Journal of Project Management 7 (2022) 95–110

Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

Review of conventional metaheuristic techniques for resource-constrained project scheduling problem

Amir Golab[a*], Ehsan Sedgh Gooya[a], Ayman Al Falou[a] and Mikael Cabon[a]

[a]*ISEN Yncréa Ouest, Brest, France*

CHRONICLE

Article history:
Received: September 10, 2021
Received in revised format: October 2, 2021
Accepted: October 21, 2021
Available online:
October 21, 2021

Keywords:
Project management
Project scheduling
Project duration
RCPSP
Meta-heuristics
Resource constraint

ABSTRACT

This paper is concerned with an overview of the Resource-Constrained Project Scheduling Problem (RCPSP) and the conventional meta-heuristic solution techniques that have attracted the attention of many researchers in the field. Therefore, researchers have developed algorithms and methods to solve the problem. This paper addresses the single-mode RCPSP where the objective is to optimize and minimize the project duration while the quantities of resources are constrained during the project execution. In this problem, resource constraints and precedence relationships between activities are known to be the most important constraints for project scheduling. In this context, the standard RCPSP is presented. Then, the classifications of the collected papers according to the year of publication and the different meta-heuristic approaches applied are presented. Five weighted articles and their meta-heuristic techniques developed for RCPSP are described in detail and their results are summarized in the corresponding tables. In addition, researchers have developed various conventional meta-heuristic algorithms such as genetic algorithms, particle swarm optimization, ant colony optimization, bee colony optimization, simulated annealing, evolutionary algorithms, and so on. It is stated that genetic algorithms are more popular among researchers than other meta-heuristics. For this reason, the various conventional meta-heuristics and their corresponding articles are also presented to give an overview of the conventional meta-heuristic optimizing techniques. Finally, the challenges of the conventional meta-heuristics are explored, which may be helpful for future studies to apply new suitable techniques to solve the Resource-Constrained Project Scheduling Problem (RCPSP).

1. Introduction

In the last few decades, the resource constrained project scheduling problem (RCPSP) and its solution techniques have been studied. RCPSP is a problem that focuses on optimizing and minimizing the total makespan of a project while resources are constrained. In this context, resource constraints and precedence relations between activities are known to be major constraints in project scheduling. Over the years, many researchers have used and developed various solution techniques classified as meta-heuristic techniques, exact methods, etc. (Ortiz-Pimiento & Diaz-Serna, 2018) to achieve a particular objective of the problem. Therefore, the number of published articles is considerable. Since RCPSP is known as NP-complete problem (lazewicz, Lenstra, & Kan, 1983), which is classified into the category of nondeterministic polynomial problems, the exact solution techniques such as the branch-and-bound method are not suitable to solve the large-scale problems due to the considerable computation time, so the exact methods are not effective enough for practical problems (Cho & Kim,

* Corresponding author.
E-mail address: amir.golab@isen-ouest.yncrea.fr (A. Golab)

doi: 10.5267/j.jpm.2021.10.002

1997) . On the other hand, a considerable number of articles have been published to develop conventional metaheuristic approaches such as Genetic Algorithms (GA), Ant-Colony Optimization (ACO), Simulated Annealing (SA), etc., which are more practical than exact techniques to achieve an optimal or near-optimal objective, especially for project scheduling with a large number of activities (Agarwal, Colak, & Erenguc, 2015; Koulinas, Kotsikas, & Anagnostopoulos, 2014).

An RCPSP is introduced by the set $A = \{1, \dots, J\}$ of activities bounded by two types of constraints, called precedence relations and resource constraints, respectively. (2) The first one means that activity j cannot be started until its immediate predecessors have been completed. In this case, a precedence feasible project schedule is achieved. On the other hand, there is a set of renewable resources $R = \{1, \dots, K\}$ during the execution of the activities of the project, i.e., each activity requires r_{jk} units per time for execution. Therefore, (3) the second respectable constraint of an RCPSP is to consider the available quantities of resources period by period in order to obtain a resource feasible project schedule (Kolisch & Hartmann, 2006; Bouleimen & Lecocq, 2003; Mendes, Gonçalves, & Resende, 2009). It is possible to obtain a feasible project schedule given the defined constraints. In short, (1) a standard RCPSP can be formulated to minimize the project duration considering the main constraints according to the following formula(Roy & Sen, 2019). $\boldsymbol{C}$ is defined as Critical Path Method (CPM).

$$min\ C \tag{1}$$

$$S_j\text{-}S_i \geq d_i \quad \forall (A_i, A_j) \in Pred \tag{2}$$

$$\textstyle\sum_{A_i \in A_t} r_{ik} \leq R_k \ , \forall R_k \in R \tag{3}$$

In the following lines the defined elements are presented:

- The set A represents the activities constituting the project with duration d_j and $j = 1, 2, 3, \dots, n$.

Note: If the dummy activities of start and end with duration of 0 are added, then $j = 0, 1, 2, 3, \dots, n, n + 1$.

- The set of R represents the renewable resources and $k = 1, 2, 3, \dots, r$ also, R_k represents the available quantities of renewable resource r
- S_j represents the start time of activity j
- S_i represents the start time of activity i, which is the immediate predecessor of activity j.
- The set of Pred consisting of ordered pairs (A_i, A_j) shows that A_i is an immediate predecessor of A_j
- r_{jk} represents the amount of renewable resources consumed by activity j.

There are three types of schedules: the semi-active schedule, which is feasible as long as activities cannot be shifted locally to the left, the active schedule, which is feasible as long as no activities can be shifted locally or globally to the left (Sprecher, Kolisch, & Drexl, 1995) , and finally Non-delay schedule, which is also feasible if no resource is idle while the resource can start processing other activities (Mendes, Gonçalves, & Resende, 2009).

The paper provides an overview of published articles that address the problem of resource-constrained project planning. The rest of the paper is divided into five sections. The second section of the paper presents two types of classifications for published articles, and the third section summarizes the weighted articles. In the fourth section, we briefly explain the summary of traditional metaheuristic solution techniques and then mention the related articles. In the following, we investigate the challenges of metaheuristics in the fifth section. And the conclusion in the last section.

2. The classification of published RCPSP articles

This section focuses on the statistics of the published RCPSP articles classified by the years of publication and the metaheuristic solution techniques. The papers were collected from the qualified databases, and they include the journal articles (63%) and conference papers (37%).

2.1. Annual classification of published articles

Fig. 1 focuses on the annual classification of published articles titled by RCPSP. The graph shows that researchers are interested in developing metaheuristic solution techniques to achieve optimal or near optimal results.

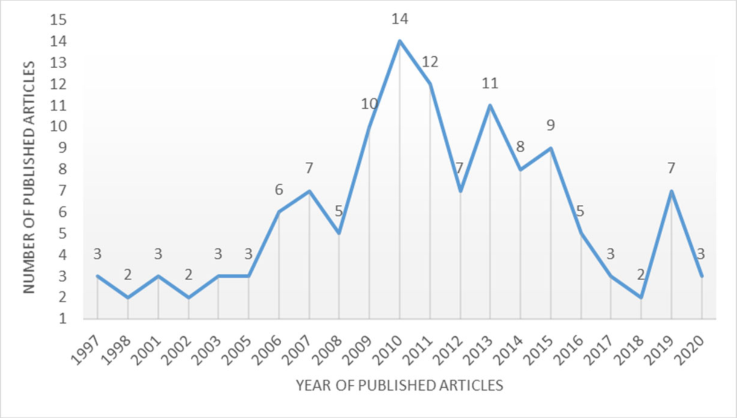

Fig. 1. Frequency of titled RCPSP articles. It is obvious that researchers worked on the problem more in 2010 and 2011 than in other years, but the trend shows that it does not stop.

2.2. Classification of published articles according to usual meta-heuristic techniques

Fig. 2 presents the frequency of published articles with the topic RCPSP. The statistics show that genetic algorithms are more popular than other optimization techniques. Also, hybrid algorithms are composed of a meta-heuristic and other solution techniques, for example a combination of GA & SA algorithms (Bettemir & Sonmez, 2015) are applied to solve the problem. However, other meta-heuristic techniques such as Particle Swarm Optimization (PSO), Ant Colony optimization (ACO), Bee Colony optimization (BCO), Simulated Annealing (SA), other Evolutionary Algorithm (EA), Tabu Search (TS), Teaching-learning- Based Optimization (TLBO), Distribution Estimation Algorithm (DEA) etc. are also developed to achieve the objective of the problem.

Based on the frequencies shown in Fig. 2, more than 60 percent of the meta-heuristics used consist of genetic algorithms, hybrid algorithms, and particle swarm optimization.

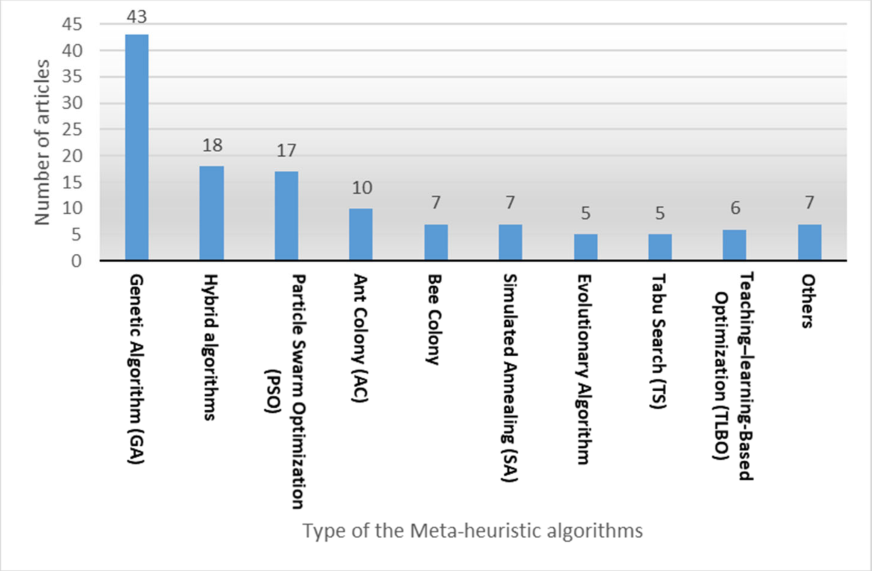

Fig. 2. Classified RCPSP articles according to the meta-heuristics. The three most popular solution techniques are genetic algorithms, hybrid algorithms, and particle swarm optimization.

3. Weighted articles

In our view, a decision method is used to select five highlighted articles that are helpful to summarize. For this purpose, the substantiated articles are categorized according to the years of their publication and then weighted based on the citations and the editor's impact factor. Table 1 presents the highlighted articles, each of which is then explained, respectively.

Table 1
Five weighted articles

No.	Articles	Citations	Year	Applied solution technique
1	A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version	762	2003	Simulated Annealing (SA)
2	A hybrid genetic algorithm for the resource-constrained project scheduling problem	370	2008	Hybrid Genetic Algorithm (HGA)
3	A random key based genetic algorithm for the resource constrained project scheduling problem	293	2009	Genetic Algorithm (GA)
4	A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem	161	2014	Particle Swarm Optimization- Hyper-Heuristic (PSO-HH)
5	An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times	108	2017	Genetic Algorithm (GA)

3.1. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version

A simulated annealing (SA) was developed for an RCPSP to minimize the project duration. The initial solution is generated by the heuristic shortest processing time method (SPT). Moreover, the presented technique uses a local search method that generates a set of neighbors. The neighbors are generated by randomly moving the activities to the new positions between the precedence constraint positions. They use the adopted Schedule Generation Scheme (SGS) to present the schedules. In this process, the two operations 'start time assignment' and 'time increment' alternate rapidly and repeatedly until all the tasks of the project are scheduled according to the predecessors and the constraint of the resources. Their proposed algorithm also benefits from multiple cooling chains and temperature lowering mechanisms to control the process of temperature lowering. The main termination condition of the algorithm is the total number of solutions generated. They perform tests to evaluate the impact of the defined parameters for tuning the algorithm parameters. They conclude that the proposed technique is simple and understandable, and it can improve the performance compared to the previously developed SA to RCPSP (Bouleimen & Lecocq, 2003).

3.2. A hybrid genetic algorithm for the resource-constrained project scheduling problem

Valls, Ballestin, and Quintanilla applied a hybrid genetic algorithm (HGA) to RCPSP. The HGA includes: a peak crossover operator that generates offspring according to the inherited peak resource consumption. Moreover, the peak point is randomly selected within [0,1]. A local improvement operator and a parent selection method are used in the procedure. Their HGA consists of two phases: the first phase is called general search and the second phase searches in the neighborhood of the best generated solutions. The population of each generation in the two phases has a size of POPsize and POPsize/2, respectively, but each activity list is defined as an individual and each individual is evaluated by a fitness function that determines the measure of the makespan. The serial schedule generation scheme (SGS) creates the active schedules that are checked for resource availability and earliest precedence. The fitted individuals in the population and another individual are randomly selected from the current population, which are called parents (Valls, Ballestin, & Quintanilla, 2008). The mutation operator exchanges the activities in the individual sequence with a certain probability (Hartmann, 1998). To select the best responses, they use the ranking method to select the best individuals of POPsize, then others are removed. In the second phase, the neighbor population of the best plan found in POPsize is used, then the 10 best schedules are selected. Finally, the procedure is applied to the standard j30, j60, j90 and j120 sets. They say that their procedure differs and is more powerful by three sections: the peak crossover operator, the use of double justification that systematically generates the qualified schedules, and the neighbor population generation procedure. (Valls, Ballestin, & Quintanilla, 2008)

3.3. A random key based genetic algorithm for the resource constrained project scheduling problem

A genetic algorithm was developed for RCPSP in which chromosomes are represented based on random keys. The random keys range from 0 to 1 and help in obtaining feasible offspring generated by crossover. In the proposed algorithm, each chromosome consists of two groups of genes where the first group represents the priorities and the second represents the delay time. The generated chromosomes are then decoded to generate the parameterized active schedules according to the priorities and the delay times of the activities. The fitness function, called the modified makespan is responsible for providing feedback to the algorithm. Since some makespans are the same and have different potential for improvement, the fitness function is formulated to combine the value of the makespan with the measure of potential improvement. To generate the next generation, the algorithm follows three actions. The best individuals are directly transferred to the next generation, while other individuals are generated by the operations crossover and mutation. To obtain the offspring, in one-point crossover, one parent is randomly selected from the top individuals while the other parent is randomly selected from the population. The defined mutation is equivalent to the generation of the original individuals, i.e., instead of the mutation operation, some new individuals are generated for the new generation. Finally, the method uses the Scheduling Generation Scheme

(SGS) to generate active schedules. The procedure was applied to the sets j30, j60 and j120. (Mendes, Gonçalves, & Resende, 2009).

3.4. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem

A particle swarm optimization (PSO)-based hyper-heuristic (PSO-HH) was developed for the classical RCPSP to obtain the feasible minimized makespan. The method is organized to consider the precedence relations and resource availability throughout the project execution. In this paper, each particle consists of eight integer numbers and each swarm consists of twenty particles. Moreover, the position of each particle of the swarm is initialized randomly and then the fitness function evaluates all the swarms. Initially, the global best solution is zero, and after processing, it is equal to the local best solution. The best solution consists of the best heuristic sequence and the best duration, which is stored to update the speed. Eight low-level heuristics are controlled by the procedure and randomly applied to the particles. They are called L1 to L8 and include the shift heuristics, replacement and interchange priorities, and crossover-based heuristics. Moreover, the method benefits from the standardized solution representation with random keys and the Serial Scheduling Generation Scheme (SSGS) to decode the particles and a forward-backward improvement method to improve the solutions. Finally, the standardized instances are used to prove the method (Koulinas, Kotsikas, & Anagnostopoulos, 2014)**.**

3.5. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times

Transfer times for units are transferred from one activity to another. They assumed that the activities have immediate predecessors and successors, where the lag times between the end and the beginning is zero and the durations are deterministic. Moreover, the objective function is defined as minimizing the project duration. The list of feasible activities contains a chromosome with two binary codes located respectively at the end of each solution. The first code indicates the type of SGS (serial or parallel), and the second binary code indicates the planning direction (forward or backward). Also, the initial population of size 200 is generated by randomly selecting three priority rules: minimum latest finish time, minimum latest start time, and minimum total slack. A two-point crossover operator is called a modified magnet-based crossover. After selecting the parents aimlessly, the crossover operator selects a block of activities from the donor parent. Then, the unconstrained activities are randomly replaced within the possible space and the other activities are copied from the receiver parent. A mutation operator is used to achieve diversity by applying the operation to all individuals in the population. The operator changes two consecutive activities if the feasible individuals are not disturbed. The procedure is applied to activity sets j30, j60, j90 and j120 (Kadri & Boctor, 2018). Table 2 and Table 3 show the results of the solution techniques in the highlighted articles.

Table 2
Average deviations from optimal solutions

Set	No.	applied algorithm	number of schedules		
			1000	5000	50000
J = 30	1	Simulated Annealing (SA)	0.38	0.23	---
	2	Hybrid Genetic Algorithm (HGA)	0.27	0.06	0.02
	3	Genetic Algorithm (GA)	0.06	0.02	0.01
	4	Particle Swarm Optimization- Hyper-Heuristic (PSO-HH)	0.26	0.04	0.01
	5	Genetic Algorithm (GA)	---	0.25	---

Table 3
Average deviations from critical path lower bounds

Set	No.	applied algorithm	number of schedules		
			1000	5000	50000
J = 60	1	Simulated Annealing (SA)	12.75	11.9	---
	2	Hybrid Genetic Algorithm (HGA)	11.56	11.1	10.73
	3	Genetic Algorithm (GA)	11.72	11.04	10.67
	4	Particle Swarm Optimization- Hyper-Heuristic (PSO-HH)	11.74	11.13	10.68
	5	Genetic Algorithm (GA)	33.83	32.61	---
j = 120	1	Simulated Annealing (SA)	42.81	37.68	---
	2	Hybrid Genetic Algorithm (HGA)	34.07	32.54	31.24
	3	Genetic Algorithm (GA)	35.87	33.03	31.44
	4	Particle Swarm Optimization- Hyper-Heuristic (PSO-HH)	35.2	32.59	31.23
	5	Genetic Algorithm (GA)	84.31	81.97	---

In the following, we briefly explain the summary of conventional metaheuristic solution techniques and then mention articles in each subsection.

4. Applied conventional Meta-heuristics to solve RCPSP

In this section, we present well-founded papers that use traditional meta-heuristic solution techniques. We briefly describe the conventional meta-heuristic solution techniques and then list the papers by the year of their publication, respectively.

4.1. Genetic algorithms and RCPSP

GA Approach comes from biology, where descendants want to inherit desirable traits. GAs also belong to evolutionary algorithms. In a standard algorithm GA an initial population is generated, a fitness function evaluates the individuals, then the algorithm tries to improve the solutions or individuals by employing the operators of crossover and mutation, and the selected operator selects the parents to generate the solutions of the offspring for the next generations. The algorithm proceeds to satisfy the given constraints. (Gendreau & Potvin, 2010). In the following, we briefly review related work A genetic algorithm was applied to solve RCPSP with minimization of project duration as the objective. Genetic encoding, priority values and priority rule representation are also used (Hartmann, 1998). Later, Hartman developed a self-adaptive genetic algorithm for RCPSP. The objective of the problem is set as minimization of makespan (Hartman, 2001). A robust genetic algorithm was presented for a resource-constrained project scheduling problem. The method generates a feasible activity list using a priority rule with a random key between 0 and 1. Also, a crossover operator randomly selects parents to mate (Alcaraz & Maroto, 2001). The developed genetic algorithm uses six operators to generate the offspring and establish diversity in the generations, while the objective of the problem is to minimize the makespan of the project (Hindi, Yang, & Fleszar, 2002). A genetic algorithm with a fuzzy logic controller was developed for RCPSP. The genetic operators were designed with fuzzy logic controller, the child casually takes some genes from one parent and then fills up the chromosome with the genes from the other parent by position based crossover, while the mutation operator randomly selects two positions of the chromosome to produce a child by swap mutation (Kim, Gen, & Yamazaki, 2003). A genetic bi-population algorithm was proposed for the RCPSP, which benefits from two separate populations containing left- and right-justified schedules, respectively. The Schedule Generation Scheme (SGS) is used to generate the schedule (Debels & Vanhoucke, 2005, May). A GA for RCPSP was investigated in which the activity durations, activity deadlines and project makespan are fuzzy defined (Wang, Lin, & Li, 2005). A hybrid genetic algorithm was presented for RCPSP, which benefits from a new representation of the activity list. The procedure uses the two-point crossover and the mutation operator that randomly selects genes for mutation (Alcaraz & Maroto, 2006). Sakalauskas and Felinskas applied a genetic algorithm based on a priority list of jobs to optimize makespan. The algorithm uses the methods of local and global search, a bit string crossover and a bit flipping mutation with given probability are used to achieve the goal (Sakalauskas L. a., 2006). The genetic algorithm is designed to minimize makespan. The initial population follows the left-justified schedule method, and a two-point crossover generates the offspring (Debels & Vanhoucke, 2007). Kim presented a permutation-based elitist GA, whose main aspect is called the elitist roulette selection operator, and which generates feasible solutions through a serial schedule generation scheme. The method uses a one-point crossover and uniform mutation to generate children (Kim J.-L. , 2007). Ballestín proposed a GA for RCPSP with minimum and maximum lag times and the objective is to minimize the cost of resource availability (Ballestin, 2007, April). An approach of GA uses an elitist strategy to select the best individuals for the next generation and a schedule generation scheme to generate feasible solutions. In addition, a one-point crossover and a uniform mutation operator were used to generate the offspring (Kim & Ellis Jr, 2008). A genetic algorithm was used to optimize the makepan of the project while resources are limited. Also, the method uses uniform crossover and swap mutation to generate the offspring (Frankola, Golub, & Jakobovic, 2008). Kim proposed an improved elite genetic algorithm to achieve project duration optimization. The method generates a random number for the initial generation. It also benefits from one-point crossover and uniform mutation to generate new solutions (Kim, 2009). A genetic algorithm based on an object-oriented model was proposed to minimize the project duration. The method uses a one-point and a two-point crossover and benefits from a mutation operator that mutates the genes of chromosomes according to a certain probability (Montoya-Torres, Gutierrez-Franco, & Pirachican-Mayorga, 2010). A genetic algorithm with neighborhood search was applied to RCPSP, including the non-preemptive activities, to minimize the makespan of the project. It is claimed that the neighborhood search operator can improve the feasible solution if the start times of some activities are fixed to search other activities (Proon & Jin, 2011). A GA was applied to RCPSP, where the procedure is equipped with random key, SSGS, backward-forward improvement, and a parameterized uniform crossover. Moreover, the objective is defined as minimizing the project duration (Gonalves, Resende, & Mendes, 2011). A procedure uses a standardized random key, a scheduling scheme, a local search procedure, and an elitist selection procedure to solve an RCPS problem. The crossover operator selects one of the parents from above and another randomly, while the procedure uses two mutation operators (Wang, Li, & Lin, 2010). A GA was developed to minimize the project duration. The initial feasible solutions are randomly generated. The method benefits from three types of crossover generations: one-point crossover, two-point crossover and priority-reserve crossover, also two mutation operators are employed to generate the offspring generation (Klimek, 2010). A genetic hyper heuristic algorithm has been proposed for the resource constrained project scheduling problem to minimize the makespan of the project (Anagnostopoulos & Koulinas, 2010, July). The method uses two crossover operators (one-point operator and uniform operator) and a classical mutation operator to minimize the makespan of the project (Ren, Kong, & Peng, 2011). To handle RCPSP, GA was used, with the fitness function feeding back the value of the project duration to the algorithm by evaluating the individuals. The crossover and mutation operators act on the current population, except for the identified fitted solutions (Zhu, Li, & Shen, 2011). The genetic algorithm was proposed to optimize the project duration of the problem, where the method uses a priority rule to generate the initial population (Gargiulo & Quagliarella, 2012). The completion phase of construction projects was considered by a GA. The algorithm benefits from a random initial generation, a one-point crossover operator and a random mutation. In addition, two selection methods are embedded in the algorithm (Dong, Ge, Fischer, & Haddad, 2012). A magnet-based crossover operator, which is a type of two-point crossover operator, was embedded in the method (Zamani, 2013). A genetic algorithm was developed to deal with a resource-constrained project scheduling problem where the fitness function has feedback with the makespan value. The binary string chromosomes represent the individuals and a one-point crossover generates the offspring (Diana, Ganapathy, & Pundir, 2013). A bi-objective problem was presented where the defined objectives are the makespan of the project and the NPV of the project. The article proposed genetic algorithms with two subpopulations to achieve the objectives (Khalili, Najafi, & Niaki, 2013). A genetic based local search approach was developed to optimize the project makespan. To achieve this, the neighborhood operator acts on a selected individual in the current population (Dridi, Krichen, & Guitouni, 2013). The article proposed a GA with the aim of minimizing the project duration. To achieve the objective, the method uses a one-point crossover operator and a mutation operator that exchange the two positions of two genes. Moreover, SGS technique is used to decode the individuals (Kadam & Kadam, 2014). A GA was developed to achieve the objective of the problem which was defined as minimizing makespan value. The procedure randomly generates the initial individuals and then benefits from a classical one-point crossover and a classical mutation operator that exchanges the position of the gene (Ali, Elsayed, Ray, & Sarker, 2015, May). A GA with priority-based crossover was introduced to minimize the project duration to deal with a resource-constrained project scheduling problem. A local search operator was also developed to improve the solutions (Kadam & Mane, 2015). A GA has been developed for minimizing the makespan. Moreover, the SGS is used to decode the chromosomes. To generate individuals, the method creates a random list of tasks. Subsequently, a local operator tries to improve feasible schedules (Goncharov & Leonov, 2017). The procedure was defined to have an optimized schedule. To achieve the objective, the elitist strategy is used to detect the fitted individuals. Then, a crossover operator generates a child from two parents. Finally, the non-fitted individuals in the current population are replaced by the offspring in the next generation. A mutation operator selects some individuals to increase diversity by applying local search. (Liu, Liu, Shi, & Li, 2020). As mentioned in section 2, the researchers developed the genetic algorithms to minimize the project duration. To achieve this, they have embedded various operations in the structure of the genetic algorithm.

In the following, the particle swarm optimization method and relevant articles are discussed

4.2. Particle Swarm Optimization (PSO) and RCPSP

Particle Swarm Optimization was modeled on social collective behavior, for example, the collective movement of birds or fishes (Kennedy & Eberhart, 1995). Normally, a PSO consists of a swarm of particles moving in an n-dimensional space. Each particle is characterized in the given time by its position, velocity vectors and its own best position. The position and velocity of the particles are initialized randomly and optimized periodically throughout the algorithm. There is also a fitness function to evaluate and compare the quality of the solutions. During the procedure, the best local particle is introduced as the global best solution. The procedure continues until the stopping conditions such as the maximum number of iterations or the computation time are satisfied (Tchomte & Gourgand, 2009). In the following the related works are mentioned.

A PSO was developed for RCPSP to minimize the project duration. Also, two methods of permutation-based representation (activity list) and priority-based representation (random key) are used (Zhang H. , Li, Li, & Huang, 2005). A hybrid particle method and mapped crossover were presented to optimize the makepan of the project. Moreover, the SGS method is used to convert the particles into feasible schedules (Zhang & Li, 2006). A PSO has been proposed to minimize the project duration, for this purpose the procedure uses an adopted updating velocity and updating position mechanisms (Peng & Wei, 2008). The anti-inertia solution generation rule and bidirectional search rule are embedded in a PSO to prevent local minimum and expand the solution space, respectively (Lo, Chen, Shiau, & Wu, 2008). A particle swarm optimization was presented to minimize the project duration, with two mechanisms embedded in the method to update the particle velocity and particle position (Zhang, Zhao, & Jiang, 2009). An improved particle swarm optimization was developed for RCPSP to obtain the optimized makespan (Wang & Qi, 2009). A delay local search rule and a bidirectional scheduling rule were added to the procedure to prevent staying in the local search and evolving the local search to achieve a global solution minimum (Chen, Wu, Wang, & Lo, 2010). A PSO was proposed for RCPSP to minimize the project duration using the methods of SSGS and forward-backward improvement (Li, Lai, & Shou, 2011). A pseudo PSO algorithm was developed where the velocity factor is not used in the procedure (Nasiri, 2013). The method used greedy random local search, double justification operator and SSGS method to minimize the makespan (Jia & Seo, 2013). The article proposed a radius particle swarm optimization, which benefits a mechanism that regroups particles within a suitable radius (Anantathanvit & Munlin, 2014, March). An improved method for determining the position of the particles and the velocities was proposed. Moreover, the objective of PSO was defined to obtain the optimized makespan (Kumar & Vidyarthi, 2016). A hybrid particle swarm optimization was developed against the RCPSP, where a maximum of one interruption per activity is allowed. The PSO benefits from three types of particle solution representation and two vector decoding methods (Shou, Li, & Lai, 2015). A PSO has been proposed to deal with RCPSP while the available quantities of resources are variable. However, the durations of the activities are fixed (Joy, Rajeev, & Narayanan, 2016). An adoptive mutation and forward-backward method were embedded in the proposed PSO to obtain a minimum project duration (Munlin M. , 2018). A particle swarm optimization based method was proposed against RCPSP. In this problem, a resource pool was defined among different sites. Moreover, two types of resources, named fixed and mobile resources, were assumed (Stiti & Driss, 2019).

Following the study of meta-heuristics, the method of Ant Colony -optimization and relevant articles are reviewed.

4.3. Ant Colony optimization (ACO) and RCPSP

Ant colony optimization is inspired by the collective social behavior of real ants. In general, ACO can be categorized as swarm intelligence. Normally, an ant colony uses pheromone trails as communication links between ants, allowing them to find short routes between their nest and food sources. The artificial ants are used to maintain a non-systematized structure and make probabilistic decisions depending on the pheromone trails (Gendreau & Potvin, 2010).

There are some types of ant colony optimization, but in a standard ACO, each ant makes a solution by probabilistic decisions. The ants that find a suitable solution then deposit a quantity of pheromone on the path of the search space. After that, the ants of the next generation follow the marked path or the suitable solution found nearby in the solution space (Merkle, Middendorf, & Schmeck, 2002). In this way, the feasible solutions in the neighborhood can be created and then evaluated to obtain solutions of good quality or the shortest solution path. The procedure continues until the termination conditions are satisfied (Dorigo & Di Caro, 1999). The related papers are mentioned below.

The ACO algorithm was proposed to optimize the project duration. To this end, the authors proposed a combination of local and global pheromone techniques to create the new solution (Merkle, Middendorf, & Schmeck, 2002). Another ant colony optimization was developed to minimize the project duration. It also uses shift and backshift operators to obtain the solutions of the neighbors (Luo, Wang, & Wang, 2003). To cope with the RCPSP, the proposed method uses delayed solution generation to escape the local optimum (Chen & Lo, 2006). The proposed method considers the effective allocation of project resources with two separate ant colonies (Shou, 2007). An ACO was developed for the problem where the duration of activities is defined within a range of lower and upper bounds (Yuan, Wang, & Ding, 2009, August). An improved ACO has been proposed for the RCPSP. The presented method uses a local search method called PC -2opt (Zhou, Guo, & Gan, 2009). An ACO-method has been proposed to minimize the project duration, benefiting from the SSGS method and dual justification (Deng, Lin, & Chen, 2010).

Bees colony optimization is the next metaheuristic technique studied in this paper.

4.4. Bees Colony Optimization (BCO) and RCPSP

Bee colony optimization belongs to the category of swarm-based optimization algorithms, which, like ant colonies, are inspired by the natural collective behavior of honeybees to find the flower patches. The search for the flower patches starts with some bees randomly searching the sources and exploring the spaces. The bees return to the hive to report the location of the flower patches by doing waggle dance to establish communication between the bees. Waggle dance helps the colony by relaying three pieces of information about the flower patches: Direction, distance, and quality of the source.

There are different types of colony optimization, but a standard bee algorithm is based on a random solution and a neighborhood solution. In this algorithm, an initial population is created, which is then evaluated using a fitness function. Additional bees are used to create the neighborhood solutions for selected parts of the search space. The process continues until the given constraints are satisfied (Pham, et al., 2006). The related papers are mentioned below.

To minimize the duration of project implementation, an artificial bee colony was developed. Moreover, the method uses SSGS to generate the feasible schedules (Akbari, Zeighami, & Ziarati, 2010). The proposed ABC algorithm uses a random number to select one of the two SGS methods, which is directly used to generate schedules (Shi, Qu, Chen, & Li, 2010). A bee algorithm introduced a new formula to evaluate the quality of the solutions found in the search space (Sadeghi, Kalanaki, Noktehdan, Samghabadi, & Barzinpour, 2011). Against the stochastic RCPSP, an artificial bee colony was proposed where the activity duration is variable with a certain probability. Moreover, the defined objective of the method is to minimize the project duration (Tahooneh & Ziarati, 2011). The paper investigated three types of bee algorithms, also a method used to convert infeasible schedules into feasible schedules. The method benefits from local search where priority values are exchanged to create a neighbor solution (Ziarati, Akbari, & Zeighami, 2011). The developed bees colony uses a one-point crossover operator to create neighbor solutions. Also in this paper, the facility layout concept method is used to formulate the RCPSP (Jia & Seo, 2013). To overcome the problem, an artificial bee colony was proposed in which three operators based on swap are randomly selected to generate the neighbor solutions (Crawford, Johnson, Norero, & Olgun, 2015). The method of simulated annealing and relevant articles are discussed below.

4.5. Simulated Annealing (SA) and RCPSP

Simulated annealing is referred to as SA and was inspired by the process of physical annealing of solids. When a crystalline solid is heated and then slowly formed into a solid, a qualified solid is produced with minimal energy. The SA algorithm combines this part of thermodynamic science with local search to obtain a minimal solution (Gendreau & Potvin, 2010).

At SA, an initial solution is used to start with, and there is always a current solution. Moreover, the neighbors of the current solution can also be replaced by the current solution if the neighbor solution is more practical than the current solution, but an impractical solution can be a current solution in certain cases to prevent a local optimum. Moreover, a temperature parameter is incorporated in the simulated annealing, which has a higher value initially and then slowly decreases to obtain a better solution. Moreover, the solutions are evaluated with a fitness function (Agarwal, Colak, & Erenguc, Metaheuristic Methods, 2015) The related papers are mentioned below.

A simulated annealing was presented to minimize the total project time for RCPSP. To achieve the objective of the problem, the generated solution is presented with a priority list (Cho & Kim, 1997). Boctor developed a SA algorithm for the RCPS problem where there are renewable resources from period to period. Moreover, the algorithm benefits from an adopted neighborhood operator (Boctor, 1996). The proposed algorithm focuses on scheduling orders, and the schedule is encoded by a priority list of jobs (Sakalauskas & Felinskas, 2006). The presented SA uses a tabu list to search for a neighbor solution (Das & Acharyya, 2011). An improved simulated annealing has been proposed to minimize the duration of project completion, while the second objective of the problem studies the consumption of resources among the same obtained solutions (Pan & Lin, 2011).

Tabu search and its relevant articles are discussed in the next section of the paper.

4.6. Tabu Search (TS) and RCPSP

Tabu search is based on a local search technique formed on the basis of displacement strategies and neighborhood solution search. It starts with an initial solution that can be feasible. The neighborhood solution is created by moving in the search space. Then the selected action is transferred to the tabu list for a certain number of iterations to avoid reaching a local minimum, but in some cases the tabu action can be selected if it leads to a better solution according to the established admission criteria. The solutions are evaluated with a fitness function. The procedure continues until the termination conditions such as the number of iterations are satisfied (Thomas & Salhi, 1998). The related papers are mentioned below.

A tabu search method has been presented to minimize the makespan of a project. The neighbor solution is obtained by a single swap or insert operation, and also the tabu status is updated repeatedly (Thomas & Salhi, 1998). The improved tabu search focuses on minimizing the project duration, while the proposed method uses the slack time and available resources to obtain an initial solution (Pan, Hsaio, & Chen, 2008). The proposed method uses prioritization of activities to obtain the initial solution according to the slack time, while anticipation of activities and partial allocation of resources are not allowed (Atli, 2011). Two different neighborhood generation approaches have been proposed for tabu search. The first one is based on the exchange of resources assigned to a pair of tasks and the second one is based on the assignment of any resource that could perform an identified task (Skowroński, Myszkowski, Adamski, & Kwiatek, 2013). In an improved TS, four Neighborhood operators are proposed: swap operation, insertion operation, exchange operation, and shift operation. In addition, two mutation operators are embedded in the procedure (Dai, Cheng, & Guo, 2018).

4.7. Teaching–Learning-Based Optimization (TLBO) and RCPSP

This meta-heuristic technique draws inspiration from the teaching-learning phenomenon that the best student can be a teacher to others. There are a few types of TLBO, but a standard algorithm consists of two phases: the teacher phase and the student phase. First, a population of solutions is randomly initialized according to the defined parameters, then the best solution plays the role of the teacher. Each student interacts with the teacher to create new solutions. Then, each student or solution interacts with other solutions to obtain a new solution. The fitness function evaluates the solutions to find a new teacher. The process continues until the defined termination condition, which can be the maximum number of generations (Rao, Savsani, & Vakharia, 2011) (Zheng, Wang, & Zheng, 2017). The related papers are mentioned below.

A coevolutionary TLBO has been proposed to deal with RCPSP while there are two initialized classes at the beginning of the process, the first step of the process is called the competition phase (Zheng, Wang, & Wang, 2014). A TLBO algorithm with ordinal interval numbers was developed for RCPSP to minimize the project duration. Two phases of self-study and testing are embedded in the algorithm (Zheng & Wang, 2015). A reinforcement phase is incorporated into the TLBO algorithm to minimize the project duration. Moreover, the task-resource list is constructed by combining the activity list and resource list (Zheng, Wang, & Zheng, 2017). Two phases of self-study and testing are proposed to increase the performance of TLBO with the objective of minimizing the makespan or total project duration (Joshi, Mittal, Sharma, & Kumar, 2019).

Evolutionary algorithms are other population-based metaheuristics that will be discussed in the next part.

4.8. Evolutionary Algorithms and RCPSP

Evolutionary algorithms are population-based metaheuristic techniques that often begin randomly generating solutions, much like genetic algorithms. There is also a fitness function to evaluate the solutions, which helps to select the appropriate solutions for the next generation. There are also improvement techniques that try to improve the solutions during the process. The related papers are mentioned below. An evolutionary multi-agent algorithm has been proposed for the RCPSP in which three operators - competition, crossover and self-learning - are used to solve the problem (Pan & Chen, 2010). An evolutionary algorithm was proposed that benefits from a conglomerate-based crossover operator that combines the good parts of solution (Ballestín, Barrios, & Valls, 2011). In the proposed method, a new solution representation technique based on ordered events list was introduced (Paraskevopoulos, Tarantilis, & Ioannou, 2012). A differential evolution algorithm with local search method was presented for resource-constrained project scheduling problem to minimize the makespan and total cost (Eshraghi, 2016). A differential evolution algorithm was developed for multi-skill RCPSP with reassignment function embedded to improve the solution quality at the end of each iteration (Quoc, The, Doan, & Thanh, 2020). Following the study of meta-heuristics, the developed hybrid algorithms and their corresponding articles are discussed.

4.9. Hybrid algorithms and RCPSP

Combining different metaheuristic techniques or combining a metaheuristic with other methods, called hybrid optimization or hybrid algorithms or metaheuristic hybrids, are used to achieve better performance on complex problems, but using an effective hybrid approach is a challenge because choosing an appropriate combination is not easy (Gendreau & Potvin, 2010). The related papers are mentioned below. A hybrid metaheuristic, which is a combination of ant colony optimization (ACO), genetic algorithm (GA) and local search method, has been adopted for the resource-constrained project scheduling problem (Tseng & Chen, 2006). A hybrid of ACO and PSO algorithms was developed for the RCPSP to optimize the makespan of the project with minimum lag times (Shan, Wu, & Peng, 2007). A mixture of genetic algorithm and simulated annealing was proposed for the RCPSP to improve the performance of the procedure where GA generates the population and SA tries to improve the individuals (Yu, Zhan, Nie, & Xu, 2009). A combination of particle swarm optimization and genetic algorithm was developed for RCPSP to minimize the project duration (Li, Zhang, Jiang, & Xie, 2009). A hybrid of ant colony optimization and scatter search was presented, where the ACO searches the solution space and generates an activity list, and then the SS algorithm tries to improve the solutions (Chen, Shi, Teng, Lan, & Hu, 2010). A neurogenetic approach has been proposed, which is a combination of genetic algorithm and neural network. The GA performs the process of global search and the NN works on local search (Agarwal, Colak, & Erenguc, 2011). A hybrid algorithm, a combination of simulated annealing, tabu search, and genetic algorithm, was developed to determine the optimal project duration (Thammano & Phu-Ang, 2012). The proposed hybrid algorithm works by the interaction of a genetic algorithm and artificial bee colony whose objective is to minimize the project duration (Zeighami, Akbari, Akbari, & Biletskiy, 2012). A hybrid strategy is based on combining the parallel search of the genetic algorithm with the tuning capabilities of the simulated annealing method against RCPSP (Bettemir & Sonmez, 2015). A hybrid approach used heuristic priority rules with ant colony optimization for multi-skill RCPSP to optimize project duration and project cost (Myszkowski, Skowronski, Olech, & Oslizlo, 2015). A hybrid greedy search and genetic algorithm were developed for minimizing project makespan (Delgoshaei, Ariffin, Baharudin, & Leman, 2015). A hybrid GA has been proposed where the SA algorithm acts like an operator of the genetic algorithm to maximize the NPV of the project (Fathallahi & Najafi, 2016). The proposed approach is based on PSO which is cooperating with mutation operators and forward-backward improvement methods to improve the process of local search methods in the procedure (Munlin & Anantathanavit, 2016). A hybrid of tabu search and simulated annealing algorithms has been presented to minimize the project duration (Afshar-Nadjafi, Yazdani, & Majlesi, 2017). A hybrid TLBO-TS algorithm has been proposed to achieve the objective of maximizing the total expected benefits from the selected project portfolio (Kumar, Mittal, Soni, & Joshi, 2018). The proposed hybrid approach is based on a combination of differential evolution algorithm and cuckoo search algorithm. In addition, a local forward-backward improvement is used to improve the new solutions (Sallam, Chakrabortty, & Ryan, 2019). A hyper-procedure, called self-adaptive differential evolution, has been developed for fussy stochastic RCPSP. In the procedure, the activity durations and makespan are estimated fussy and random; new individuals are also generated by the operators of mutation and crossover (Alipouri, Sebt, Ardeshir, & Chan, 2019). The frequency of developed hybrid algorithms present that choosing an appropriate combination is important. Therefore, researchers try different combinations to obtain an efficient hybrid algorithm.

4.10. Other metaheuristics for RCPSP

A social evolutionary multi-agent algorithm in which agents behave in three ways: competition, crossover and self-learning. The method was proposed to optimize the project duration (Pan & Chen, 2010). A distribution estimation algorithm was proposed for RCPSP, in which the solutions are generated by the priority rule of latest finish time (LST) and the random method, and then decoded using SSGS (Fang, Wang, & Xu, 2010). An artificial immune algorithm was considered for RCPSP to minimize the project duration. The method uses two mutation operations to generate new solutions (Mobini, Mobini, & Rabbani, 2011). A distribution estimation algorithm was developed to optimize the project duration. The method uses a local search operator and a forward-backward iteration method to improve the solutions (Wang & Fang, 2012). A firefly algorithm was employed to deal with the RCPSP to minimize the project duration (Sanaei, Akbari, Zeighami, & Shams, 2013). A distribution estimation algorithm with a binary random variable matrix was proposed to solve the RCPSP (Fang, Kolisch, Wang, & Mu, 2015).

In Section 4, the developed meta-heuristics were presented, and the published articles were discussed in the corresponding subsections.

Based on the frequency of the published articles, it is understandable that researchers pay special attention to the resource-constrained project scheduling problem. So, it is obvious that researchers try to develop various methods to minimize the project makespan.

5. Challenges of meta-heuristics methods

There are many heuristics and metaheuristics solution techniques for the Resource-Constrained Project Scheduling Problem (RCPSP), but they have difficulty optimizing in the real projects (Koulinas, Kotsikas, & Anagnostopoulos, 2014). As mentioned in the previous sections, these methods start with a solution or a set of solutions as a population and then try to generate the more suitable solutions during the process. So, it is obvious that the quality of the initial solutions is important to obtain the optimal or near-optimal solution at the end of the process. On the other hand, there are two main constraints: precedence constraints, and resource constraints mentioned in the introduction. So, it is very important to obtain a feasible activity list or solution according to the activity relations because without considering the constraints, the solution is wrong. So, these methods use rules like priority rules to get feasible solutions (Alipouri, Sebt, Ardeshir, & Chan, 2019). Some procedures generate the solutions and then test their feasibility, i.e., they need to change the placement of the activities in the activity list to obtain the feasible schedule. As explained later in this paper, after initializing the solutions, the procedures use the operators to improve the solutions. However, one of the difficulties of these techniques is tuning the parameters of the algorithms, which is a time-consuming process (Koulinas, Kotsikas, & Anagnostopoulos, 2014). For example, in a simulated annealing algorithm, tuning the parameters for the initial temperature, the attenuation factor of the temperature in each step, and the number of neighbor solutions is important for the method to work well. Moreover, the termination condition of the procedure can be determined by the total time of the procedure implementation, the total number of neighbor solutions generated, or a defined condition for the objective value, which should be tested by previous experiments (Bouleimen & Lecocq, 2003). So, these methods try to improve the solutions by generating a group of solutions or generating the neighbor solutions for the next generation of solutions, because these methods are based on the search and repeated generation of solutions, as mentioned later in the paper. On the other hand, the methods are not always successful in finding the optimum. Sometimes they find a local solution or a solution close to the optimum, or the procedures even fail due to the recursion of the procedure when the algorithm is executed. For example, a meta-heuristic obtains the optimal solutions for the 89 instances by running 120 instances(Kadri & Boctor, 2018). As mentioned in the introduction, after the ineffectiveness of the exact method to solve large RCPSP, researchers have developed heuristics and metaheuristics for large problems, but the common and traditional metaheuristics, such as the approaches explained in the main body of this paper, cannot be efficient enough given the limitations of this type of algorithms. Therefore, researchers are searching and developing more effective algorithms (Jedrzejowicz & Ratajczak-Ropel, 2014). From our point of view, the creation of an optimal activity sequence is the key to a suitable method that is better able to generate an optimal activity sequence, as described in this article. Therefore, using new approaches such as machine learning and neural networks based on learning and prediction may be more useful to achieve the objective.

6. Conclusion

In this paper, a review on Resource-Constrained Project Scheduling Problem (RCPSP) was carried out. Since the RCPSP is known as NP -complete (non-deterministic polynomial problem), a plenty of approaches are proposed to solve this problem. Due to the large number of studies conducted in the problem, it is required to summarize these articles. For this reason, the aim of this article was to review the papers on RCPSPs. In this respect, the standard RSPCP was discussed, and then the weighted articles were stated. Following the article, the conventional metaheuristics and related papers were presented. It was pointed out that the conventional metaheuristics are more practical than exact methods to deal with large-size problems. Moreover, it was also mentioned that the conventional methods cannot guarantee to find the optimum or an optimum close to it due to existing limitations. It was noticed that there is a lack of investigating the problem using new methods such as neural networks and machine learning. Following the studied papers, we claimed that the creation of an optimal activity sequence is the key to a suitable method. Therefore, we suggested that using new methods such as neural networks and machine learning could be more helpful than traditional methods to achieve an optimal project duration. In the future, we will focus on the new methods to deal with the RCPSP and present the advantages or disadvantages of the new approaches compared to conventional techniques to solve the RCPSP.

References

Afshar-Nadjafi, B., Yazdani, M., & Majlesi, M. (2017). A Hybrid of Tabu Search and Simulated Annealing Algorithms for Preemptive Project Scheduling Problem. *In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 102-111). Cham: Springer.

Agarwal, A., Colak, S., & Erenguc, S. (2011). A Neurogenetic approach for the resource-constrained project scheduling problem. *Computers & Operations Research, 38*(1), 44–50.

Agarwal, A., Colak, S., & Erenguc, S. (2015). Metaheuristic Methods. In *Handbook on Project Management and Scheduling* (Vol. 1, pp. 57-74). cham: Springer.

Akbari, R., Zeighami, V., & Ziarati, K. (2010). Artificial Bee colony for resource constrained project scheduling problem. *International Journal of Industrial Engineering Computations, 2*(1), 45-60.

Alcaraz, J., & Maroto, C. (2001). A Robust Genetic Algorithm for Resource Allocation in Project Scheduling. *Annals of operations Research, 102*(1), 83-109.

Alcaraz, J., & Maroto, C. (2006). a hybrid genetic algorithm based on intelligant endoding for project schedulig. In *Perspectives in modern project scheduling* (pp. 249-274). Boston: Springer.

Ali, I. M., Elsayed, S. M., Ray, T., & Sarker, R. A. (2015, May). Memetic Algorithm for solving Resource Constrained Project Scheduling Problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2761-2767). IEEE.

Alipouri, Y., Sebt, M. H., Ardeshir, A., & Chan, W. T. (2019). Solving the FS-RCPSP with hyper-heuristics: A policy-driven approach. *Journal of the Operational Research Society, 70*(3), 403-419.

Anagnostopoulos, K. P., & Koulinas, G. K. (2010, July). A Genetic Hyperheuristic Algorithm for the Resource Constrained Project Scheduling Problem. In *IEEE Congress on Evolutionary Computation* (pp. 1-6). IEEE.

Anantathanvit, M., & Munlin, M.-A. (2014, March). Radius particle swarm optimization for resource constrained project scheduling problem. In *16th Int'l Conf. Computer and Information Technology* (pp. 24-29). IEEE.

Atli, O. (2011). Tabu Search and an Exact Algorithm for the Solutions of Resource-Constrained Project Scheduling Problems. *International Journal of Computational Intelligence Systems, 4*(2), 255-267.

Ballestin, F. (2007, April). A Genetic Algorithm for the Resource Renting Problem with Minimum and Maximum Time Lags. Dans *European Conference on Evolutionary Computation in Combinatorial Optimization* (pp. 25-35). Berlin, Heidelberg: Springer.

Ballestín, F., Barrios, A., & Valls, V. (2011). An evolutionary algorithm for the resource-constrained project scheduling problem with minimum and maximum time lags. *Journal of scheduling, 14*(4), 391-406.

Bettemir, Ö. H., & Sonmez, R. (2015). Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. *Journal of Management in Engineering, 31*(5).

Bettemir, Ö. H., & Sonmez, R. (2015). Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. *Journal of Management in Engineering, 31*(5).

Boctor, F. F. (1996). Resource-constrained project scheduling by simulated annealing. *International Journal of Production Research, 34*(8), 2335-2351.

Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European journal of operational research, 149*(2), 268-281.

Chen, R.-M., & Lo, S.-T. (2006). Using an Enhanced Ant Colony System to Solve Resource-Constrained Project Scheduling Problem. *Int. J. Comput. Sci. Netw. Secur, 6*, 75-85.

Chen, R.-M., Wu, C.-L., Wang, C.-M., & Lo, S.-T. (2010). Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB. *Expert systems with applications, 37*(3), 1899–1910.

Chen, W., Shi, Y.-j., Teng, H.-f., Lan, X.-p., & Hu, L.-c. (2010). An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences, 180*(6), 1031-1039.

Cho, J., & Kim, Y.-D. (1997). A simulated annealing algorithm for resource constrained project scheduling problems. *Journal of the Operational Research Society, 48*(7), 736-744.

Cho, J., & Kim, Y.-D. (1997). A simulated annealing algorithm for resource-constrained project scheduling problems. *Journal of the Operational Research Society, 48*(7), 736-744.

Crawford, B. a., Johnson, F., Norero, E., & Olgun, E. (2015). An Artificial Bee Colony Algorithm for the Resource Contrained Project Scheduling Problem. In *International Conference on Human-Computer Interaction* (pp. 582-586). Springer.

Dai, H., Cheng, W., & Guo, P. (2018). An Improved Tabu Search for Multi-skill Resource-Constrained Project Scheduling Problems Under Step-Deterioration. *Arabian Journal for Science and Engineering, 43*(6), 3279–3290.

Das, P. P., & Acharyya, S. (2011). Simulated Annealing Variants for Solving Resource Constrained Project Scheduling Problem: A Comparative Study. In *14th International Conference on Computer and Information Technology (ICCIT 2011)* (pp. 469-474). IEEE.

Debels, D., & Vanhoucke, M. (2005, May). A Bi-population Based Genetic Algorithm for the Resource-Constrained Project Scheduling Problem. In *International Conference on Computational Science and Its Applications* (pp. 378--387). Berlin, Heidelberg: Springer.

Debels, D., & Vanhoucke, M. (2007). A Decomposition-Based Genetic Algorithm for the Resource-Constrained Project-Scheduling Problem. *Operations Research, 55*(3), 457-469.

Delgoshaei, A., Ariffin, M., Baharudin, B., & Leman, Z. (2015). Minimizing makespan of a resource-constrained scheduling problem: A hybrid greedy and genetic algorithms. *International Journal of Industrial Engineering Computations, 6*(4), 503–520.

Deng, L., Lin, V., & Chen, M. (2010). Hybrid ant colony optimization for the resource-constrained project scheduling problem. *Journal of Systems Engineering and Electronics, 21*(1), 67-71.

Diana, S., Ganapathy, L., & Pundir, A. K. (2013). An Improved Genetic Algorithm for Resource Constrained Project Scheduling Problem. *International Journal of Computer Applications, 78*(9), 34-39.

Dong, N., Ge, D., Fischer, M., & Haddad, Z. (2012). A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects. *Advanced Engineering Informatics, 26*(4), 737-748.

Dorigo, M., & Di Caro, G. (1999). Ant Colony Optimization: A New Meta-Heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)* (pp. 1470-1477). IEEE.

Dridi, O., Krichen, S., & Guitouni, A. (2013). Solving Resource-Constrained Project Scheduling Problem by A Genetic Local Search Approach. Dans *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)* (pp. 1-5). IEEE.

Eshraghi, A. (2016). A new approach for solving resource constrained project scheduling problems using differential evolution algorithm. *International journal of industrial engineering computations, 7*(2), 205-216.

Fang, C., Kolisch, R., Wang, L., & Mu, C. (2015). An estimation of distribution algorithm and new computational results for the stochastic resource constrained project scheduling problem. *Flexible Services and Manufacturing Journal, 27*(4), 585–605.

Fang, C., Wang, L., & Xu, Y. (2010). An estimation of distribution algorithm for resource-constrained project scheduling problem. In *2010 Chinese Control and Decision Conference* (pp. 265-270). IEEE.

Fathallahi, F., & Najafi, A. A. (2016). A hybrid genetic algorithm to maximize net present value of project cash flows in resource constrained project scheduling problem with fuzzy parameters. *Scientia Iranica, 23*(4), 1893-1903.

Frankola, T., Golub, M., & Jakobovic, D. (2008). Evolutionary Algorithms for the Resource Constrained Scheduling Problem. In *ITI 2008-30th International Conference on Information Technology Interfaces* (pp. 715-722). IEEE.

Gargiulo, F., & Quagliarella, D. (2012). Genetic Algorithms for the Resource Constrained Project Scheduling Problem. In *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)* (pp. 39-47). IEEE.

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of metaheuristics* (Vol. 2). New York: Springer.

Gonalves, J. F., Resende, M. G., & Mendes, J. J. (2011). A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem. *Journal of Heuristics, 17*(5), 467–486.

Goncharov, E. N., & Leonov, V. V. (2017). Genetic Algorithm for the Resource-Constrained Project Scheduling Problem. *Automation and Remote Control, 78*(6), 1101-1114.

Hartman, S. (2001). A comparative genetic algorithm for resource-constraint project scheduling. *Naval Research Logistics, 49*, 433-488.

Hartmann, S. (1998). A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. *Naval Research Logistics, 45*(7), 733-750.

Hindi, K. S., Yang, H., & Fleszar, K. (2002). An Evolutionary Algorithm for Resource-Constrained Project Scheduling. *IEEE Transactions on evolutionary computation, 6*(5), 512-518.

Jedrzejowicz, P., & Ratajczak-Ropel, E. (2014). Reinforcement Learning strategies for A-Team solving the Resource-Constrained Project Scheduling Problem. *Neurocomputing, 146*, 301-307.

Jia, Q., & Seo, Y. (2013). An improved particle swarm optimization for the resource-constrained project scheduling problem. *The International Journal of Advanced Manufacturing Technology, 67*(9-12), 2627-2638.

Jia, Q., & Seo, Y. (2013). Solving resource-constrained project scheduling problems: Conceptual validation of FLP formulation and efficient permutation-based ABC computation. *Computers & Operations Research, 40*(8), 2037-2050.

Joshi, D., Mittal, M., Sharma, M. K., & Kumar, M. (2019). An effective teaching-learning based optimization algorithm for the multi-skill resource-constrained project scheduling problem. *Journal of Modelling in Management, 14*(4), 195-211.

Joy, J., Rajeev, S., & Narayanan, V. (2016). Particle swarm optimization for resource Constrained-project scheduling problem with varying resource levels. *Procedia Technology, 25*, 948-954.

Kadam, S. U., & Kadam, N. S. (2014). Solving Resource-Constrained Project Scheduling Problem by Genetic Algorithm. In *2014 2nd International Conference on Business and Information Management (ICBIM)* (pp. 159-164). IEEE.

Kadam, S. U., & Mane, S. U. (2015). A Genetic-Local Search Algorithm Approach for Resource Constrained Project Scheduling Problem. In *2015 International Conference on Computing Communication Control and Automation* (pp. 841-846). IEEE.

Kadri, R. L., & Boctor, F. F. (2018). An effcient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times. *European Journal of Operational Research, 295*(2), 454-462.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of ICNN'95-international conference on neural networks* (pp. 1942-1948). IEEE.

Khalili, S., Najafi, A. A., & Niaki, S. T. (2013). Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: two meta-heuristic algorithms. *The International Journal of Advanced Manufacturing Technology, 69*(1-4), 617-626.

Kim, J.-L. (2007). Permutation-based elitist genetic algorithm using serial scheme for large-sized resource-constrained project schedulig. In *2007 Winter Simulation Conference* (pp. 2112-2118). IEEE.

Kim, J.-L. (2009). Improved genetic algorithm for resource constrained scheduling of large projects. *Canadian journal of civil engineering, 36*(6), 1016-1027.

Kim, J.-L., & Ellis Jr, R. D. (2008). Permutation-Based Elitist Genetic Algorithm for Optimization of Large-Sized Resource-Constrained Project Scheduling. *Journal of construction engineering and management, 134*(11), 904-913.

Kim, K. W., Gen, M., & Yamazaki, G. (2003). Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied soft computing, 2*(3), 174-188.

Klimek, M. (2010). A genetic algorithm for the project scheduling with the resource constraints. *Annales Universitatis Mariae Curie-Sklodowska, sectio AI--Informatica, 10*(1), 117-130.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research, 174*(1), 23-37.

Koulinas, G., Kotsikas, L., & Anagnostopoulos, K. (2014). A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. *Information Sciences, 277*, 680-693.

Kumar, M., Mittal, M. L., Soni, G., & Joshi, D. (2018). A hybrid TLBO-TS algorithm for integrated selection and scheduling of projects. *Computers & Industrial Engineering, 119*, 121-130.

Kumar, N., & Vidyarthi, D. P. (2016). A model for resource-constrained project scheduling using adaptive PSO. *Soft Computing, 20*(4), 1565-1580.

lazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete applied mathematics, 5*(1), 11-24.

Li, F., Lai, C., & Shou, Y. (2011). Particle swarm optimization for preemptive project scheduling with resource constraints. In *2011 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 869-873). IEEE.

Li, M., Zhang, Y., Jiang, W., & Xie, J. (2009). A Particle Swarm Optimization Algorithm with Crossover for Resource Constrained Project Scheduling Problem. In *2009 IITA International Conference on Services Science, Management and Engineering* (pp. 69-72). IEEE.

Liu, J., Liu, Y., Shi, Y., & Li, J. (2020). Solving Resource-Constrained Project Scheduling Problem via Genetic Algorithm. *Journal of Computing in Civil Engineering, 34*(2).

Lo, S.-T., Chen, R.-M., Shiau, D.-F., & Wu, C.-L. (2008). Using particle swarm optimization to solve resource-constrained scheduling p,roblems. In *2008 IEEE Conference on Soft Computing in Industrial Applications* (pp. 38-43). IEEE.

Luo, S., Wang, C., & Wang, J. (2003). Ant Colony Optimization for Resource-Constrained Project Scheduling with Generalized Precedence Relations. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligenc* (pp. 284-289). IEEE.

Mendes, J. J., Gonçalves, J. F., & Resende, M. G. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research, 36*(1), 92-109.

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation, 6*(4), 333-346.

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant Colony Optimization for Resource-Constrained Project Scheduling. *IEEE transactions on evolutionary computation, 6*(4), 333-346.

Mobini, M., Mobini, Z., & Rabbani, M. (2011). An Artificial Immune Algorithm for the project scheduling problem under resource constraints. *Applied Soft Computing, 11*(2), 1975-1982.

Montoya-Torres, J. R., Gutierrez-Franco, E., & Pirachican-Mayorga, C. (2010). Project scheduling with limited resources using a genetic algorithm. *International Journal of Project Management, 28*(6), 619-628.

Munlin, M. (2018). Solving resource-constrained project scheduling problem using metaheuristic algorithm. In *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)* (pp. 344-349). IEEE.

Munlin, M., & Anantathanavit, M. (2016). Hybrid Radius Particle Swarm Optimization. In *2016 IEEE Region 10 Conference (TENCON)* (pp. 2180-2184). IEEE.

Myszkowski, P. B., Skowronski, M. E., Olech, L. P., & Oslizlo, K. (2015). Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. *Soft Computing, 19*(12), 3599-3619.

Nasiri, M. M. (2013). A pseudo particle swarm optimization for the RCPSP. *The International Journal of Advanced Manufacturing Technology, 65*(5-8), 909-918.

Ortiz-Pimiento, N. R., & Diaz-Serna, F. J. (2018). The project scheduling problem with non-deterministic activities duration: A literature review. *Journal of Industrial Engineering and Management (JIEM), 11*(1), 116-134.

Pan, N.-H., & Lin, Y.-Y. (2011). Using hybrid simulated annealing algorithm in resource constrained project scheduling problem. *Journal of Statistics and Management Systems, 14*(3), 555-582.

Pan, N.-H., Hsaio, P.-W., & Chen, K.-Y. (2008). A study of project scheduling optimization using Tabu Search algorithm. *Engineering Applications of Artificial Intelligence, 21*(7), 1101–1112.

Pan, X., & Chen, H. (2010). A Multi-Agent Social Evolutionary Algorithm for Resource-Constrained Project scheduling. In *2010 International Conference on Computational Intelligence and Security* (pp. 209-213). IEEE.

Pan, X., & Chen, H. (2010). A Multi-Agent Social Evolutionary Algorithm for Resource-Constrained Project Scheduling. In *2010 International Conference on Computational Intelligence and Security* (pp. 209-213). IEEE.

Paraskevopoulos, D. C., Tarantilis, C. D., & Ioannou, G. (2012). Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm. *Expert Systems with Applications, 39*(4), 3983–3994.

Peng, W., & Wei, Y. (2008). PSO for Solving RCPSP. In *2008 Chinese control and decision conference* (pp. 818-822). IEEE.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The Bees Algorithm- A Novel Tool for Complex Optimisation Problems. In *Intelligent production machines and systems* (pp. 454-459). Elsevier.

Proon, S., & Jin, M. (2011). A Genetic Algorithm with Neighborhood Search for the Resource-Constrained Project Scheduling Problem. *Naval Research Logistics (NRL), 58*(2), 73-82.

Quoc, H. D., The, L. N., Doan, C. N., & Thanh, T. P. (2020). New Effective Differential Evolution Algorithm for the Project Scheduling Problem. In *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)* (pp. 150-157). IEEE.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching–learning-based optimization : A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303-315.

Ren, Y. H., Kong, D. C., & Peng, W. L. (2011). A Genetic Algorithm based Solution with Schedule Mode for RCPSP. In *Advanced Materials Research* (pp. 1802-1805). Trans Tech Publ.

Roy, B., & Sen, A. K. (2019). Meta-heuristic Techniques to Solve Resource-Constrained Project Scheduling Problem. In *International conference on innovative computing and communications* (pp. 93-99). Springer.

Sadeghi, A., Kalanaki, A., Noktehdan, A., Samghabadi, A. S., & Barzinpour, F. (2011). Using Bees Algorithm to Solve the Resource Constrained Project Scheduling Problem in PSPLIB. In *International Conference on Theoretical and Mathematical Foundations of Computer Science* (pp. 486-494). Berlin, Heidelberg: Springer.

Sakalauskas, L. a. (2006). Optimization of resource constrained project schedules by genetic algorithm based on the job priority list. *Information technology and control, 35*(4), 412-419.

Sakalauskas, L., & Felinskas, G. (2006). Optimization of Resource-Constrained Project Schedules by Simulated Annealing and Variable Neighborhood Search. *Technological and economic development of economy, 12*(4), 307-313.

Sallam, K. M., Chakrabortty, R. K., & Ryan, M. J. (2019). A Hybrid Differential Evolution with Cuckoo Search for Solving Resource Constrained Project Scheduling Problems. In *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 1344-1348). IEEE.

Sanaei, P., Akbari, R., Zeighami, V., & Shams, S. (2013). Using Firefly Algorithm to Solve Resource Constrained Project Scheduling Problem. In *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)* (pp. 417-428). Springer.

Shan, M., Wu, J., & Peng, D. (2007). Particle Swarm and Ant Colony Algorithms Hybridized for Multi-mode Resource-constrained Project Scheduling Problem with Minimum Time Lag. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing* (pp. 5898-5902). IEEE.

Shi, Y.-j., Qu, F.-Z., Chen, W., & Li, B. (2010). An Artificial Bee Colony with Random Key for Resource-Constrained Project Scheduling. In *Life system modeling and intelligent computing* (pp. 148-157). Berlin, Heidelberg: Springer.

Shou, Y. (2007). A Bi-directional Ant colony algorithm for resource constrained project scheduling. In *2007 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1027-1031). IEEE.

Shou, Y., Li, Y., & Lai, C. (2015). Hybrid particle swarm optimization for preemptive resource-constrained project scheduling. *Neurocomputing, 148*, 122–128.

Skowroński, M. E., Myszkowski, P. B., Adamski, M., & Kwiatek, P. (2013). Tabu Search approach for Multi–Skill Resource–Constrained Project Scheduling Problem. In *2013 Federated Conference on Computer Science and Information Systems* (pp. 153-158). IEEE.

Sprecher, A., Kolisch, R., & Drexl, A. (1995). Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research, 80*(1), 94-102.

Stiti, C., & Driss, O. B. (2019). A new approach for the multi-site resource-constrained project scheduling problem. *Procedia Computer Science, 164*, 478-484.

Tahooneh, A., & Ziarati, K. (2011). Using Artificial Bee Colony to Solve Stochastic Resource Constrained Project Scheduling Problem. In *International Conference in Swarm Intelligence* (pp. 293-302). Berlin, Heidelberg: Springer.

Tchomte, S. K., & Gourgand, M. (2009). Particle swarm optimization : A study of particle displacement for solving continuous and combinatorial optimization problems. *International Journal of Production Economics, 121*(1), 57-67.

Thammano, A., & Phu-Ang, A. (2012). A hybrid evolutionary algorithm for the resource-constrained project scheduling problem. *Artificial Life and Robotics, 17*(2), 312–316.

Thomas, P. R., & Salhi, S. (1998). A Tabu Search Approach for the Resource Constrained Project Scheduling Problem. *Journal of Heuristics, 4*(2), 123-139.

Thomas, P. R., & Salhi, S. (1998). A Tabu Search Approach for the Resource Constrained Project Scheduling Problem. *Journal of Heuristics, 4*, 123-139.

Tseng, L.-Y., & Chen, S.-C. (2006). A hybrid metaheuristic for the resource-constrained project scheduling problem. *European Journal of Operational Research, 175*(2), 707-721.

Valls, V., Ballestin, F., & Quintanilla, S. (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research, 185*(2), 495-508.

Wang, H., Li, T., & Lin, D. (2010). Efficient Genetic Algorithm for Resource-Constrained Project Scheduling Problem. *Transactions of Tianjin University, 16*(5), 376–382.

Wang, H., Lin, D., & Li, M. (2005). A Genetic Algorithm for Solving Fuzzy Resource-Constrained Project Scheduling. In *International Conference on Natural Computation* (pp. 171-180). Berlin, Heidelberg: Springer.

Wang, L., & Fang, C. (2012). A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem. *Journal of Expert Systems with Applications, 39*, 2451-2460.

Wang, Q., & Qi, J. (2009). Improved Particle Swarm Optimization for RCP Scheduling Problem. In *The Sixth International Symposium on Neural Networks (ISNN 2009)* (pp. 49-57). Berlin: Springer.

Yu, X., Zhan, D., Nie, L., & Xu, X. (2009). A Novel Genetic Simulated Annealing Algorithm for the Resource-Constrained Project Scheduling Problem. In *2009 International Workshop on Intelligent Systems and Applications* (pp. 1-4). IEEE.

Yuan, Y., Wang, K., & Ding, L. (2009, August). A Solution to Resource-Constrained Project Scheduling Problem. In *2009 Ninth International Conference on Hybrid Intelligent Systems* (Vol. 1, pp. 446-450). IEEE.

Zamani, R. (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *European Journal of Operational Research, 229*(2), 552–559.

Zeighami, V., Akbari, R., Akbari, I., & Biletskiy, Y. (2012). An ABC-Genetic method to solve resource constrained project scheduling problem. *Artificial Intelligence Research (AIR) Journal, SCIEDU, Canada, 1*(2), 185-197.

Zhang, H., & Li, H. :. (2006). Permutation-based particle swarm optimization for resource-constrained project scheduling. *Journal of computing in civil engineering, 20*(2), 141-149.

Zhang, H., Li, X., Li, H., & Huang, F. (2005). Particle swarm optimization-based schemes for resource-constrained project scheduling. *Automation in construction, 14*(3), 393-404.

Zhang, K., Zhao, G., & Jiang, J. (2009). Particle Swarm Optimization Method for Resource-Constrained Project Scheduling Problem. In *2009 9th International Conference on Electronic Measurement & Instruments* (pp. 4-792). IEEE.

Zheng, H.-y., & Wang, L. (2015). An effective teaching–learning-based optimisation algorithm for RCPSP with ordinal interval numbers. *International Journal of Production Research, 53*(6), 1777-1790.

Zheng, H.-y., Wang, L., & Wang, S.-y. (2014). A Co-evolutionary Teaching-learning-based Optimization Algorithm for Stochastic RCPSP. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 587-594). IEEE.

Zheng, H.-y., Wang, L., & Zheng, X.-l. (2017). Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem. *Soft Computing, 21*(6), 1537-1548.

Zhou, Y., Guo, Q., & Gan, R. (2009). Improved ACO Algorithm for Resource-Constrained Project Scheduling Problem. In *2009 international conference on artificial intelligence and computational intelligence* (Vol. 3, pp. 358-365). IEEE.

Zhu, J., Li, X., & Shen, W. (2011). Effective genetic algorithm for resource-constrained project scheduling with limited preemptions. *International Journal of Machine Learning and Cybernetics, 2*(2), 55–65.

Ziarati, K., Akbari, R., & Zeighami, V. (2011). On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing, 11*(4), 3720–3733.