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

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
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 MRCPS. 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 MRCPS 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 Tchomt\'e 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-Material 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 consisting 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:

\[
\begin{align*}
\min Z &= \sum_{i=1}^{n} \sum_{m=1}^{M_i} \sum_{t=1}^{T} p_{imt} c_r d_{imt} x_{imt} + \sum_{q=1}^{Q} \sum_{t=1}^{T} p_{iqt} h_q I_{iqt} + \sum_{q=1}^{Q} \sum_{t=1}^{T} A_q \lambda_{qt} \\
&+ \sum_{q=1}^{Q} \sum_{t=1}^{T} \sum_{k=1}^{K} \mu_{qk} p_{qkt} Q_{qkt} + \alpha (FT - D) y - \beta (D - FT) (1 - y)
\end{align*}
\]

subject to:

1. \[
\sum_{m=1}^{M_i} \sum_{t=1}^{T} x_{imt} = 1 \quad \forall i \in \{0,1,\ldots,n\}
\]

2. \[
pred_{ij} \sum_{m=1}^{M_i} \sum_{t=1}^{T} x_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=1}^{T} \left(t - d_{jmt}\right) x_{jmt} \quad \forall i, j \in \{0,1,\ldots,n\}
\]
\[
\sum_{m=1}^{M} \sum_{t=1}^{T} x_{imt} \leq FT \quad \forall i \in \{0,1,\ldots,n\} \tag{4}
\]

\[
\sum_{i=1}^{n} M \sum_{p=1}^{P} x_{ipt} \leq res_\text{pe} \quad \forall t \in \{1,\ldots,T\}, \forall p \in \{1,2,\ldots,P\} \tag{5}
\]

\[
t_{q} = t_{q-1} + Q_{q} - \sum_{i=1}^{n} M \sum_{b=1}^{b_{x-1}} a_{qim} x_{imb} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{6}
\]

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

\[
Q_{q} \leq \lambda_{q} \text{BigM} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{8}
\]

\[
1 - (1 - \lambda_{q}) \text{BigM} \leq Q_{q} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{9}
\]

\[
\mu_{qk} O_{k-1} \leq Q_{q} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\}, \forall k \in \{1,\ldots,K_q\} \tag{10}
\]

\[
\mu_{qk} O_{k} \leq Q_{q} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\}, \forall k \in \{1,\ldots,K_q\} \tag{11}
\]

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

\[
FT(1-y) \leq D \tag{13}
\]

\[
Dy \leq FT \tag{14}
\]

\[
I_{q} \geq 0 \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{15}
\]

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

\[
y \in \{0,1\} \tag{17}
\]

\[
\lambda_{q} \in \{0,1\} \quad \forall t \in \{1,\ldots,T\}, \forall q \in \{1,2,\ldots,Q\} \tag{18}
\]
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:

\[
Q = \sum_{i=1}^{n} \sum_{m=1}^{M_i} a_{qm} x^*_{im(t+r_i+1-d_{im})}
\]  

(20)

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

So \( TC_i^t \) indicates the Total Cost for \( Q \) which is ordered and consumed in period \( t \):

\[
TC_i^t = A_q + P_q Q
\]  

(21)

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

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

(22)

And

\[
TC_i^t < TC_e^t
\]  

(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 (\( TC_i^t \)) in the comparison with \( TC_i^t \), 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 \( Y_{2j} \) 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, \( X = \left[ X_{jq} \right]_{Q \times U} \) 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.

\( X_{jq} \) 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 \( X_{jq} \) equals 0, it means the required
quantity of material $q$ in the $j$th period will be ordered with the last ordering of material $q$. $X_q, q \in \{1, 2, \ldots, Q\}$ 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 MRCSP solutions (Ys)
   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 (Xs)
      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 MRCSP 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:

\[
soc_c^{p,t} = Y = \left[ Y_{ji} \right]_{2^n}
\] (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 \( Y_{ji} \cdot j \in \{1,2\} \) and \( i \in \{1,2,\ldots,n\} \) is the social condition which is the random variable calculated as follow:

\[
Y_{ji} = lb_{ji} + r_{ji} (ub_{ji} - lb_{ji})
\] (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 (\( f(x) \)):

\[
fit_c^{p,t} = f\left( soc_c^{p,t} \right)
\] (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:

\[
P_e = 0.005N_c^2
\] (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:
\[
p_{\text{pup}}^{c,t} = \begin{cases} 
    \text{soc}^{c,t}_{i,k}, & \text{rnd}_k < P_a \text{ork}_k = k_1 \\
    \text{soc}^{c,t}_{i,k}, & \text{rnd}_k \geq P_s + P_a \text{ork}_k = k_2 \\
    R_k, & \text{otherwise}
\end{cases}
\]

(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 \( \text{rnd}_k \) is a random number inside \([0,1]\). \( P_s \) and \( P_a \) are defined as:

\[
P_s = \frac{1}{D}
\]

(29)

\[
P_a = \frac{1 - P_s}{2}
\]

(30)

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

Let \( \text{age}^{c,t}_{c} \) be the age of \( c \) th coyote in the \( p \) th 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, \( \text{alpha}^{p,t} \) 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:

\[
cult^{c,t}_{i} = \begin{cases} 
    \frac{O^{c,t}_{(N_i+1)} - 1}{2}, & \text{N}_i \text{is odd} \\
    \frac{O^{c,t}_{N_i} + O^{c,t}_{N_i+1}}{2}, & \text{otherwise}
\end{cases}
\]

(31)
where \( O_{i,j}^{p,t} \) represents the \( j \)th rank of the ordered \( i \)th social condition of all coyotes in the \( pt \)th pack at the \( t \)th instant of time.

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

\[
new - soc_{e}^{p,t} = soc_{e}^{p,t} + r_1 (\text{alpha}_{e}^{p,t} - soc_{c_{r_1}}^{p,t}) + r_2 (\text{cult}_{e}^{p,t} - soc_{c_{r_2}}^{p,t})
\]

(32)

Where \( r_1 \) and \( r_2 \) are two random numbers inside \([0,1]\) and \( c_{r_1} \) and \( c_{r_2} \) are two random coyotes in the pack.

The next population would be updated as follows:

\[
soc_{e}^{p,t+1} = \begin{cases} 
new - soc_{e}^{p,t} & f \left( new - soc_{e}^{p,t} \right) < f \left( soc_{e}^{p,t} \right) \\
 soc_{e}^{p,t} & \text{otherwise} 
\end{cases}
\]

(33)

The pseudo code of the COA algorithm is as follows:

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

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:

\[ Y_k(t+1) = Y_p(t) - A |C Y_p(t) - Y_k(t)| \]  

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

\[ A = 2a r_1 - a. \]  

\[ C = 2r_2. \]

in which \( r_1 \) and \( r_2 \) are random vectors in \([0,1]\), 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 \( (Y_\alpha, Y_\beta \text{ and } Y_\gamma) \) are saved and others including omegas should update their positions according to the positions of the best search agents, based on the following formulas:

\[
Y_1 = Y_\alpha - A_1 |C_1 Y_\alpha - Y|, \\
Y_2 = Y_\beta - A_2 |C_2 Y_\beta - Y|, \\
Y_3 = Y_\gamma - A_3 |C_3 Y_\gamma - Y|, \\
Y(t+1) = \frac{Y_1 + Y_2 + Y_3}{3}.
\]

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 \(|A| \leq 1\), 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:

\[
\begin{align*}
\text{Initialize the grey wolf population } Y_i, \ i \in \{1, 2, \ldots, \text{popsize}\}, \\
\text{Initialize } a, A, \text{ and } C, \\
\text{Calculate the fitness of each search agent,} \\
Y_\alpha = \text{the best search agent,} \\
Y_\beta = \text{the second best search agent,} \\
Y_\gamma = \text{the third best search agent,} \\
\text{While } (t < \text{Max number of iterations}) \\
\quad \text{for each agent} \\
\quad \quad \text{Update the position of the current search agent by equation (40)} \\
\quad \text{end for}
\end{align*}
\]
Update \( a, A, \text{and} C \),
Calculate the fitness of all search agents
Update \( Y_a, Y_{\beta}, Y_\delta \),
\[ t = t + 1 \]
end while
Return \( Y_a \)

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:

\[
V(t+1) = \omega V(t) + c_1 r_1 (Y_{\text{pbest}}(t) - Y(t)) + c_2 r_2 (Y_{\text{gbest}}(t) - Y(t)) \tag{41}
\]

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

Where \( \omega \) 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, \( Y_{\text{pbest}}(t) \) is the best position of the particle and \( Y_{\text{gbest}}(t) \) 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 $X = \left[ X_{\alpha} \right]_{q=1}^Q$; which was explained in Sect. 5.1; is encoded to a chromosome as shown in Figure 2 (as mentioned before $X_{\alpha}$ 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:

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

(43)

where \( y_{ij} \) is the response value of \( i \) th 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:

\[
RDP_{ij} = \frac{Z_{ij}^* - Min\left(Z_{ik}^*\right)}{Min\left(Z_{ik}^*\right)}
\]

(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, \( L_9(3^4) \), 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:

\[
\begin{align*}
H_0 : \mu_d &= 0, \\
H_1 : \mu_d &> 0.
\end{align*}
\]

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


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 metaheuristics and BARON

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

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

Table 16- Comparison of the metaheuristics based on $RDP$
An example of solution representation (Y)

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

Figure 2

a)

Offspring 1

Offspring 2

b)

Offspring 1

Offspring 2

Figure 3

Figure 4
Figure 9

Figure 10
Figure 11
Table 1

<table>
<thead>
<tr>
<th>Indices</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i, j \in {0,1,\ldots,n}$</td>
<td>$d_{im}$ Duration of activity $i$ executed in mode $m$</td>
</tr>
<tr>
<td>$t, b \in {1,\ldots,T}$</td>
<td>$\text{pred}_{ij} = \left{ \begin{array}{ll} 1 \ 0 \end{array} \right.$</td>
</tr>
<tr>
<td>$p \in {1,2,\ldots,P}$</td>
<td>$\text{res}_{pt}$ Number of units of renewable resource $p$ available in period $t$</td>
</tr>
<tr>
<td>$q \in {1,2,\ldots,Q}$</td>
<td>$r_{pim}$ Number of units of renewable resource $p$ required by activity $i$ executed in mode $m$</td>
</tr>
<tr>
<td>$m \in {1,\ldots,M_i}$</td>
<td>$a_{qim}$ Number of units of non-renewable resource $q$ required by activity $i$ executed in mode $m$</td>
</tr>
<tr>
<td>$k \in {1,\ldots,K_q}$</td>
<td>$cr_p$ Cost of renewable resource $p$ per period</td>
</tr>
</tbody>
</table>
\( H_q \)  
Inventory holding cost per unit of material \( q \) per period

\( A_q \)  
Ordering cost per replenishment of material \( q \)

\( \text{BigM} \)  
Big number

\[
P_q = \begin{cases} 
  P_{q1} & 1 \leq Q_{qt} \leq O_{q1} \\
  P_{q2} & O_{q1} + 1 \leq Q_{qt} \leq O_{q2} \\
  \vdots & \vdots \\
  P_{qK} & O_{qK-1} + 1 \leq Q_{qt} 
\end{cases}
\]

Purchasing cost of material \( q \) under AUD scheme

\( \varepsilon \)  
Penalty per period of delay

\( \vartheta \)  
Bonus per period of early completion

\( D \)  
Due date of the project

\( T \)  
Planning horizon

\( \pi \)  
Project income at the end of the project

**Binary variables**

\[
x_{imt} = \begin{cases} 
  1 & \text{If activity } i \text{ is performed in mode } m \text{ and finished at time period } t \\
  0 & \text{Otherwise}
\end{cases}
\]

\[
y = \begin{cases} 
  1 & \text{If the project is finished after due date} \\
  0 & \text{Otherwise}
\end{cases}
\]

\[
\lambda_{qt} = \begin{cases} 
  1 & \text{If material } q \text{ is ordered in period } t \\
  0 & \text{Otherwise}
\end{cases}
\]

\[
\mu_{qtk} = \begin{cases} 
  1 & \text{If material } q \text{ is purchased at price breakpoint } k \text{ in period } t \\
  0 & \text{Otherwise}
\end{cases}
\]
## 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

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

### Table 4

<table>
<thead>
<tr>
<th>Factors</th>
<th>( P_c )</th>
<th>( P_m )</th>
<th>( MG )</th>
<th>( NGA )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside search</td>
<td>( P_c )</td>
<td>( P_m )</td>
<td>( MG )</td>
<td>( NGA )</td>
</tr>
<tr>
<td>Optimal levels</td>
<td>0.9</td>
<td>0.3</td>
<td>10</td>
<td>150</td>
</tr>
</tbody>
</table>

### Table 5

<table>
<thead>
<tr>
<th>Factors</th>
<th>( Iter )</th>
<th>( N_p )</th>
<th>( N_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal levels</td>
<td>200</td>
<td>15</td>
<td>14</td>
</tr>
</tbody>
</table>
### Table 6

<table>
<thead>
<tr>
<th>Factors</th>
<th>Iter</th>
<th>SearchAgents_no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal levels</td>
<td>200</td>
<td>30</td>
</tr>
</tbody>
</table>

### Table 7

<table>
<thead>
<tr>
<th>Factors</th>
<th>Swarm size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter</td>
</tr>
<tr>
<td>Optimal levels</td>
<td>400</td>
</tr>
</tbody>
</table>

### Table

<table>
<thead>
<tr>
<th>Factors</th>
<th>$P_c$</th>
<th>$P_m$</th>
<th>MG</th>
<th>NGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal levels</td>
<td>0.9</td>
<td>0.3</td>
<td>20</td>
<td>200</td>
</tr>
</tbody>
</table>
Table 9

<table>
<thead>
<tr>
<th>Activity</th>
<th>Mode 1 Duration</th>
<th>Required resources</th>
<th>Mode 2 Duration</th>
<th>Required resources</th>
<th>Mode 3 Duration</th>
<th>Required resources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Duration</td>
<td>R1</td>
<td>R2</td>
<td>NR1</td>
<td>NR2</td>
<td>Duration</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>10</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>7</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

R: Renewable resource; NR: Non-Renewable resource

Table 10.

<table>
<thead>
<tr>
<th>Non-renewable resources</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holding cost ($) per time unit</td>
<td>100</td>
<td>1,200</td>
</tr>
<tr>
<td>Ordering cost ($)</td>
<td>1,000</td>
<td>1,500</td>
</tr>
</tbody>
</table>
Table 11.

<table>
<thead>
<tr>
<th>Non-renewable resources</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Price ($)</td>
<td>Range</td>
</tr>
<tr>
<td>$1 \leq O \leq 10$</td>
<td>700</td>
<td>$1 \leq O \leq 5$</td>
</tr>
<tr>
<td>$11 \leq O \leq 20$</td>
<td>650</td>
<td>$6 \leq O \leq 10$</td>
</tr>
<tr>
<td>$21 \leq O \leq 30$</td>
<td>630</td>
<td>$11 \leq O$</td>
</tr>
<tr>
<td>$31 \leq O \leq 40$</td>
<td>600</td>
<td>41 \leq O</td>
</tr>
</tbody>
</table>

$O$: Order quantity

Table 12.

<table>
<thead>
<tr>
<th>Renewable resources</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$/time unit</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>Max level</td>
<td>19</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 13.

<table>
<thead>
<tr>
<th>Method</th>
<th>BARON</th>
<th>COA-GA</th>
<th>GWO-GA</th>
<th>PSO-GA</th>
<th>GA-GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total cost ($)</td>
<td>227,650</td>
<td>223,000</td>
<td>223,000</td>
<td>233,400</td>
<td>235,900</td>
</tr>
<tr>
<td>FT</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Renewable Resource cost</td>
<td>93,500</td>
<td>90,600</td>
<td>90,600</td>
<td>92,500</td>
<td>93,400</td>
</tr>
<tr>
<td>Holding cost</td>
<td>7,300</td>
<td>700</td>
<td>700</td>
<td>1,100</td>
<td>5,100</td>
</tr>
<tr>
<td>Ordering cost</td>
<td>16,000</td>
<td>14,000</td>
<td>14,000</td>
<td>14,000</td>
<td>15,000</td>
</tr>
<tr>
<td>Procurement cost</td>
<td>130,850</td>
<td>137,700</td>
<td>137,700</td>
<td>135,800</td>
<td>132,400</td>
</tr>
<tr>
<td>Penalty (Bonus)</td>
<td>(20,000)</td>
<td>(20,000)</td>
<td>(20,000)</td>
<td>(10,000)</td>
<td>(10,000)</td>
</tr>
<tr>
<td>Activity</td>
<td>BARON Start time</td>
<td>Mode</td>
<td>COA-GA Start time</td>
<td>Mode</td>
<td>GWO-GA Start time</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>------</td>
<td>------------------</td>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>2</td>
<td>14</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>15</td>
<td>17</td>
<td>2</td>
<td>16</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>Date</td>
<td>BARON</td>
<td>COA-GA</td>
<td>GWO-GA</td>
<td>PSO-GA</td>
<td>GA-GA</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>Material1</td>
<td>Material2</td>
<td>Material1</td>
<td>Material2</td>
<td>Material1</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>6</td>
<td>18</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0</td>
<td>12</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>1</td>
<td>15</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>11</td>
<td>12</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>23</td>
<td>11</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>N</td>
<td>RR</td>
<td>NR</td>
<td>CPU time (s)</td>
<td>COA-GA</td>
<td>GWO-GA</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>60</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>100</td>
<td>0.00%</td>
<td>0.31%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>120</td>
<td>0.21%</td>
<td>0.35%</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>100</td>
<td>0.02%</td>
<td>0.22%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>120</td>
<td>0.03%</td>
<td>2.03%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>150</td>
<td>0.03%</td>
<td>2.03%</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>120</td>
<td>0.12%</td>
<td>1.02%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>150</td>
<td>1.32%</td>
<td>3.07%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>200</td>
<td>1.35%</td>
<td>1.07%</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>150</td>
<td>0.21%</td>
<td>0.18%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>200</td>
<td>2.06%</td>
<td>2.34%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>250</td>
<td>2.21%</td>
<td>1.18%</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>200</td>
<td>2.53%</td>
<td>3.07%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>250</td>
<td>3.51%</td>
<td>2.28%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>300</td>
<td>1.50%</td>
<td>2.54%</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>250</td>
<td>3.23%</td>
<td>4.07%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>300</td>
<td>3.23%</td>
<td>7.63%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>350</td>
<td>2.23%</td>
<td>4.07%</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>300</td>
<td>0.68%</td>
<td>1.54%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>350</td>
<td>3.25%</td>
<td>4.00%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>400</td>
<td>2.90%</td>
<td>2.76%</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>1</td>
<td>500</td>
<td>0.25%</td>
<td>1.93%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>600</td>
<td>3.90%</td>
<td>1.66%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>700</td>
<td>4.23%</td>
<td>6.63%</td>
</tr>
<tr>
<td>90</td>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>3.50%</td>
<td>3.54%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>1000</td>
<td>5.23%</td>
<td>6.00%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>1000</td>
<td>6.12%</td>
<td>7.76%</td>
</tr>
</tbody>
</table>
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.