2017). Sliding mode control of fuzzy singularly perturbed systems with application to electric circuit. *IEEE Transactions on Systems Man, and Cybernetics: Systems*, PP(99), 1–8.

Wang, Y., Shen, H., Karimi, H. R., & Duan, Dengping. (2017). Dissipativity-based fuzzy integral sliding mode control of continuous-time TS fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 26(3), 1164–1176.

Wei, Y., Qiu, J., & Fu, S. (2015). Mode-dependent nonrational output feedback control for continuous-time semi-markovian jump systems with time-varying delay. *Nonlinear Analysis: Hybrid Systems*, 16, 52–71.

Wei, Y., Qiu, J., Karimi, H. R., & Quantized, H. (2015). Filtering for continuous-time Markovian jump systems with deficient mode information. *Asian Journal of Control*, 17(5), 1914–1923.

Wang, Y., Xia, Y., Shen, H., Zhou P. (2018). SMC design for robust stabilization of nonlinear Markovian jump singular systems. *IEEE Transactions on Automatic Control*, 63(1), 219–224.