Integration of Multi-mode Resource-Constrained Project Scheduling under bonus-penalty policies with Material Ordering under Quantity Discount scheme for minimizing project cost

M. Akhbari[1]

Department of Industrial Engineering, Electronic Branch, Islamic Azad University, Tehran, Iran

Abstract

In this paper a mixed binary integer mathematical programming model is developed for integration of Multimode Resource-Constraint Project Scheduling Problem (MRCPSP) under bonus–penalty policies and Quantity Discount Problem in Material Ordering (QDPMO) with the objective of minimizing the total project cost. By proving a theorem, an important property of the optimum solution of the problem is found which reduces the search space significantly compared to previous studies. Since the RCPSP belongs to the class of problems that are NP-hard, four hybrid meta-heuristic algorithms called COA-GA, GWO-GA, PSO-GA and GA-GA is developed and tuned to solve the problem. Each of the proposed algorithms consists of outside and inside search components, which determine the best schedule, and materials procurement plan respectively. Finally a set of standard PROGEN test problems is solved by the proposed hybrid algorithms under fixed CPU time. The results show that the COA-GA algorithm outperforms others.

Keywords**:** Project scheduling, Multi-mode resource-constrained, Quantity Discount Problem in Material Ordering, Grey Wolf Optimizer, Coyote Optimization Algorithm, Genetic algorithm, Particle swarm optimization

1. Introduction

Over the past decades, project scheduling has attracted extensive attention from both scientific and practical perspectives. The Resource-Constrained Project Scheduling Problem (RCPSP) introduced by Kelley in 1963 is one of the most widely studied cases of project scheduling problems. In its standard form, RCPSP attempts to minimize the makespan of a

[1] Corresponding author. Tel.: +98 21 42863000
E-mail address: m_akhbari@iauec.ac.ir

project by assigning a start time to each non-preemptive activity with respect to the precedence relations and the scarce renewable resource availabilities. While the makespan minimization is one of the most popular objectives, there are various other objectives such as the project cost minimization, the earliness and tardiness minimization, the minimization of resource idle time or the maximization of project net present value (NPV).

The standard RCPSP assumed that each activity can have only one execution way which is determined by a fixed duration and fixed required resources. Elmaghraby [1] changed this assumption by allowing several alternatives or modes in which an activity can be executed. This extension leads to the well-known multi-mode resource-constrained project scheduling problem (MRCPSP).

Slowinski [2] proposed a linear programming optimization method to solve the MRCPSP and Talbot [3]and Patterson et al. [4] designed an approach based on an enumeration scheme. Drexl and Gruenewald [5] suggested an efficient stochastic scheduling method which solves MRCPSPs. Also, Speranza and Vercellis [6] suggested a depth-first branch-and-bound algorithm; their algorithm correctness was examined by Hartmann and Sprecher [7], and the results showed that the algorithm may not lead to an optimal solution when involving more than one renewable resource. Sprecher et al. [8] presented an exact algorithm based on a generalization of the branch-and-bound algorithm which was proposed by Demeulemeester and Herroelen [9] to solve RCPSP. They enhanced the basic enumeration scheme by defining dominance rules to increase the performance of the algorithm.

Boctor [10] attempted to identify the most efficient heuristics in a comparative study and suggested a combination of five-heuristics to solve MRCPSPs. In 1996, he developed another efficient heuristic for solving MRCPSPs outperformed the heuristics of his previous five-heuristics [11]. A year later, Kolisch and Drexl [12] proposed a local search method consisting of three phases: 1) A local search phase which tries to reach an initial feasible solution, 2) A construction phase which performs a single-neighborhood search on the sets of feasible mode assignments and finally, 3) An intensification phase which tries to find a schedule with an improved objective function based on the best mode assignment. Besides, they showed in the presence of at least two nonrenewable resources, the problem of discovering a feasible schedule is NP-complete. Moreover, Sprecher and Drexl [13] proved that exact optimization methods are not able to solve the MRCPSP with more than 20 activities and three modes for each activity in the logical elapsed run time.

Due to the hardness of the problem, various meta-heuristic methods have been proposed in the literature. Bouleimen and Lecocq [14] presented a simulated annealing algorithm to solve this problem. In 2001, Hartmann [15] proposed a solution method based on genetic algorithm to solve MRCPSP. Jozefowska et al. [16] developed another simulated annealing approach and compared results with two previous algorithms. Alcaraz et al. [17], developed a new genetic algorithm, extending the representation, operators, and fitness function in comparison with Hartmann's algorithm. In their algorithm, infeasible individuals can participate in the genetic process and transmit their good characteristics to their offspring. Mika et al. [18], considered MRCPSP with positive discounted cash flows and different payment models and proposed solution methods based on simulated annealing and tabu search algorithms. Zhang et al. [19] introduced a methodology for solving the MRCPSP based on particle swarm optimization (PSO). Jarboui et al. [20] suggested a combinatorial particle swarm optimization for solving this problem. Van Peteghem and Vanhoucke [21] presented an Artificial Immune System (AIS) for the MRCPSP. They also proposed solution methods based on genetic algorithm to solve preemptive and non-preemptive MRCPSPs [22]. Barrios et al. [23] developed a two-phase genetic algorithm for the MRCPSP with the maximum time lags. Khalilzadeh et al. [24] presented a metaheuristic algorithm based on the Tchomté and Gourgand's modified Particle Swarm Optimization (PSO) to solve MRCPSP with the objective function of minimization the total costs of both renewable and nonrenewable resource usage. They developed a prioritization rule for activities and several improvements and local search methods. Wang and Fang [25] designed an estimation of distribution algorithm (EDA) for solving the MRCPSP. Li and Zhang [26] presented another solution method for the MRCPSP based on the Ant colony optimization (ACO). Messelis and Causmaecker [27] constructed an automatic algorithm selection tool for the MRCPSP. This super-algorithm chooses an algorithm from a portfolio of state-of-the-art algorithms based on the characteristics of the given instance. Results showed that it outperforms all of the algorithms individually.

In all of the mentioned studies, if non-renewable resources (material) are presented, the availability of them throughout the project lifetime is one of the assumptions of the model. In fact, If the material planning is done before the scheduling of the project activities, it may lead to increased costs such as holding costs because of the time difference between the supply and consumption dates of the materials or the cost of delay in the projects with the bonus–penalty policies due to the lack of the materials to start the activities earlier. In this study, we consider simultaneous multi-mode resource-constrained project scheduling and material ordering problem.

Aquilano and Smith [28] considered simultaneous material ordering and project scheduling problem firstly and suggested a hybrid model combining the critical path method (CPM) with material requirements planning (MRP). Later, their work was improved by Smith-Daniels and Aquilano [29] presenting a heuristic procedure for scheduling large projects subject to the availability of renewable and non-renewable resources. Smith-Daniels and Smith-Daniels [30] presented a mixed integer binary programming formulation of a Project Scheduling-Materials Ordering Problem (PSMOP) to find an optimal schedule of both project activities and materials orders. In another study, they developed their work by considering maximizing net present value as the objective and suggested the late start date schedule [31]. Erbasi and Sepil [32] also considered a heuristic method to find the trade-off between material ordering expenses and project delay.

In all the studies mentioned above, the price of the material was considered constant, while in the real world, purchasing options might change according to the order quantity.

Dodin and Elimam [33] extended the previous works considering variable activity duration, the bonus–penalty policies and material ordering quantity discounts and formulated the problem as a mixed-integer programming model. They used the conventional branch and bound algorithm to solve the instance problems with up to thirty activities.

Sajadieh et al. [34] extended the research of Dodin and Elimam [33] by developing a solution approach based on GA, so that the model can be solved for large-scale PSMOPs. Tabrizi and Ghaderi [35] proposed a mixed-integer programming model for PSMOP with the objective of maximizing NPV. They considered the presence of multiple suppliers offering distinctive discount strategies. They developed a genetic algorithm to solve large-scale problems. Moreover, they tested the effect of inflation on the objective function value (NPV) by a sensitivity analysis. In the subsequence study, Tabrizi and Ghaderi [36] developed a robust multi-objective mixed-integer programming mathematical model for PSMOP. The purpose of the research was to minimize execution costs and maximize the schedule robustness. They applied the NSGA-II and a modified version of multi-objective differential evolution algorithm as the solution methodologies.

Zoraghi et al. [37], considered the multi-mode resource-constrained project scheduling and the material ordering problem (MRCSMOP) simultaneously to minimize the total project's costs consisting of material holding cost, material ordering cost, bonus paid by the client and penalty of possible delays. To solve the problem, they developed three hybrid meta-heuristic algorithms called PSO-GA, GA-GA, and SA-GA, which each of them includes two parts: an outside search, for the finding the best schedule and mode assignment, and inside search to determine time and quantity of orders. Shahsavar et al. [38], considered a combination of project scheduling problem (PSP) and material ordering problem (MOP) with the quantity discount policies and constructed three hybrid methods: GA–DP, GA–SA or GA–PSO. Zoraghi et al. [39] extended the MRCSMOP to a problem with three objectives: minimizing the makespan, maximizing the schedule robustness and minimizing the total costs including renewable and nonrenewable resources costs. They also investigated the total quantity discount policy. They applied Four multi-objective evolutionary algorithms; which are: 1) non-dominated sorting genetic algorithm II (NSGAII), 2) strength Pareto evolutionary algorithm II (SPEAII), 3) multi-objective particle swarm optimization (MOPSO), and 4) multi-objective evolutionary algorithm based on decomposition (MOEAD); in order to find an optimal Pareto frontier for the developed triple objective model.

Recently, Tabrizi [40] addressed simultaneous planning of the project schedule and material procurement problems and developed a bi-objective mathematical model with the goal to minimize total project costs and the environmental impacts of its execution. They applied NSGA-II and multi-objective migrating bird optimization (MOMBO) algorithms to find solutions considering the start time of activities as well as the time and quantity of material orders to each supplier.

In this study, a combination of Multimode Resource-Constraint Project Scheduling Problem (MRCPSP) and Quantity Discount Problem in Material Ordering (QDPMO) is addressed. It is worth noting, this study is an extension to the problems investigated by Zoraghi et al. [37] and Shahsavar et al. [38] considering multi-mode activities, quantity discount policies and bonus–penalty policy, simultaneously in order to get close to the real-world applications. Also, according to the assumptions, by proving a theorem, a property of the optimum solution will be presented for simplicity and higher efficiency of problem-solving algorithms.

The paper is organized in 8 sections. In Sec.2, the problem is defined and the assumptions are introduced. The notations and the integrated model entitled to MRCPSP–QDPMO is fully described and mathematically formulated in Sect.3. Sect.4 demonstrates a mathematical proof to narrow the solutions' search space. In Sect. 5 we provide the descriptions of the hybrid algorithms developed for the problem. Sec.6, we first explained the process of tuning the algorithm parameters using statistical techniques and then included the computational results of algorithms. A sensitivity analysis comes next in Sect. 7 and finally the conclusion remarks are reported in Sec.8.

2. Problem definition

Consider a project consist of non-preemptable activities which each of them can be executed in multiple modes with different duration and resource requirements. These activities have to be scheduled under precedence rules and resource constraints. Each activity could start if all of its predecessors have finished, all of its renewable resources are available and also its required non-renewable resources or materials have been provided. The main assumptions are: the initial inventory of all items with the number of non-renewable resources set to zero, no more than one order can be placed for a specific item in a period of time, and depending on the order quantity, there is an all unit discount scheme to purchase some items.

Also, we defined the total cost of project as costs of renewable and non-renewable resources plus the penalty or bonus resulting from the project tardiness or earliness.

The solving procedure for MRCPSP–QDPMO problem tries to find a schedule satisfying relations and resource feasibilities, as well as a material ordering plan (times and quantities of orders) with the objective of minimizing the total cost of project.

3. Mathematical formulation

Summary of all notations used in the model is presented in Table 1:

By using the above notations, the proposed Mixed Integer Programming (MIP) model can be formulated as:

MRCPSP–QDPMO Model:

$$MinZ = \sum_{i=1}^{n}\sum_{m=1}^{M_i}\sum_{t=1}^{T}\sum_{p=1}^{P} r_{pim} cr_p d_{im} x_{imt} + \sum_{q=1}^{Q}\sum_{t=1}^{T} H_q I_{qt} + \sum_{q=1}^{Q}\sum_{t=1}^{T} A_q \lambda_{qt}$$
$$+\sum_{q=1}^{Q}\sum_{t=1}^{T}\sum_{k=1}^{K} \mu_{tqk} P_{qk} Q_{qt} + \varpi\left(FT - D\right) y - \vartheta\left(D - FT\right)\left(1 - y\right) \tag{1}$$

subject to:

$$\sum_{m=1}^{M_i}\sum_{t=1}^{T} x_{imt} = 1 \quad \forall i \in \{0,1,\ldots,n\} \tag{2}$$

$$pred_{ij} \sum_{m=1}^{M_i}\sum_{t=1}^{T} t x_{imt} \le \sum_{m=1}^{M_i}\sum_{t=1}^{T} \left(t - d_{jm}\right) x_{jmt} \quad \forall i, j \in \{0,1,\ldots,n\} \tag{3}$$

$$\sum_{m=1}^{M_i}\sum_{t=1}^{T} t x_{imt} \le FT \quad \forall i \in \{0,1,\ldots,n\} \tag{4}$$

$$\sum_{i=1}^{n}\sum_{m=1}^{M_i} r_{pim} \sum_{b=t}^{t+d_{im}-1} x_{imt} \le res_{pt} \quad \forall t \in \{1,\ldots,T\}, \forall p \in \{1,2,\ldots,P\} \tag{5}$$

$$I_{qt} = I_{qt-1} + Q_{qt} - \sum_{i=1}^{n}\sum_{b=1}^{t+d_{im}-1}\sum_{m=1}^{M_i} a_{qim} x_{imb} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{6}$$

$$I_{q0} = 0 \forall q \in \{1,2,\ldots,Q\} \tag{7}$$

$$Q_{qt} \le \lambda_{qt} BigM \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{8}$$

$$1-(1-\lambda_{qt})BigM \le Q_{qt} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{9}$$

$$\mu_{tqk} O_{k-1q} \le Q_{qt} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\}, \forall k \in \{1,\ldots,K_q\} \tag{10}$$

$$\mu_{tqk} Q_{qt} \le O_{kq} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\}, \forall k \in \{1,\ldots,K_q\} \tag{11}$$

$$\sum_{k=1}^{K} \mu_{tqk} = 1 \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{12}$$

$$FT(1-y) \le D \tag{13}$$

$$Dy \le FT \tag{14}$$

$$I_{qt} \ge 0 \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{15}$$

$$x_{imt} \in \{0,1\} \quad \forall i \in \{0,1,\ldots,n\}, \forall t \in \{1,\ldots,T\}, \forall m \in \{1,\ldots,M_i\} \tag{16}$$

$$y \in \{0,1\} \tag{17}$$

$$\lambda_{qt} \in \{0,1\} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{18}$$

$$\mu_{tqk} \in \{0,1\} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\}, \forall k \in \{1,\ldots,K_q\} \tag{19}$$

The objective function (1) minimizes the total cost of project considering six factors which are:

1. The renewable resources cost
2. The inventory holding cost
3. The material ordering cost
4. Procurement cost
5. The project tardiness penalty
6. The project earliness bonus

Eq. (2) states that every activity is assigned exactly one mode and exactly one finishing time. Constraint (3) ensures the precedence relations between activities. Inequality (4) ensures that the makespan is the maximum of its all activities finish times. Constraint (5) enforces the renewable resource constraints at time interval t. Equation (6) determines the inventory level of materials at the end of each period of the project. Based on equation (7) no inventory level is available in the advent of the project. Inequalities (8) and (9), ensure that the order quantity of each material for each period can be provided if it is ordered in that period. Constraints (10) and (11) determine the discount range of each material in each period. Equation (12) ensures only one price is allocated to each material in each period. Inequalities (13) and (14) determine the earliness or tardiness of the project. Finally, constraints (15-19) denote the domain of the variables.

4. Optimum Solution Property

This section demonstrates an important property of the optimum MRCPSP–QDPMO solution, which can reduce the search space significantly. The following theorem establishes that in the optimum solution, only the start time of activities can be used to order tasks.

Theorem 1: In the optimum MRCPSP–QDPMO solution, the required quantity of each material in each period is ordered at the beginning of that period time or is merged with the last order of the material.

In other words, if the required quantity of a material is ordered any time after the last replenishment time (s) of this item and before t, the total cost of project will be increased.

Proof:

Let Q. be the required quantity of material q in the period t of the optimum MRCPSP–QDPMO solution, so:

$$Q = \sum_{i=1}^{n}\sum_{m=1}^{M_i} a_{qim} x^{*}_{im(t+d_{im}-1)} \tag{20}$$

Let PTC_b^t be the Total Cost of Q if i th is ordered in period b and consumed in period t ($s \le b \le t$).

So TC_t^t indicates the Total Cost for Q which is ordered and consumed in period t:

$$TC_t^t = A_q + P_q Q \tag{21}$$

and TC_e^t is the Total Cost of Q if it is ordered in period e , $s < e < t$:

$$TC_e^t = A_q + P_q Q + \sum_{j=1}^{t-e} H_q Q \quad s < e < t \tag{22}$$

And

$$TC_t^t < TC_e^t \tag{23}$$

So the early ordering in period e, which $s < e < t$, is not suggested. But the ordering of Q in period s, and buying the material in large quantities may bring better price and decrease the total cost (TC_s^t) in the comparison with TC_t^t , so the best order period of the required quantity of material q is either t or s.

Remark 1. Based on Theorem 1, in the optimum solution, only the start times of activities can be used as the ordering points.

Using this important property of the optimum solution significantly reduces the search space of the algorithm in comparison with the methods used by Tabrizi and Ghaderi [36], Zoraghi et al. [37], Shahsavar et al. [38], and Zoraghi et al. [39] in which all time periods are considered as the possible ordering points.

5. Solving Methodologies

In this section, we discuss four hybrid meta-heuristic algorithms, proposed to solve the problem. Each hybrid solving procedure consists of two optimization levels: an outside search level and an inside search level which generate the MRCPSP solution and QDPMO solution parts of the problem respectively. The MRCPSP solution part represents the best priority of the project activities and their assignment mode and the QDPMO solution part represents its best ordering plan.

5-1. Solution Representation

MRCPSP solution part is represented by a 2-dimensional matrix, $Y = \left[Y_{ji}\right]_{2\times(n)}$ where n is the number of project activities. Each Y_{ji} is generated randomly within interval [0,1]. The first row of Y relates to the sequence in which the activities are scheduled and the second row contributes to the mode of activities and each Y_{2i} is converted to a mode based on $m_i = \left[Y_{2i} * M_i\right] + 1$,where m_i is the execution mode of activity i.

The activity sequence list and the activity mode list act as the inputs of the serial scheduling scheme (SSS). In each step of the SSS, the activity with no unscheduled predecessor with the least sequence value is selected to be scheduled based on its mode [41].

Fig. 1 represented an example of the MRCPSP solution part of a project with five activities which can be executed in one of three possible modes.

QDPMO solution part is represented by a 2-dimensional matrix, $X = \left[X_{qj}\right]_{Q\times U}$ where Q is the number of project material types and U is the number of unique activities' start time based on the schedule generated by SSS. Each $j \in \{1, 2, \ldots, U\}$ is a period of time in project in which at least one activity starts.

X_{qj} is a 0-1 integer. If it equals to 1, it means the required quantity of material q in the j th period will be ordered at the same period. Otherwise, if X_{qj} equals 0, it means the required quantity of material q in the j th period will be ordered with the last ordering of material q. $X_{qj}, q \in \{1,2,\ldots,Q\}$ and $j \in \{2,3,\ldots,U\}$ generated randomly and since there is not any ordering point before the first start time, each $X_{q1}, q \in \{1,2,\ldots,Q\}$ equals 1.

5.2- Solution procedure

The pseudo-code of the solution procedure follows:

Call Outside search (COA, GWO, PSO or GA):

Execute following steps till the stopping condition is met:

Initialize MRCPSP solutions (Y s)

***For** each Y*

Call Serial Scheduling Scheme to find activities' start time

Call Inside search (GA):

Execute following steps till the stopping condition is met:

Receive activities' start time set based on Y

Initialize QDPMO solutions (X s)

Execute the procedure of the inside search and create new QDPMO solutions

Find the best ordering plan(bestX)

Compute the fitness value of whole solution Y and bestX

End For

Receive the outcome of Inside search

Execute the procedure of the outside search and create new MRCPSP solutions

Call Inside search (GA) to find the best ordering plan and fitness value

Return the best solution

The characteristics of these algorithms are explained in the following subsections.

5.3- The outside search algorithms

5-3-1. Coyote Optimization Algorithm (COA)

Coyote Optimization Algorithm (COA) was proposed by Pierezan and Coalho [42] is a recent population-based algorithm which is inspired by the social structure and experience exchange among coyotes in nature.

In the COA, the population of coyotes is divided into N_p packs with N_c coyotes in each pack. So, the total population is $N_p * N_c$. The social conditions of the coyotes (soc) are the solutions for the optimization problem and its adaptation to the environment is the cost of the objective function. Thus each $soc$is the set of decision variables, as follows:

$$soc_c^{p.t} = Y = \left[Y_{ji} \right]_{2\times n} \tag{24}$$

where t is the current iteration, c and p indicate the related coyote and pack respectively and n is the number of project activities and each Y_{ji}. $j \in \{1,2\}$ and $i \in \{1,2,\ldots,n\}$ is the social condition which is the random variable calculated as follow:

$$Y_{ji} = lb_{ji} + r_{ji}\left(ub_{ji} - lb_{ji}\right) \tag{25}$$

where lb_{ji} and ub_{ji} are lower and upper bounds of Y_{ji} (in this problem lb_{ji} is 0 and ub_{ji} is 1), and r_{ji} is a random number between$[0.1]$, so $Y_{ji} = r_{ji}$.

The coyotes' adaptation in the respective current social conditions ($fit_c^{p,t}$) is evaluated based on the objective function ($f(x)$):

$$fit_c^{p,t} = f\left(soc_c^{p,t}\right) \tag{26}$$

Initially, the coyotes are randomly assigned to the packs, however the coyotes sometimes leave their pack and joint to another pack by chance. The coyote eviction probability depends on the number of coyotes of the pack and is calculated as follows:

$$P_e = 0.005 N_c^{\ 2} \tag{27}$$

Based on equation (27), the number of coyotes per pack is at most 14. This process helps the COA to diversify the interaction between all the coyotes and simulates a cultural exchange in the global population.

The birth of a new coyotes is written as a combination of the social conditions of parents (randomly chosen) plus an environmental influence, such that:

$$pup_k^{p,t} = \begin{cases} soc_{r_1,k}^{p,t}, & rnd_k < P_a or k = k_1 \\ soc_{r_2,k}^{p,t}, & rnd_k \geq P_s + P_a or k = k_2 \\ R_k, & otherwise \end{cases} \tag{28}$$

where r_1 and r_2 are random coyotes from the p th pack, k_1 and k_2 are two random dimensions of the problem. P_s is the scatter probability while P_a is the association probability. Also R_k is a random number inside the k th decision variable bound and rnd_k is a random number inside [0,1]. P_s and P_a are defined as:

$$P_s = \frac{1}{D} \tag{29}$$

$$P_a = \frac{1-P_s}{2} \tag{30}$$

where D is the size of the problem which in this study is 2*n*. It is worth noting that *n* is the number of project activities.

Let $age_c^{p,t}$ be the age of c th coyote in the pth pack at the t th instant of time. The higher the coyote's age, the higher the mortality probability.

In order to keep the population size constant, COA performs this procedure: if the new pup has the worst objective value in the pack, it doesn't have the chance to live. Otherwise in the group of coyotes with the adaption value worse than the pup, the oldest one dies and if there is more than one coyote with the same highest age in this group, the coyote with the least adaption value dies.

In this algorithm, $alpha^{p,t}$ is the fittest solution in the p th pack in the t th instant of time.

Due to the evident signs of swarm intelligence in this specie, the COA assumes that the coyotes are sufficiently organized to share the social conditions. Thus, the COA aggregates all information from the coyotes and the cultural tendency of each pack is computed as the median social conditions of all coyotes as follows:

$$cult_i^{p,t} = \begin{cases} O_{\frac{(N_c+1)}{2},i}^{p,t} & N_c is odd \\ \dfrac{O_{\frac{N_c}{2},i}^{p,t} + O_{\frac{N_c}{2}+1,i}^{p,t}}{2} & otherwise \end{cases} \tag{31}$$

where $O_{j,i}^{p,t}$ represents the jth rank of the ordered ith social condition of all coyotes in the pth pack at the tth instant of time.

In each iteration of COA, every coyote is updated using alpha and the cultural tendency of its pack as follows:

$$new-soc_c^{p,t} = soc_c^{p,t} + r_1\left(alpha^{p,t} - soc_{cr_1}^{p,t}\right) + r_2\left(cult^{p,t} - soc_{cr_2}^{p,t}\right) \tag{32}$$

Where r_1 and r_2 are two random numbers inside [0,1] and cr_1 and cr_2 are two random coyotes in the pack.

The next population would be updated as follows:

$$soc_c^{p,t+1} = \begin{cases} new-soc_c^{p,t} & f\left(new-soc_c^{p,t}\right) < f\left(soc_c^{p,t}\right) \\ soc_c^{p,t} & otherwise \end{cases} \tag{33}$$

The pseudo code of the COA algorithm is as follows:

```
Initialize  Np packs with Nc coyotes each
Calculate the coyotes' adaptation (Eq. 26)
While (t<Max number of iterations)
        for each pack
                Find the alpha coyote of the pack
                Compute the social tendency of the pack (Eq. 31)
                for each coyote
                        Update the social condition (Eq. 32)
                        Evaluate the new social condition and update population (Eq. 33)
                end for
        Choose two coyotes randomly as parents and generate a new coyote (Eq. 28)
                If there is(are) coyote(s) with adaptation worse than the new coyote
                The new coyote survives and the oldest worse coyote dies
                else
                The new coyote dies
                end if
        end for
        Replacement of coyotes between packs
        Update the coyotes ages
        t = t + 1
end while
Return the best coyote
```

5-3-2. Grey Wolf Optimizer (GWO)

Grey Wolf Optimizer (GWO) described by Mirjalili et al [43] is a population-based algorithm which is inspired by the social hierarchy and hunting behavior of grey wolves in nature.

Social hierarchy is simulated by defining four types of grey wolves such as alpha, beta, delta, and omega. As well as, hunting behavior is simulated by implementing the three main steps of hunting, searching for prey, encircling prey and attacking prey [44].

A. Hierarchy of grey wolves

The leaders of wolves are a male and a female, called alphas. They typically responsible for making decisions about hunting, sleeping time and place, and so on. The wolves in the second level are called beta, which are the successor of alphas and help them to make a decision. The lowest on the hierarchy is called omega which plays the role of scapegoat. Omega is able to satisfy all the other dominant wolves. The third level on the hierarchy is called delta, which is subordinate to alpha and beta, but they dominate the omega. Delta should scout to protect and ensure the safety of the whole grey wolves' group.

In the mathematical model of GWO, alpha (α) is considered as the fittest solution. Consequently, beta (β) and delta (δ) are the second and third best solutions respectively. The remaining solutions are omega (ω). In the GWO algorithm the hunting (optimization) is guided by α, β, and δ. The ω wolves follow them.

B. Mathematical model of hunting behavior

Another amusing gregarious characteristic of gray wolves is their flock hunting process. In their hunting process, there are three main phases [45]: firstly, tracking, chasing, and approaching the prey; and then pursuing, encircling, and harassing the prey until it stops moving and finally, attacking the prey.

In order to mathematically model the encircling behavior, the following formulations are proposed:

$$Y_k\left(t+1\right)=Y_p\left(t\right)-A\left|CY_p\left(t\right)-Y_k\left(t\right)\right| \tag{34}$$

where t is the current iteration, Y_k, is the kth grey wolf position vector, Y_p is the position vector of the prey. A and C are coefficient vectors and calculated as follows:

$$A=2a.r_1-a. \tag{35}$$

$$C=2.r_2. \tag{36}$$

in which r_1 and r_2 are random vectors in $\left[0.1\right]$, components of a are linearly decreased from 2 to 0 over the course of iterations.

In order to mathematically simulate the hunting behavior of grey wolves to find the location of the optimum (prey), the first three best solutions obtained (Y_α, Y_β and Y_δ) are saved and others including omegas should update their positions according to the positions of the best search agents, based on the following formulas:

$$Y_1 = Y_\alpha - A_1 \left| C_1 Y_\alpha - Y \right|, \tag{37}$$

$$Y_2 = Y_\beta - A_2 \left| C_2 Y_\beta - Y \right|, \tag{38}$$

$$Y_3 = Y_\delta - A_1 \left| C_3 Y_\delta - Y \right|, \tag{39}$$

$$Y\left(t+1\right) = \frac{Y_1 + Y_2 + Y_3}{3}. \tag{40}$$

When the prey stops moving, the grey wolves attack it and finish the hunting. As mentioned before, the values of the components of a are decreased linearly over the course of iterations, and based on equation (35) the fluctuation range of A is also decreased, and so approaching the prey is mathematically modeled. When random values of A are in [-1,1] or $\left|A\right| \le 1$, the next position of a search agent can be in any position between its current position and the position of the prey and it forces the wolves to attack the prey.

In order to mathematically model divergence, the random values of A with $|A|>1$ oblige the search agent to diverge from the prey to hopefully find a better prey. This highlights exploration and allows the GWO algorithm to search globally.

Another component of GWO that helps exploration is C. This component emphasizes (when ($|C|>1$) or deemphasizes (when $|C|<1$) the effect of prey position in defining the distance in Equation (34).

The pseudo code of the GWO algorithm is as follows:

Initialize the grey wolf population $Y_i,\ i \in \{1, 2, \ldots, popsize\}$,
Initialize $a, A, andC$,
Calculate the fitness of each search agent,
Y_α *=the best search agent,*
Y_β *=the second best search agent,*
Y_β *=the third best search agent,*
While *(t<Max number of iterations)*
    for *each agent*
    Update the position of the current search agent by equation (40)
    end for

Update $a, A, and C,$ *Calculate the fitness of all search agents* *Update* $Y_\alpha, Y_\beta, Y_\delta$ $t = t + 1$ ***end while*** *Return* Y_α

5-2-3. The particle swarm optimization (PSO)

PSO described by Eberhart and Kennedy [46] is a swarm-based algorithm which is inspired by the social behavior of bird flocking and fish schooling. Each particle in PSO has its own velocity, position vector and a fitness value determined by the objective function. Additionally, PSO has memory and the previous particle information is reflected in it and is used in general backpropagation processes. Hence, a continuous progression towards the global optimum point is provided. The following equations show how the velocity and new position of each particle is calculated in every iteration:

$$V(t+1) = \omega V(t) + c_1 r_1 \left(Y_{pbest}(t) - Y(t)\right) + c_2 r_2 \left(Y_{gbest}(t) - Y(t)\right) \tag{41}$$

$$Y(t+1) = Y(t) + V(t+1) \tag{42}$$

Where ω is inertia weight, $V(t)$ is the former velocity, c_1 and c_2 are acceleration constants which control how the particles approach the local best and global best position, respectively. r_1 and r_2 are random values within $[0,1]$. $Y(t)$ is the former position, $Y_{pbest}(t)$ is the best position of the particle and $Y_{gbest}(t)$ is the global best position of the whole swarm.

5-2-4. The genetic algorithm (GA)

The genetic algorithm (GA) is the evolutionary methodology introduced by Holland in 1975, based on natural selection and genetics. A GA works on an initial population; it selects parents, applies crossover and mutation operators and evaluates the children. The goal is to successively produce better solutions by selecting the better ones among existing solutions with a higher chance for recombination. For the description about GA used in our study we refer readers to Zoraghi et al. [37].

5-3. The inside search :The genetic algorithm (GA)

This section describes another GA which is applied in the second level of hybrid algorithms to discover the best ordering policy of the schedule.

First QDPMO solution part $X = \left[X_{qj} \right]_{Q \times U}$; which was explained in Sect. 5.1; is encoded to a chromosome as shown in Figure 2 (as mentioned before X_{q1} is equal 1 so it is omitted from the chromosome).

This algorithm starts with a random population of initial chromosomes and continues to produce new generations using the crossover and mutation operators.

In each generation, a set of chromosomes is generated through a recombination process. For this purpose, the roulette wheel selection pattern is used to select parents and two kinds of crossover operators are applied with the same probability to produce offsprings (Figure 3).

In the one-point crossover, a number *r* is randomly generated form the range of $\{1,2,\ldots, Q(U-1)\}$. The data for each offspring is a mixed combination of one parent (before *r*) and the other parent (after *r*). Figure 3a shows a simple example of this operator.

Also, the two-point crossover in which two points $r_1, r_2 \in \{1, 2, \ldots, Q(U-1)\}$ of the parents are randomly selected such that $r_1 < r_2$ is employed. The columns between these two points of parents are exchanged and two offspring are created. Figure 3b shows a simple example of this operator.

In addition, a single point mutation is applied with a certain probability which changes a 1 to 0, and vice versa.

6. Computational experiments

In this step, we implement the model using hybrid algorithms to show their performance and applicability in practice. To do so, first a parameter tuning procedure is utilized based on Taguchi method to provide robust solutions. Next, the developed mathematical model is also solved using the BARON solver in GAMS and the obtained results are compared with those obtained from the hybrid algorithms. The comparisons were carried out on problems in different sizes to show the performance of the meta-heuristics. All computations were performed on a Core (TM) i7-2600k PC with 3.4 GHz CPU speed and 8 GB of RAM. We used MATLAB Software to code proposed meta-heuristics.

6-1. Parameter tuning

The performance of meta-heuristic algorithms is highly related to the values of their structural parameters (factors). In this regard, there are different ways for tuning the parameters, in order to improve the robustness of the algorithms. In this study, we applied the

Taguchi method, which can consider a large number of decision variables with a small number of experiments [47].

In Taguchi method the factors are separated into two main categories: controllable and noise factors. However, noise factors don't have direct effect on results, their removal is impractical and often impossible; The Taguchi method tries to minimize the effect of noise and determine the optimal level of significant controllable factors to achieve robustness.

Taguchi method uses a special set of standard arrays called orthogonal arrays. The orthogonal arrays determine how to conduct the minimal number of experiments which could give the full information of all the factors that affect the response variables. An appropriate orthogonal array is determined based on the number of factors and their levels. In the orthogonal array, the columns correspond to the parameters, the entries in the columns correspond to the levels of the factors and the rows correspond to the experiments.

This method uses the signal-to-noise (S/N) ratio to reflect the extant variation in the response variable, where a higher S/N ratio means smaller variation. Equation (43) shows the calculation for S/N ratio in a minimization (the smaller-the-better type) problem:

$$SN_j = -10log\left(\frac{\sum_{i=1}^{n} {y_{ij}}^2}{n}\right) \tag{43}$$

where y_{ij} is the response value of ith observation at trial j of the orthogonal array and n is the number of observations in each trial.

In this study, we assumed the relative deviation percentage for a single problem i at trial j (RDP_{ij}) of the orthogonal array as the response value (y_{ij}), defined as follows:

$$RDP_{ij} = \frac{Z_{ij}^* - Min\left(Z_{ik}^*\right)}{Min\left(Z_{ik}^*\right)} \tag{44}$$

Where Z_{ij}^* is the objective value for the problem i at the trial j of the orthogonal array.

We tuned the hybrid algorithms' parameters in two steps:

Step1: Tuning of the inside search algorithm (GA)

Inside search uses the schedule of activities and their modes generated by outside search to solve the material ordering problem. So, we assumed some problems with different sizes and their schedules and modes of activities and used Taguchi method to tune the parameters of inside search algorithm (GA).

Step2: Tuning of the outside search algorithms

In this section, we describe the Taguchi method for tuning the parameters influencing the performance of the inside algorithm, each at three levels. In Table 2, we presented four parameters affecting the performances of the GA and their different levels. Based on Taguchi method, at first, an orthogonal array, $L9\left(3^4\right)$, is selected as the fittest design. Then the experiments are performed for a set of 5 problems from the PSPLIB library with 10, 16, 20, 30, and 60 activities. However, other essential assumptions such as holding costs, ordering costs, material prices and in addition a schedule for activities and their modes for each problem are added to complete the matter.

The data in Table 3 show the orthogonal array of parameters for the Inside GA and observations' results including RDP's mean and S/N ratio in each trial.

In Figure 4, there is a S/N ratio plot resulting from the Inside GA and its best parameter levels are shown in Table 4.

The same approach was carried out to tune parameters for outside search algorithms and the optimum levels are illustrated in Tables 5 - 8.

Performance evaluation of the hybrid algorithms uses these optimal parameter values.

6-2. Performance analysis of the hybrid algorithms

Initially, the performance of the aforementioned meta-heuristics to solve the MRCPSP–QDPMO problem is considered for a typical instance, and the results are compared with that of BARON solver.

Consider a project including 16 activities with an AON precedence network as shown in figure 5. Each activity has three possible execution modes.

Table 9 presents the list of activities with their durations and required renewable and non-renewable resources separately for each mode. Project's due date is at the end of time unit 25. Delay penalty is 10,000 per time unit and earliness bonus is 15,000 $ per time unit.

Table 10 presents the costs of materials and Table 11 shows the prices for each of them in different break points.

Table 12 includes the available number of each renewable resources and its cost per time unit.

Tables 13-15 shows the results after implementing the meta-huristics and BARON.

The results in table 13 shows COA-GA and GWO-GA outperformed BARON significantly. For the purpose of evaluating the performance of four solving algorithms, we used the project generator software (PROGEN) developed by Kolisch and Sprecher in 1996 which resulted in 270 numerical MRCPSP problems having 10, 12, 14, 16, 18, 20, 30, 60 and 90 non-dummy activities; each one can be executed in one of three modes. As illustrated in Table 16, the problems differ with respect to the number of renewable and nonrenewable resources. For each of 27 classes of problems, 10 instances were generated; resulting in 270 test problems. To solve each problem, every hybrid algorithm run for a constant CPU time based on its size. Table 16 demonstrates the mean of the relative deviation percentage from the best-known solution ($\overline{RDP}$), resulted from every four algorithms for each class. In this table, the algorithms are compared using $\overline{RDP}$. RDP of 0.00% means the best result or the minimum total project cost.

According to Figure 6 we can strongly claim that the proposed hybrid COA-GA and GWO-GA algorithms are quite effective to solve MRCPSP–QDPMO problems, evidently outperforming PSO-GA and GA-GA algorithms.

To compare the COA-GA and GWO-GA algorithms, a paired sample T-test is conducted to test the equality of the $\overline{RDP}$ means obtained by the two algorithms (H_0) against the COA-GA outperforms GWO-GA or the $\overline{RDP}$ mean of COA-GA is less than the $\overline{RDP}$ mean of GWO-GA (H_1). The two hypotheses for this test are:

$$\begin{cases} H_0 : \mu_d = 0, \\ H_1 : \mu_d > 0. \end{cases}$$

In which d is the difference between two paired $\overline{RDP}$ ($\overline{RDP}_{GWO-GA} - \overline{RDP}_{COA-GA}$).

The test statistic (2.88) denotes that null hypothesis is rejected at 95% confidence level, indicating that COA-GA algorithm significantly outperforms GWO-GA.

7- Sensitivity analysis

This section seeks to determine the effect of different features of the problem on the total project cost. To deal with this issue, again we used the typical instance illustrated in section 6-2 to assess the sensitivity of the objective function in respect to changing five factors including holding cost, ordering cost, material price, penalty, and bonus. Figures 7-11[2] shows the sensitivity of objective function to each factor based on changing its value by a different percentage compared to its base value, while the other factors remain unaltered. Figures 7-11 show the results after solving the problems with the best metaheuristic, COA-GA.

According to the results in Figure 7, increasing the holding costs in the presence of fixed other costs leads to smaller order sizes to reduce the material inventories and projects' holding costs.

As shown in Figure 8, increasing the ordering costs results in reducing the number of orders and increasing their amount, and as a result bigger chance to use better discounts. The increase in project's holding cost actually represents more inventories due to the increase in orders amounts.

As Figure 9 shows, the higher material prices, leads to increase in the amount of orders, in order to make use of more discounts. In such situations, higher holding costs are acceptable.

The impact of the penalty factor can be seen in its elimination. When there is no tardiness penalty in this project, the activities are carried out in modes with lower costs and longer durations, thus it is obvious that the project finish time increases (Figure 10).

According to Figure 11, with low earliness bonus, selection of activity modes with lower duration and more cost is not justifiable, simply because the bonus can't compensate for the increased project cost.

7- Conclusions

In this study we investigated a more realistic class of project scheduling problems, a combination of Multi-mode Resource-Constraint Project Scheduling Problem (MRCPSP), under bonus–penalty policies and Quantity Discount Problem in Material Ordering (QDPMO) under an all unit discount scheme, which was called MRCPS- QDPMO problem. The model was mathematically formulated with the objective of minimizing the total cost of project including costs of renewable resources, non-renewable resources (ordering, holding and purchasing) and the penalty or bonus resulting from the project's tardiness or earliness.

By proving a theorem, we found out that an important property of the optimum solution, is the fact that activities's start times are the only choice for ordering points. This property reduces the search space significantly compared to previous studies.

---

[2] Logarithmic diagrams are used to represent all the cases in a graph according to different cost scales.

Since this problem is NP-hard, four hybrid algorithms (COA-GA, GWO-GA, PSO-GA and GA-GA) incorporating an outside and an inside search were presented to solve the model.

In fact, the outside search (COA, GWO, PSO and GA) focuses on finding solutions for the MRCPSP to present the schedules and the inside search (GA) which is the same among all proposed hybrid algorithms focused on solving Quantity Discount Problem in Material Ordering (QDPMO) to discover best ordering plan for the generated schedules.

After tuning the parameters of the algorithms by the Taguchi method, for solving a typical instance, the algorithms were applied and their results were compared with solution obtained from BARON solver of GAMS software. The comparison showed COA-GA and GWO-GA outperformed BARON. Lastly, hybrid metaheuristics were compared with each other in order

to find the best solutions in a fixed time. In this regard, the developed algorithms were tested on a set of problems generated by the PROGEN with some assumptions, and observed that the hybrid COA-GA outperforms others.

References

1. Elmaghraby, S. E. "Activity networks: Project planning and control by network models", *John Wiley & Sons* (1977).
2. Słowiński, R. "Two approaches to problems of resource allocation among project activities—a comparative study", *Journal of the Operational Research Society*, **31**(8), pp. 711-723 (1980).
3. Talbot, F. B. "Resource-constrained project scheduling with time-resource tradeoffs: The non-preemptive case", *Management science*, **28**(10), pp. 1197-1210 (1982).
4. Patterson JH, Słowinski R, Talbot FB, Weglarz J. "An algorithm for a general class of precedence and resource constrained scheduling problems", *Advances in project scheduling. Elsevier, Amsterdam*, pp. 3–28 (1989)
5. Drexl, A., & Gruenewald, J. "Nonpreemptive multi-mode resource-constrained project scheduling", *IIE transactions*, **25**(5), pp. 74-81 (1993).
6. Speranza, M. G., & Vercellis, C. "Hierarchical models for multi-project planning and scheduling", *European Journal of Operational Research*, 64(2), pp. 312-325 (1993).

7. Hartmann, S., & Briskorn, D. “A survey of variants and extensions of the resource-constrained project scheduling problem”, *European Journal of Operational Research*, **207**, pp. 1–14 (2010).

8. Sprecher, A., Hartmann, S., & Drexl, A. “An exact algorithm for project scheduling with multiple modes”, *Operations-Research-Spektrum*, **19**(3), pp. 195-203 (1997).

9. Demeulemeester, E. L., & Herroelen, W. S. “An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem”, *European Journal of Operational Research*, **90**(2), pp. 334-348 (1996).

10. Boctor F.F. “Heuristics for scheduling projects with resource restrictions and several resource-duration modes”, *The international journal of production research*, **31**(11), pp. 2547–2558 (1993)

11. Boctor F.F. “A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes”, *European Journal of Operational Research* **, 90**(2), pp. 349–361 (1996)

12. Kolisch R, Drexl A. “Local search for non-preemptive multi-mode resource-constrained project scheduling*”, IIE transactions*, **29**(11), pp. 987–999 (1997)

13. Sprecher A, Drexl A. “Solving multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm”, *European Journal of Operational Research*, **107**(2), pp. 431–450 (1998).

14. Bouleimen, K., & Lecocq, H. A. “new efficient simulated annealing algorithm for resource constrained scheduling problem” Technical report, *service de Robotique et automatisation, university de Liege*. pp. 1-10 (1998).

15. Hartmann, S. “Project scheduling with multiple modes: a genetic algorithm”, *Annals of Operations Research*, **102**(1-4), pp. 111-135. (2001).

16. Józefowska, J., Mika, M., Różycki, R., Waligóra, G., & Węglarz, J. “Simulated annealing for multi-mode resource-constrained project scheduling”, *Annals of Operations Research*, **102**(1-4), pp. 137-155 (2001).

17. Alcaraz J, Maroto C, Ruiz R. “Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms”, *Journal of the Operational Research*, **54**(6), pp. 614–626 (2003)

18. Mika M, Waligóra G, Weglarz J. “Simulated annealing and Tabu-search for multi-mode resource constrained project scheduling with positive discounted cash flows and different payment models”, *European Journal of Operational Research*, **164** (3), pp. 639–668 (2005)

19. Zhang H, Tam CM, Li H. “Multi-mode project scheduling based on particle swarm optimization”, *Computer‐Aided Civil and Infrastructure Engineering*, **21**(2), pp. 93–103 (2006).

20. Jarboui B, Damak N, Siarry P, Rebai A. “A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems”, *Applied Mathematics and Computation***, 195**(1), pp. 299–308 (2008)

21. Van Peteghem, V., & Vanhoucke, M. “An artificial immune system for the multi-mode resource-constrained project scheduling problem” In European Conference on Evolutionary Computation in Combinatorial Optimization, Springer, Berlin, Heidelberg. pp. 85-96, (2009, April).

22. Van Peteghem V, Vanhoucke M. “Genetic algorithm for the preemptive and non-preemptive multimode resource-constrained project scheduling problem”, *European Journal of Operational Research***, 201**(2), pp. 409–418 (2010).

23. Barrios, A., Ballestín, F., & Valls, V. “A double genetic algorithm for the MRCPSP/max” *Computers & Operations Research*, **38**(1), pp. 33-43 (2011).

24. Khalilzadeh, M., Kianfar, F., Shirzadeh Chaleshtari, A., Shadrokh, S., & Ranjbar, M. “A modified PSO algorithm for minimizing the total costs of resources in MRCPSP”, *Mathematical Problems in Engineering*, (2012).

25. Wang, L., & Fang, C. “An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem”, *Computers & Operations Research*, **39**(2), pp. 449-460 (2012).

26. Li, H., & Zhang, H. “Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints”, *Automation in construction*, **35**, pp. 431-438 (2013).

27. Messelis, T., & De Causmaecker, P. “An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem”, *European Journal of Operational Research*, **233**(3), pp. 511-528 (2014).

28. Aquilano N.J., Smith D.E. “A formal set of algorithms for project scheduling with critical path method material requirements planning. *Journal of Operations Management,* **1**(2):_pp. 57–67 (1980).

29. Smith-Daniels, D. E., & Aquilano, N. J. “Constrained resource project scheduling subject to material constraints”, *Journal of Operations Management*, **4**(4), 369-387 (1984).

30. Smith-Daniels, D. E., & Smith-Daniels, V. L. “Optimal project scheduling with materials ordering”, *IIE transactions*, **19**(2), pp. 122-129 (1987).

31. Smith-Daniels, D. E., & Smith-Daniels, V. L. “Maximizing the net present value of a project subject to materials and capital constraints”, *Journal of Operations Management*, **7**(1-2), pp. 33-45 (1987).

32. Erbasi, A., & Sepil, C. “A modified heuristic procedure for materials management in project networks”, *International Journal of Industrial Engineering: Theory, Applications and Practice*, **6**(2), pp. 132-140 (1999).

33. Dodin, B., & Elimam, A. A. “Integrated project scheduling and material planning with variable activity duration and rewards”, *Iie Transactions*, **33**(11), pp. 1005-1018 (2001).

34. Sajadieh, M. S., Shadrokh, S., & Hassanzadeh, F. “Concurrent project scheduling and material planning: A genetic algorithm approach”, *Scientia Iranica. Transaction E, Industrial Engineering*, **16**(2), pp. 91-99 (2009).

35. Tabrizi, B. H., & Ghaderi, S. F. “An Integrated Mixed-Integer Programming Model to Address Concurrent Project Scheduling and Material Ordering”, *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, **9** (2015).

36. Tabrizi, B. H., & Ghaderi, S. F. “A robust bi-objective model for concurrent planning of project scheduling and material procurement”, *Computers & Industrial Engineering*, **98**, pp. 11-29 (2016).

37. Zoraghi, N., Shahsavar, A., Abbasi, B., & Van Peteghem, V. “Multi-mode resource-constrained project scheduling problem with material ordering under bonus–penalty policies”, *Top*, **25**(1), pp. 49-79 (2017).

38. Shahsavar, A., Zoraghi, N., & Abbasi, B. “Integration of resource investment problem with quantity discount problem in material ordering for minimizing resource costs of projects”, *Operational Research*, **18**(2), pp. 315-342 (2018).

39. Zoraghi, N., Shahsavar, A., & Niaki, S. T. A. “A hybrid project scheduling and material ordering problem: Modeling and solution algorithms”, *Applied Soft Computing*, **58**, pp. 700-713 (2017).

40. Tabrizi, B. H. “Integrated planning of project scheduling and material procurement considering the environmental impacts”, *Computers & Industrial Engineering*, **120**, pp. 103-115 (2018).

41. Kolisch, R. “Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation”, *European Journal of Operational Research*, **90**(2), pp. 320-333 (1996).

42. Pierezan, J. and Coalho, L. “Coyote Optimization Algorithm: A new metaheuristic for global optimization problems”, *IEEE Congress on Evolutionary Computation (IEEE CEC), 2018 IEEE*, pp. 2633-2640 (2018).

43. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). “Grey wolf optimizer”, *Advances in engineering software*, **69**, pp. 46-61.

44. Long, W., & Xu, S. “A novel grey wolf optimizer for global optimization problems”, *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC),* pp. 1266-1270 (2016).

45. Muro, C., Escobedo, R., Spector, L., & Coppinger, R. P. “Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations”, *Behavioural processes*, **88**(3), pp. 192-197 (2011).

46. Eberhart, R., & Kennedy, J. “A new optimizer using particle swarm theory”, In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43 (1995).

47. Taguchi, G. “Introduction to quality engineering”, *White Plains, Asian Productivity Organization, NewYork* (1986).

48. Eberhart, R., & Kennedy, J. “A new optimizer using particle swarm theory”, In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.

Figure captions

Figure 1- An example of the MRCPSP solution part (Zoraghi et al. (2016a))

Figure 2- Chromosome representation of the QDPMO solution part X

Figure 3- a) the one-point crossover, b) the two-point crossover

Figure 4- Main effects plot for the inside GA factor levels. SN = signal-to-noise ratio.

Figure 5- AON precedence network of the typical instance

Figure 6- Comparison of the algorithms based on the mean of the relative deviation percentage

Figure 7- The costs and the finish time of the problem found COA-GA with different value of holding costs

Figure 8- The costs and the finish time of the problem found COA-GA with different value of ordering costs

Figure 9- The costs and the finish time of the problem found COA-GA with different value of material prices

Figure 10- The costs and the finish time of the problem found COA-GA with different value of penalty

Figure 11- The costs and the finish time of the problem found COA-GA with different value of bonus

Table captions

Table 1- Summary of notation.

Table 2- Inside GA factors and their levels

Table 3-The orthogonal array for parameters of the inside GA and results

Table 4. Optimum factor levels and values for inside GA algorithm

Table 5- Optimum factor levels and values for COA algorithm

Table 6- Optimum factor levels and values for GWO algorithm

Table 7- Optimum factor levels and values for PSO algorithm

Table 8 - Optimum factor levels and values for GA algorithm

Table 9- the list of activities with their durations and required renewable and non-renewable resources with respect to each mode

Table 10- The costs of materials

Table 11- The prices of materials in the different break points.

Table 12- The cost and maximum level of each renewable resources

Table 13- The costs and the finish time of the problem found by the metahuristics and BARON

Table 14- The schedules of the problem found by the metahuristics and BARON

Table 15- The procurement plan of the problem obtained from the metahuristics and BARON

Table 16- Comparison of the metaheuristics based on $\overline{RDP}$

0.097	0.546	0.964	0.970	0.485
0.278	0.957	0.157	0.557	0.800

An example of solution representation (Y)

Activity	A_1	A_2	A_3	A_5	A_4
Mode	1	3	1	3	2

Activity sequence list and activity mode list representation

Figure 1

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Material type 1					Material type 2					Material type 3				
1	0	0	1	0	1	0	1	0	1	0	1	0	1	1

Figure 2

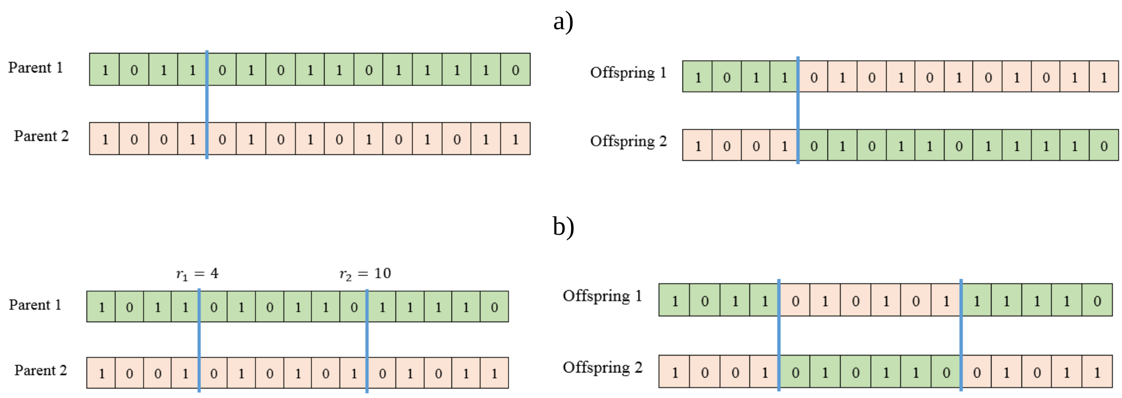

Figure 3

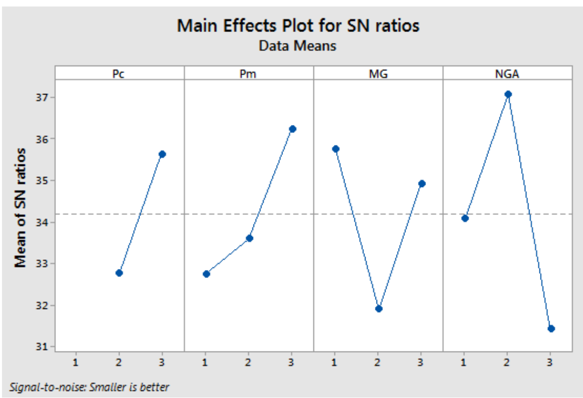

Figure 4

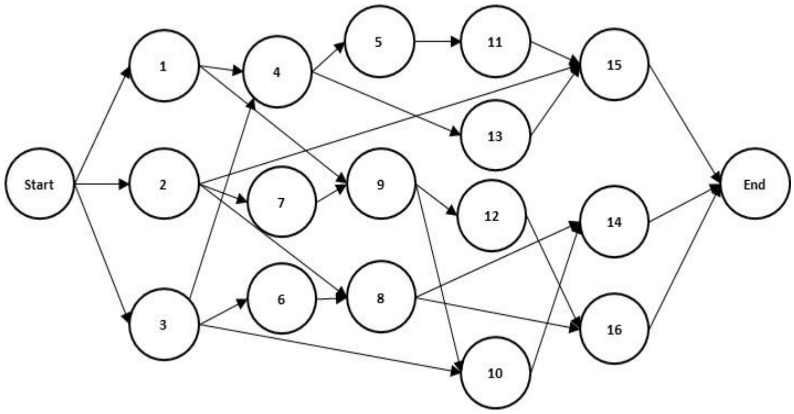

Figure 5

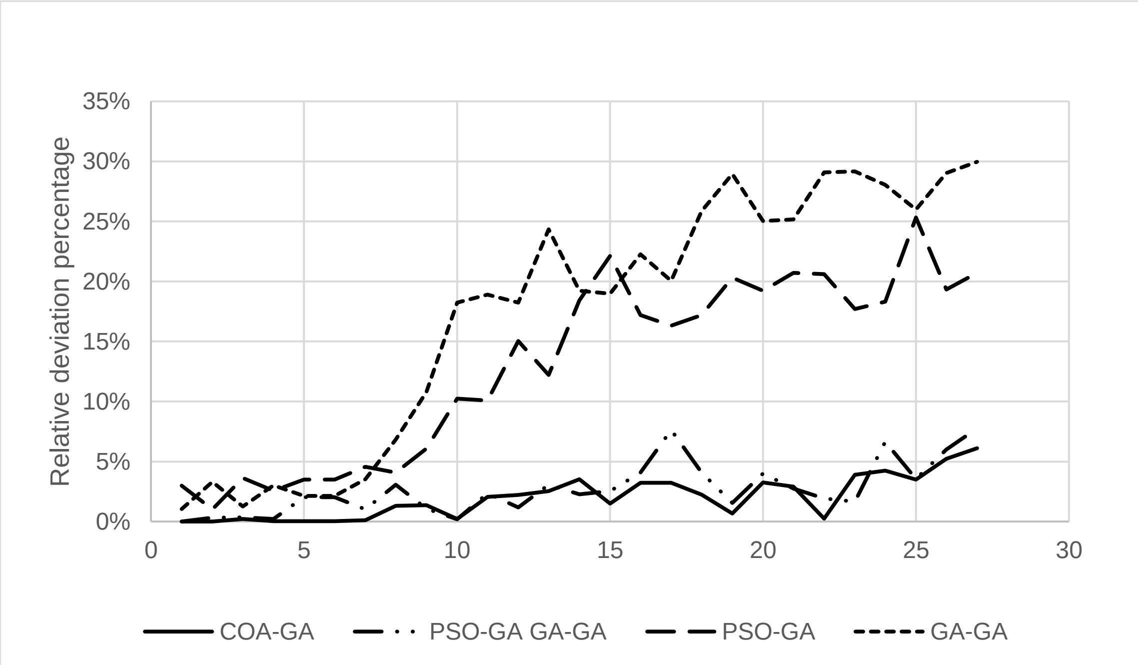

Figure 6

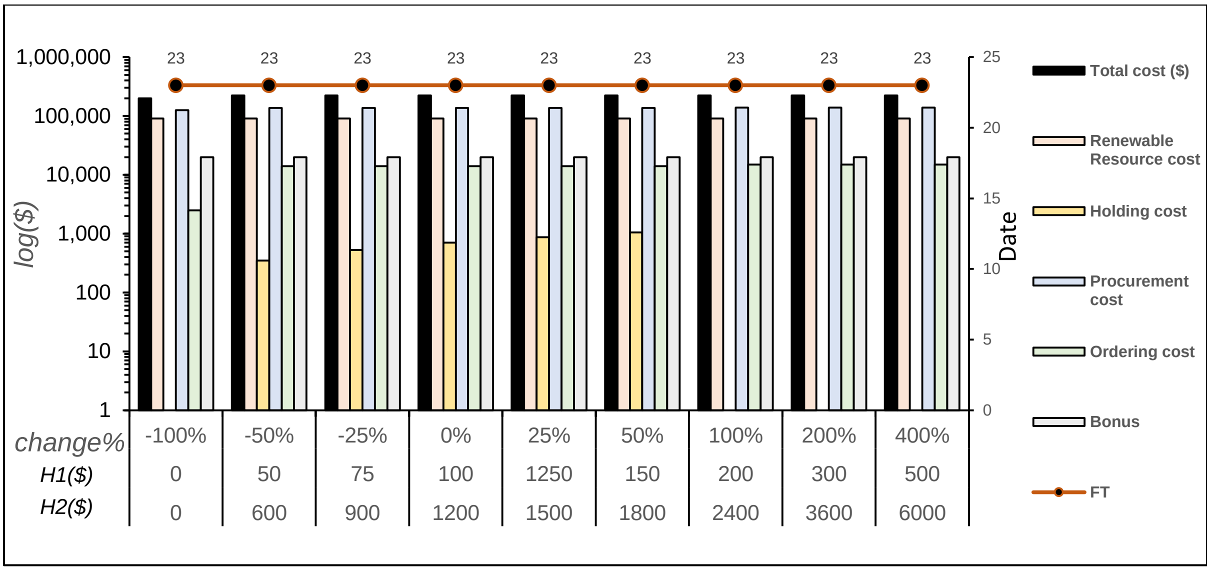

Figure 7

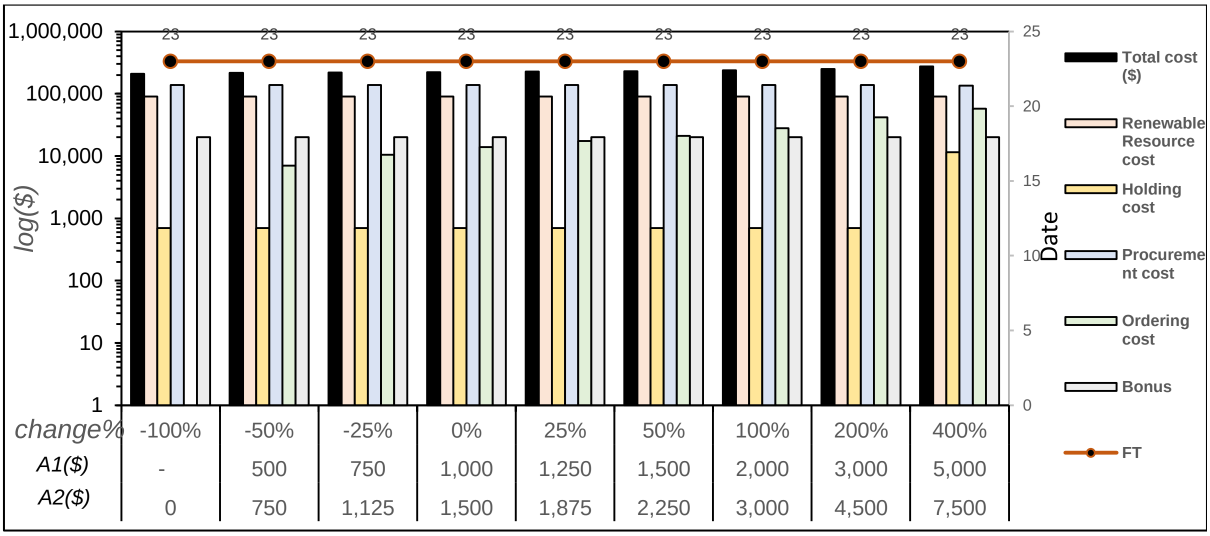

Figure 8

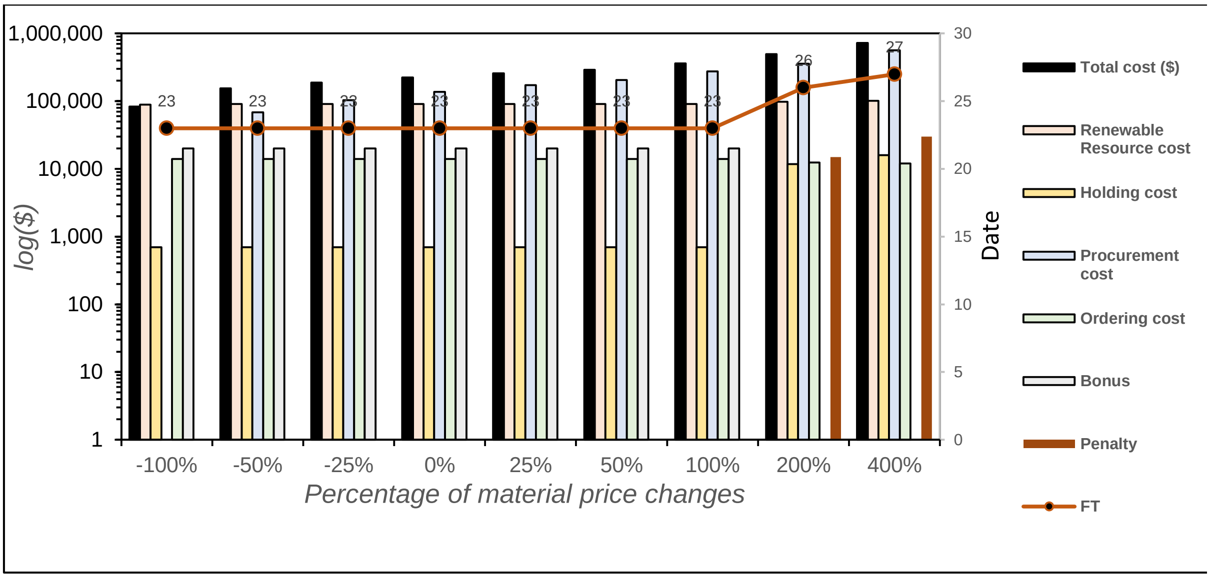

Figure 9

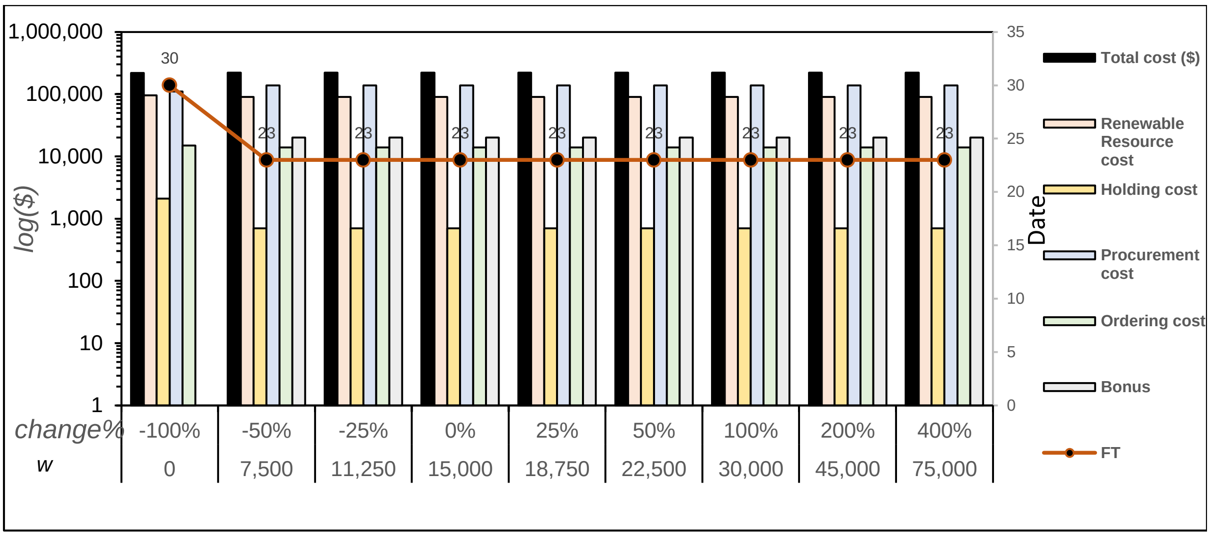

Figure 10

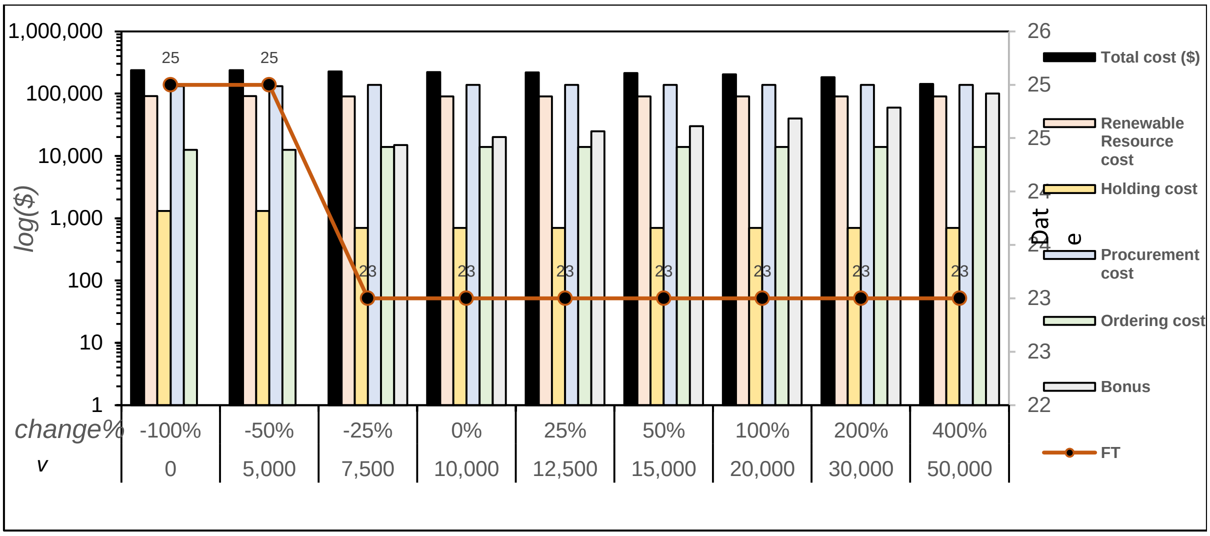
1,000,000
100,000
10,000
1,000
100
10
1
log($)
26
25
25
24
24
23
23
22
Date
25
25
23
23
23
23
23
23
23
change%
-100%
-50%
-25%
0%
25%
50%
100%
200%
400%
v
0
5,000
7,500
10,000
12,500
15,000
20,000
30,000
50,000
Total cost ($)
Renewable Resource cost
Holding cost
Procurement cost
Ordering cost
Bonus
FT

Figure 11

Table 1

Indices	
$i, j \in \{0,1,\ldots,n\}$	Index of project activities
$t, b \in \{1,\ldots,T\}$	Index of time periods
$p \in \{1,2,\ldots,P\}$	Index of renewable resources
$q \in \{1,2,\ldots,Q\}$	Index of materials
$m \in \{1,\ldots,M_i\}$	Index of modes of activity for the i^{th} activity
$k \in \{1,\ldots,K_q\}$	Index of price breakpoint for material q
Parameters	
d_{im}	Duration of activity i executed in mode m
$pred_{ij} = \begin{cases} 1 \\ 0 \end{cases}$	If activity i is the predecessor of activity j Otherwise
res_{pt}	Number of units of renewable resource p available in period t
r_{pim}	Number of units of renewable resource p required by activity i executed in mode m
a_{qim}	Number of units of non-renewable resource q required by activity i executed in mode m
cr_p	Cost of renewable resource p per period

H_q	Inventory holding cost per unit of material q per period
A_q	Ordering cost per replenishment of material q
$BigM$	Big number
$P_q = \begin{cases} P_{q1} & 1 \le Q_{qt} \le O_{q1} \\ P_{q2} & O_{q1}+1 \le Q_{qt} \le O_{q2} \\ \vdots \quad \vdots & \vdots \\ P_{qK} & O_{qK-1}+1 \le Q_{qt} \end{cases}$	Purchasing cost of material q under AUD scheme
ϖ	Penalty per period of delay
ϑ	Bonus per period of early completion
D	Due date of the project
T	Planning horizon
π	Project income at the end of the project
Binary variables	
$x_{imt} = \begin{cases} 1 \\ 0 \end{cases}$	If activity i is performed in mode m and finished at time period t Otherwise
$y = \begin{cases} 1 \\ 0 \end{cases}$	If the project is finished after due date Otherwise
$\lambda_{qt} = \begin{cases} 1 \\ 0 \end{cases}$	If material q is ordered in period t Otherwise
$\mu_{tqk} = \begin{cases} 1 \\ 0 \end{cases}$	If material q is purchased at price breakpoint k in period t Otherwise

Continuous variables	
FT	Makespan
Q_{qt}	Ordered quantity of material q in period t
I_{qt}	Inventory level of material q at the end of period t

Table 3

Trial	P_c	P_m	MG	NGA	$\overline{RDP}$	S/N
1	1	1	1	1	0.038	28.312
2	1	2	2	2	0.023	32.670
3	1	3	3	3	0.014	36.002
4	2	1	2	3	0.014	35.330
5	2	2	3	1	0.019	34.507
6	2	3	1	2	0.017	30.682
7	3	1	3	2	0.006	39.702
8	3	2	1	3	0.012	37.007
9	3	3	2	1	0.011	37.728

Table 4

Factors	*Inside search*			
	P_c	P_m	MG	NGA
Optimal levels	0.9	0.3	10	150

Table 5

Factors	$Iter$	N_p	N_c
Optimal levels	200	15	14

Table 6

Factors	Iter	SearchAgents_no
Optimal levels	200	30

Table 7

Factors	$Iter$	Swarm size	c_1	c_2	ω
Optimal levels	400	30	2.9	1.9	0.65

Table

Factors	P_c	P_m	MG	NGA
Optimal levels	0.9	0.3	20	200

Table 9

Activity	Mode1					Mode2					Mode3				
	Duration	Required resources				Duration	Required resources				Duration	Required resources			
		R1	R2	NR1	NR2		R1	R2	NR1	NR2		R1	R2	NR1	NR2
1	1	9	3	10	5	2	5	2	12	3	3	3	1	12	3
2	2	6	0	0	3	4	4	0	0	3	5	3	0	0	2
3	2	5	10	6	0	3	3	5	5	0	3	6	4	4	0
4	2	10	8	0	4	4	8	8	0	4	6	6	8	0	3
5	8	9	2	8	2	10	8	1	7	5	10	5	2	8	4
6	1	0	5	8	0	3	0	5	4	0	5	0	4	3	0
7	7	7	5	0	5	9	5	3	0	4	10	10	1	0	2
8	1	7	9	7	0	2	5	5	3	0	4	2	2	3	0
9	5	5	6	4	0	7	5	3	2	1	10	3	2	3	1
10	4	8	0	0	8	6	6	0	0	6	8	5	0	0	5
11	2	10	0	0	8	2	0	8	0	3	5	12	0	0	6
12	5	2	10	7	0	9	6	5	4	1	9	5	4	6	3
13	6	8	8	4	1	8	12	5	5	2	10	10	4	15	0
14	2	7	8	6	0	3	5	4	10	0	5	3	3	3	0
15	5	6	7	0	3	6	3	5	5	3	9	6	2	15	3
16	2	5	2	5	1	4	3	1	5	0	7	5	0	2	1

R: Renewable resource; NR: Non-Renewable resource

Table 10.

	Non-renewable resources	
	1	2
Holding cost ($) per time unit	100	1,200
Ordering cost ($)	1,000	1,500

Table 11.

Non-renewable resources			
1		2	
Range	Price ($)	Range	Price ($)
$1 \le O \le 10$	700	$1 \le O \le 5$	3,200
$11 \le O \le 20$	650	$6 \le O \le 10$	2,800
$21 \le O \le 30$	630	$11 \le O$	2,500
$31 \le O \le 40$	600		
$41 \le O$	590		

O: Order quantity

Table 12.

	Renewable resources	
	1	2
$/time unit	100	200
Max level	19	15

Table 13.

Method	BARON	COA-GA	GWO-GA	PSO-GA	GA-GA
Total cost ($)	227,650	223,000	223,000	233,400	235,900
FT	23	23	23	24	24
Renewable Resource cost	93,500	90,600	90,600	92,500	93,400
Holding cost	7,300	700	700	1,100	5,100
Ordering cost	16,000	14,000	14,000	14,000	15,000
Procurement cost	130,850	137,700	137,700	135,800	132,400
Penalty (Bonus)	(20,000)	(20,000)	(20,000)	(10,000)	(10,000)

Table 14

Activity	BARON		COA-GA		GWO-GA		PSO-GA		GA-GA	
	Start time	Mode	Start time	Mode	Start time	Mode	Start time	Mode	Start time	Mode
1	0	2	0	2	0	2	0	2	0	2
2	0	1	0	1	0	1	0	1	0	1
3	0	1	0	1	0	1	0	1	0	1
4	2	1	2	1	2	1	2	1	2	1
5	4	1	4	3	4	3	4	1	10	1
6	4	2	10	1	10	1	4	1	10	1
7	2	2	2	2	2	2	2	1	2	1
8	17	1	11	2	11	2	5	3	11	3
9	11	1	11	1	11	1	12	1	9	1
10	17	1	16	1	16	1	17	1	14	3
11	15	2	14	2	14	2	15	2	18	1
12	16	1	16	1	16	1	17	1	14	1
13	10	1	4	1	4	1	9	1	4	1
14	21	1	21	1	21	1	22	1	22	1
15	17	2	16	2	16	2	17	2	20	1
16	21	1	21	1	21	1	22	1	20	1

Table 15

Date	BARON		COA-GA		GWO-GA		PSO-GA		GA-GA	
	Material1	Material2	Material1	Material2	Material1	Material2	Material1	Material2	Material1	Material2
1	18	6	18	6	18	6	18	6	18	6
2	0	0	0	0	0	0	0	0	0	0
3	0	10	0	8	0	8	0	8	0	11
4	3	0	0	0	0	0	0	0	0	0
5	9	0	12	5	12	5	12	5	19	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	4	1
11	8	1	15	0	15	0	8	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	4	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	3	0	3	0	3	0	0
16	7	3	0	0	0	0	0	0	0	3
17	0	0	12	11	12	11	19	9	0	0
18	12	11	0	0	0	0	0	0	12	11
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	11	0	0	0	0	0	0	0	0	0
22	0	1	11	1	11	1	0	0	0	0
23							11	1	11	1

Table 16

N	RR	NR	CPU time (s)	COA-GA	GWO-GA	PSO-GA	GA-GA
10	1	1	60	0.00%	0.00%	3.00%	1.03%
	2	2	100	0.00%	0.31%	1.02%	3.32%
	3	3	120	0.21%	0.35%	3.63%	1.25%
12	1	1	100	0.02%	0.22%	2.55%	3.01%
	2	2	120	0.03%	2.03%	3.50%	2.13%
	3	3	150	0.03%	2.03%	3.50%	2.13%
14	1	1	120	0.12%	1.02%	4.55%	3.53%
	2	2	150	1.32%	3.07%	4.08%	6.83%
	3	3	200	1.35%	1.07%	6.08%	10.80%
16	1	1	150	0.21%	0.18%	10.23%	18.23%
	2	2	200	2.06%	2.34%	10.07%	18.91%
	3	3	250	2.21%	1.18%	15.03%	18.25%
18	1	1	200	2.53%	3.07%	12.21%	24.35%
	2	2	250	3.51%	2.28%	18.43%	19.23%
	3	3	300	1.50%	2.54%	22.13%	18.99%
20	1	1	250	3.23%	4.07%	17.21%	22.27%
	2	2	300	3.23%	7.63%	16.31%	20.04%
	3	3	350	2.23%	4.07%	17.21%	25.87%
30	1	1	300	0.68%	1.54%	20.33%	28.95%
	2	2	350	3.25%	4.00%	19.23%	25.04%
	3	3	400	2.90%	2.76%	20.73%	25.17%
60	1	1	500	0.25%	1.93%	20.60%	29.09%
	2	2	600	3.90%	1.66%	17.71%	29.17%
	3	3	700	4.23%	6.63%	18.31%	28.04%
90	1	1	1000	3.50%	3.54%	25.33%	25.99%
	2	2	1000	5.23%	6.00%	19.32%	29.04%
	3	3	1000	6.12%	7.76%	20.73%	29.96%

Biography

Mahdieh Akhbari received her BS degree in Industrial Engineering from Khajeh Nasir Toosi University of Technology in 2004, an MS degree in Industrial Engineering from Isfahan University of Technology , in 2007, and a PhD degree in Industrial Engineering, in 2014, from Science And Research Branch of Islamic Azad University . She is currently Assistant Professor in the Industrial Engineering Department of the Electronic Branch of the Islamic Azad University. Her research interests include soft computing, game theory, and project planning.