Journal of Structural and Construction Engineering

www.jsce.ir

# Scheduling Projects and Levelling Resources by Particle Swarm Optimization

**Fateme. Jazebi[1*], Moslem. Bakhshi [2]**

1- *Instructor, Department of Civil Engineering, Payame Noor University, Tehran, Iran*

2- *Senior Engineer, Department of Technical office, Havayar Company, Tehran, Iran.*

## ABSTRACT

*One of the main objectives of project management is fluctuations minimization in resource loading. This leads to increasing the utilization of equipment and labor force available. In the other words it is essential to increase work productivity. When solving construction resource levelling problems, traditional analytical approaches are inefficient and inflexible. Therefore, Many heuristic techniques have been developed for this problem to overcome drawbacks of traditional construction resource levelling algorithms. They assign priorities to the project activities based on measures obtained from some calculations. In this research particle swarm optimization is adopted to aim at finding delay range of each activity in a project that optimizes project time and resource levelling index. The project contains activities interrelated by finish-start type precedence relations with a time lag of zero, which require a set of renewable resources. The approach was tested on a problem. The computational results validate the effectiveness of the proposed algorithm. In addition, to study the computational efficiency of the model, it was applied for real projects of different structures and sizes with activities of different time–resource options. It was concluded that it is independent from these factors. Altogether, the application field of the presented method is wide and of great use for real projects because of many reasons such as improvement of the situation of managers, outperforming many past robust algorithms, its improved solution quality and lower runtime, providing the possibility of risk, delay and activities' relationships analysis.*

## ARTICLE INFO

**Receive Date:** 07 February 2025
**Revise Date:** 17 May 2025
**Accept Date:** 09 July 2025

**Keywords:**
*Particle Swarm Optimization,*
*Resource Levelling,*
*Delay Range*



**doi:** 10.22065/jsce.2025.503959.3648

*Corresponding author: Fateme. Jazebi
Email address: fa.jazebi@pnu.ac.ir

## 1. Introduction

Effective project scheduling techniques are important to ensure successful project performance; a poor strategy can easily turn expected profit into loss. Project scheduling problems consist of finding the start time for all activities, and a common objective in practice is to minimize the completion time of the project (makespan). In general, activities require different amounts of resources of which only a certain amount is available at each time step. More than that, scheduling activities at their earliest precedence feasible start time, usually causes great fluctuations in the daily resource requirements. Therefore, the need for resource levelling techniques becomes inevitable to ensure efficient utilization of resources.

This problem has so far been one of the most extensively studied subjects in the existing literature and forms an important building block of the scheduling science. For example, G.K. Koulinas developed an improved Tabu Search model to solve resource levelling problems [1]. Kastor and Sirakoulis evaluated the effectiveness of resource levelling tools of three popular packages by comparing the results when levelling two real construction projects as case studies[2]. Yibiao Liu et al optimized construction program schedule plans based on multi-objective optimization theory [3]. Lin-Yu Tseng et al. introduced a hybrid meta-heuristic for the resource-constrained project scheduling problem [4]. Behrouz Afshar-Nadjafi et al. presented a genetic algorithm for mode identity and the resource constrained project scheduling problem [5]. Mohammad Hassan Sebt et al. presented an efficient genetic algorithm for solving the multi-mode resource-constrained project scheduling problem based on random key representation [6]. Francesco Gargiulo et al. presented a hybrid genetic algorithm for resource constrained scheduling problem [7]. Jirachai Buddhakulsomsiri presented Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting [8]. Vincent Van Peteghem et al. presented a genetic algorithm for the pre-emptive and non-pre-emptive multi-mode resource-constrained project scheduling problem [9]. Jonathan Jingshen Shi et al proposed Object-oriented resource-based planning method (ORPM) for Construction [10]. Elham Nabipoor Afruzi et al. proposed A Multi-Objective Imperialist Competitive Algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations [11]. Po-Han Chen et al. developed a two-phase GA model for resource-constrained project scheduling [12]. Gizem Çakır et al. used a genetic algorithm-based model to schedule each activity in one of its modes in order to minimize the project makespan[13].

There has been research about resources scheduling under particular circumstances such as multi-project scheduling. For example E. Nabipoor Afruzi et al. proposed a robust optimization for the resource-constrained multi-project scheduling problem with uncertain activity durations [14]. Jinghua Li et al. represented a hybrid genetic algorithm for the resource constrained multi-project scheduling problem [15]. Ali Moradi Vartouni et al proposed a hybrid genetic algorithm and fuzzy set applied to multi-mode resource-constrained project scheduling problem [16]. Ashish Sharma presented a Fuzzy Multi-Objective Genetic Algorithm Based Resource Constrained Time-Cost Trade-Off Model under Uncertain Environment [17]. For example Amir Hossein Hosseinian et al. studied about the A two-phase approach for solving the multi-skill resource-constrained multi-project scheduling problem: a case study in construction industry [18]. Gopinath Selvam et al. developed a Genetic algorithm based optimization model for resource levelling problem with precedence constrained scheduling [19]. Zhiqiang Ma et al developed a genetic algorithm for the proactive resource-constrained project scheduling problem with activity splitting [20].

All of the mentioned research and other research about resource levelling and resource-constrained project scheduling have disadvantage/s such as being unusable for big projects and lack of the possibility of assignment of importance levels of project time and resource levelling index. In this research a useful simple model is produced. The model is produced by particle algorithm and aims to optimize a variable which is function of project time and resource levelling index. Importance level of project time and resource levelling index which are assigned by decision maker are also parts of the variable.

The remaining of this paper starts with a definition of the project scheduling problem being solved. The focus then moves to an introduction of swarm intelligence and the details of the proposed algorithm. The application of the proposed algorithm is demonstrated through an example project, which is followed by analyses on the results. Conclusions and future developments appear in the last section.

## 2. Problem Statement

In project scheduling, if resources fluctuations are beyond acceptable limits, actions must be taken to move activities or resources around in order to level out the resource-loading graph. For example, it is bad for employee morale and public relations when a company has to hire and lay people off indiscriminately. Proper resource planning will facilitate a reasonably stable level of the work force. Other advantage of resources levelling includes simplified resource tracking and control, lower cost of resources management, and improved opportunity for learning. For levelling resources there are some tools such as Primavera and MS Project. Each tool uses a special approach to level resources.

This scheduling tools largely relies on heuristics in levelling resource in forward pass scheduling; thus, in general, they do not yield the optimum solution in terms of shortest project duration. In addition, the proposed resource levelling methods, though, does not necessarily optimize the allocation of resources so as to minimize the project duration. For example, a resource levelling may leave the project schedule unchanged even if a different resource allocation may result in a shorter or longer project schedule. Therefore, it is essential to use more appropriate method, which considers project time, for levelling resources. In the other words, both of the objectives, project time minimization and resources levelling, should be considered. To this purpose, a resourse levelling index should be used in project scheduling and in addition to project time, the mentioned index should be optimized.

In addition, if several resources need to be leveled simultaneously, the objective can be defined as the sum of the multiplication of the index for each resource and its corresponding weighting score. The weighting score is a man-made value and its value, ranging from 0 to 1, depends on the importance of the resource.

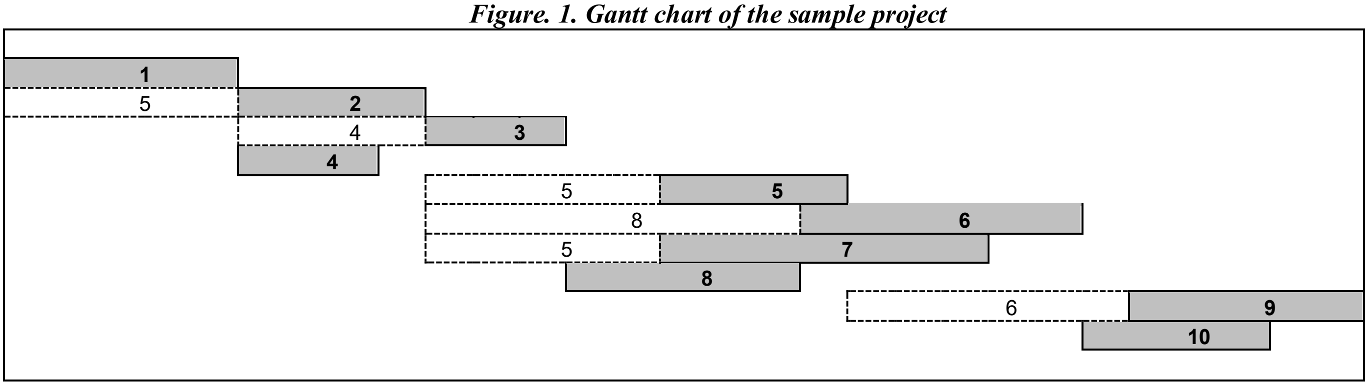
In reference to the project duration, the first criterion set is that the project completion date must be as much as possible equal to or earlier than the specified required project completion date. The required project completion date is up to the project manager to specify and may depend on several considerations. This requirement is such that any project completion later than this, have low value. Thus, the required project completion date serves as a limit or deadline on how late the completion of the project can be. In reference to the fluctuations in the daily resource requirements, the second criterion set is that the fluctuations in the daily resource requirements, should be minimized. Thus, time and fluctuations in the daily resource requirements optimization should arrive at the set of shifting amount of activity that results in a valid project completion date and has the minimum fluctuations in the daily resource requirements.

Some of the activities should be delayed for levelling resources. So delay range of each activity determination, project time and resource levelling are considered, is the under research problem. Table 1 and figure 1 demonstrate the concept of delaying activities after completion time of their predecessors.

***Table 1. A sample project***

| activity | Duration | pred | Delay range |
|---|---|---|---|
| 1 | 5 | - | 0 |
| 2 | 4 | - | 5 |
| 3 | 3 | 1 | 4 |
| 4 | 3 | 1 | 0 |
| 5 | 4 | 2 | 5 |
| 6 | 6 | 1,2 | 8 |
| 7 | 7 | 1,2 | 5 |
| 8 | 5 | 3,4 | 0 |
| 9 | 5 | 4,5 | 6 |
| 10 | 4 | 6,7 | 0 |

***Figure. 1. Gantt chart of the sample project***




The project scheduling and resources levelling problem under study is based on the following assumptions: (1) the activities composing a project have certain and known durations; (2) all predecessors must be finished before an activity can start (i.e., precedence constraints); (3) resources are renewable from period to period; (4) activities are non-preemptive, that is, cannot be interrupted when in progress.

# 3- Particle swarm optimization for levelling resources by delaying the activities in scheduling projects

## 3-1-Objective Function

The objective function includes two objectives:

Objective 1

The objective 1 is to determine the minimal duration for a construction project; that is, it is better if the project completion time is shorter.

Minimize ‘T’

where ‘T’ is the set of the maximum finish time of all activities in the project.

Objective 2

The second objective is defined to minimize deviation of resource usage amount. There are some resource levelling indexes[21-22]. In this research the objective 2 implemented the minimum moment algorithm to make the daily resource consumption $r_{tj}$ as smooth as possible. Also, the objective 2 use an amplification factor which is the difference between the daily resource consumption $r_{tj}$ and the resource consumption of the last day $r_{(t-1)j}$. This difference is used to select the optimal daily resource consumption and enable it to reach the optimum use of resources. In the paper, also, considering the multi-type resource allocation problem, the weight of every kind of resource is generally added, wherein the weighted value $W_j$ representing its importance. For example, whether a resource type is important or not could depend on its unit price. The higher the unit price of this resource type, the more important is this resource type.

Minimize

$$\sigma = \sum_{j=1}^{m} \left\{ \left[ \sum_{t=1}^{n} {r_{tj}}^2 + \sum (r_{tj} - r_{(t-1)j})^2 \right] \times W_j \right\}$$

where $r_{tj}$ is the consumption of type j at time t in the project, $W_j$ the weight of the resource j, n the number of optimal project duration on the condition of the resource constraints, and m the number of total type resources project days[23].

total objective:

A compound of project time index and resource levelling index makes the total objective:

$$Total \quad objective = IL_T \times \frac{T}{T_{basis}} + IL_\sigma \times \frac{\sigma}{\sigma_{basis}}$$

Where $IL_T$ is importance level of project time index, $IL_\sigma$ is importance level of resource levelling index, T is project time, σ is resource levelling index, $T_{basis}$ is basis amount of project time index, $\sigma_{basis}$ is basis amount of resource levelling index.

## 3-2-Particle swarm optimization

The application of swarm intelligence in optimization was first developed by Eberhart and Kennedy under the name of particle swarm optimization (PSO) [24]. The strength of PSO is underpinned by the fact that decentralized (without central supervision) biological creatures can often accomplish complex goals by cooperation [25].

A standard PSO algorithm is initialized with a population (swarm) of random potential solutions (particles). Each particle iteratively moves across the search space and is attracted to the position of the best fitness (evaluation of the objective function) historically achieved by the particle itself (local best) and by the best among the neighbors of the particle (global best). In essence, each particle continuously focuses and refocuses the effort of its search according to both local and global best. This behaviour mimics the cultural adaptation of a biological agent in a swarm: it evaluates its own position based on certain fitness criteria, compares with others, and imitates the best in the entire swarm [26].

A standard PSO updates from time step t to t+1. The update moves a particle by adding a change velocity $\vec{v}_i(t+1)$ to the current position $\vec{x}_i(t)$ as follows:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{3}$$

The velocity is a combination of three contributing factors: (1) previous velocity $\vec{v}_i(t)$ (2) movement in the direction of the local best $\vec{x}_{lbest}$, and (3) movement in the direction of the global best $\vec{x}_{gbest}$. The mathematical formulation is expressed as

$$\vec{v}_i(t+1) = \omega \times \vec{v}_i(t) + r_1 c_1(\vec{x}_{lbest} - \vec{x}_i(t)) + r_2 c_2(\vec{x}_{gbest} - \vec{x}_i(t)) \tag{4}$$

Where ω is an inertia weight to control influence of the previous velocity; $r_1$ and $r_2$ are two random numbers uniformly distributed in the range of (0,1); $c_1$ and $c_2$ are two acceleration constants.

The parameters in Eq. (4) can be used to fine-tune the optimization process. These tuning parameters regulate the balance between local exploitation and global exploration; the former ensures a faster convergence whereas the latter increases chances to discover the global optimal solution. Experimental results indicate that a smaller ω promotes exploitation while a larger inertia weight ω places emphasis on exploration. It is therefore common for a PSO algorithm to start with a large ω for greater exploration and generally decreases it to obtain refined solutions [27].

In the remaining part of Eq. (4), $\vec{x}_{lbest} - \vec{x}_i(t)$ models the ‘‘learning from own experience’’ behavior of individual particles whereas $\vec{x}_{gbest} - \vec{x}_i(t)$ reflects the ‘‘social interaction’’ phenomenon among particles. These two terms are stochastically weighted by $r_1c_1$ and $r_2c_1$. The randomness provides the possibility to reach the global optimal solution without being trapped at local optimal solutions (exploration). This mechanism is similar to other local search methods, such as simulated annealing or tabu search.

To prevent undesired oscillations (i.e., the velocity may expand wider and wider until approaching infinity), the velocity is often dampened by an upper limit. Since the velocity may reverse (be positive or negative), the damping process includes two parts

$$\text{If } \vec{v}_i(t+1) > V_{max},\ \vec{v}_i(t+1) = V_{max} \tag{5}$$

$$\text{If } \vec{v}_i(t+1) < -V_{max},\ \vec{v}_i(t+1) = -V_{max} \tag{6}$$

Where $V_{max}$ is the specified damping limit[28].

The mentioned procedure is repeated, until certain stopping criteria are met (e.g. a fixed number of iterations; or changes on $\vec{x}_{gbest}$ becomes negligible.).

The pseudo code of the PSO algorithm is summarized as follows:

- Initialize a population of particles with random positions and velocities.
- Update the velocity of each particle using Eq. (4).
- Update the position of each particle according to Eq. (3).
- Evaluate each particle’s fitness value, and update pbest and gbest position if necessary.
- Repeat this procedure until the number of iterations reaches a predetermined value[29].

There have been several studies that apply particle swarm optimization algorithms for solving project scheduling problem [30-33]. In this research, a GA-based analysis is used to determine which execution mode should be selected for each activity to minimize project cost.

### 3-3- Proposed PSO technique for solving Resource levelling scheduling

In the proposed algorithm, each particle is located in an N-dimensional search space where each dimension represents an activity delay range in the project. For simplicity, the position of the particle in every dimension is expressed as a vector element $x_{ij}$ for i=1, 2, . . ., N; j=1, 2, . . ., M, where i represents an activity in the project; j stands for a particle; N is the number of activities; and M is the size of the swarm.

The delay ranges of activities are mapped to the project schedule. This requires a subroutine to perform standard CPM (critical path method) computations. The mentioned objective function represents the fitness of the particle at its current position.

The procedure of the proposed algorithm includes the following steps:

1. Generate an initial population of particles randomly in the feasible region.

2. Evaluate the fitness of each particle at its current position according to the objective function in Eq. (2).
3. If any particle has located a better (smaller) fitness, replace its previous best position $\vec{x}_{lbest}$ by the current one.
4. Compare the best fitness of the current particle to the neighbors; if the current article outperforms its neighbors, update the swarm global best position $\vec{x}_{gbest}$.
5. Determine the new velocity according to Eq. (4); adjust it by the damping process with Eqs. (5) and (6) if necessary.
6. Check if the new velocity will cause the particle to move outside the feasible boundary. If the answer is no, update each particle's position by Eq. (3); otherwise, skip the particle. In addition, if the position of the particle in each dimension gets a negative value, turn the negative value into zero.
7. Repeat Steps 2–6 until the termination criteria is met, i.e., the maximum number of iterations is reached. Then $\vec{x}_{gbest}$ is the best solution found.

## 4-Computational study

A project is considered to illustrate the algorithm application. which consist of fifteen activities. The activity data are shown in Table 2. Resources cost rates are 20 and 15 units per day. The objective of the proposed model is to identify all solutions for different values of $c_1$ , $c_2$ and $IL_T$.

***Table.2. activity data***

| activity | prec | duration | Res1 requirement | Res2 requirement |
|---|---|---|---|---|
| 1 | - | 25 | 5 | 2 |
| 2 | 1 | 35 | 10 | 3 |
| 3 | 1 | 22 | 8 | 5 |
| 4 | 1 | 25 | 10 | 7 |
| 5 | 2 | 13 | 10 | 15 |
| 6 | 3,4 | 20 | 18 | 20 |
| 7 | 5,6 | 18 | 10 | 14 |
| 8 | 4,6 | 17 | 3 | 9 |
| 9 | 5,6,8 | 13 | 5 | 9 |
| 10 | 7,9 | 20 | 12 | 9 |
| 11 | 7,9 | 14 | 9 | 9 |
| 12 | 9,10 | 15 | 4 | 8 |
| 13 | 8,11 | 10 | 4 | 6 |
| 14 | 12 | 16 | 14 | 12 |
| 15 | 12 | 15 | 9 | 5 |

The produced model was run based on different values of $c_1$, $c_2$ and $IL_T$. The values of $T_{basis}$ and $\sigma_{basis}$ are equal to project time and resource levelling index of the project schedule in which the delay ranges of all activities are equal to zero. The value of ω change from one to zero graduely at run time. The results of model running based on different values of $c_1$, $c_2$ and $IL_T$ are presented in table 3.

***Table.3.result of the model-***

| | C1=1, c2=1 | c1=1, c2=1.5 | c1=1, c2=2 | c1=1.5, c2=1 | c1=1.5, c2=1.5 | c1=1.5, c2=2 | c1=2, c2=1 | c1=2, c2=1.5 | C1=2, c2=2 |
|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.4 | tot | 1.52 | 1.48 | 1.34 | 1.54 | 1.45 | 1.52 | 1.44 | 1.35 | 1.52 |
| | num | 10 | 14 | 4 | 4 | 4 | 41 | 4 | 6 | 22 |
| | T | 207 | 242 | 227 | 176 | 202 | 195 | 175 | 218 | 250 |
| | Σ | 1139455 | 987755 | 862465 | 1265565 | 1079115 | 1178555 | 1147735 | 902715 | 1010225 |
| 0.6 | tot | 1.77 | 1.78 | 1.71 | 1.76 | 1.79 | 1.77 | 1.66 | 1.72 | 1.63 |
| | num | 2 | 13 | 2 | 75 | 2 | 2 | 10 | 12 | 32 |
| | T | 208 | 226 | 174 | 203 | 191 | 163 | 169 | 196 | 210 |
| | σ | 1110305 | 1034925 | 1201775 | 1119855 | 1210145 | 1319485 | 1162665 | 1107595 | 938275 |
| 0.8 | tot | 2.06 | 1.9 | 3.03 | 2.11 | 2.05 | 1.98 | 2.04 | 2.06 | 1.99 |
| | num | 24 | 2 | 61 | 8 | 68 | 55 | 2 | 36 | 82 |
| | T | 197 | 197 | 178 | 170 | 202 | 180 | 172 | 224 | 225 |
| | σ | 1194895 | 1002975 | 1274415 | 1427135 | 1144870 | 1202085 | 1326905 | 1022445 | 927995 |
| 1 | tot | 2.32 | 2.24 | 2.32 | 2.25 | 2.28 | 2.26 | 2.34 | 2.18 | 2.12 |
| | num | 5 | 99 | 33 | 41 | 44 | 10 | 72 | 47 | 50 |
| | T | 175 | 171 | 197 | 173 | 161 | 173 | 153 | 177 | 170 |
| | Σ | 1359595 | 1296085 | 1184175 | 1292255 | 1423015 | 1309535 | 1558205 | 1186915 | 1167645 |
| 1.2 | tot | 2.49 | 2.48 | 2.52 | 2.49 | 2.53 | 2.48 | 2.5 | 2.36 | 2.45 |
| | num | 56 | 35 | 83 | 4 | 16 | 21 | 57 | 44 | 18 |
| | T | 162 | 172 | 173 | 159 | 155 | 163 | 152 | 188 | 153 |
| | Σ | 1412735 | 1305615 | 1337565 | 1441885 | 1525555 | 1387545 | 1514135 | 1005645 | 1451755 |
| 1.4 | tot | 2.73 | 2.7 | 2.7 | 2.78 | 2.71 | 2.74 | 2.66 | 2.59 | 2.71 |
| | num | 10 | 56 | 10 | 53 | 8 | 29 | 4 | 7 | 57 |
| | T | 158 | 164 | 165 | 183 | 159 | 159 | 150 | 164 | 174 |
| | Σ | 1485895 | 1383405 | 1373545 | 1268145 | 1449985 | 1480245 | 1487955 | 1254585 | 1288545 |
| 1.6 | tot | 2.93 | 2.98 | 2.93 | 2.88 | 2.87 | 2.84 | 2.98 | 2.9 | 2.94 |
| | num | 58 | 10 | 55 | 27 | 76 | 70 | 86 | 75 | 18 |
| | T | 155 | 173 | 167 | 150 | 169 | 150 | 151 | 155 | 150 |
| | σ | 1511545 | 1337705 | 1359955 | 1508385 | 1265180 | 1464085 | 1619805 | 1478765 | 1587095 |

In table 3, tot is the value of total objective, num the run number in which the model result was obtained, T project time and σ is resource levelling index. For each $IL_T$ , the minimum total objective is selected from different obtained results based on different values of $c_1$ and $c_2$, and used as solution. The mentioned solutions were shaded in table 3.

As it can be seen in table 3, the bigger $IL_T$ is selected, the shorter project time and bigger resource levelling index are resulted. In addition, different values of $c_1$ and $c_2$ cause low difference in the PSO performance. But generally big $c_1$ and big $c_2$ result in better solutions. It reveals the particles' initial positions are not good and should be improved. In addition, for small $IL_T$ particles get the solution sooner. It reveals delay ranges in particles' initial positions are big. So the model is run based on different value of $c_1$, $c_2$ and $IL_T$. In the second run, delay ranges in particles' initial positions are smaller. The result of second model running are presented in table 4.

***Table 4. result of the model-***

| | | c1=1, c2=1 | c1=1, c2=1.5 | c1=1, c2=2 | C1=1.5, c2=1 | c1=1.5, c2=1.5 | c1=1.5, c2=2 | c1=2, c2=1 | c1=2, c2=1.5 | c1=2, c2=2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.4 | tot | 1.48 | 1.48 | 1.47 | 1.49 | 1.47 | 1.47 | 1.36 | 1.36 | 1.36 |
| | num | 2 | 29 | 55 | 2 | 2 | 2 | 2 | 66 | 67 |
| | T | 207 | 201 | 235 | 222 | 199 | 205 | 199 | 201 | 198 |
| | ind | 1093125 | 1119295 | 1001515 | 1056785 | 1111975 | 1096145 | 977905 | 977905 | 981015 |
| 0.6 | tot | 1.77 | 1.71 | 1.67 | 1.62 | 1.61 | 1.62 | 1.61 | 1.62 | 1.64 |
| | num | 84 | 6 | 26 | 2 | 2 | 2 | 2 | 2 | 2 |
| | T | 184 | 176 | 183 | 192 | 184 | 190 | 200 | 206 | 212 |
| | ind | 1221335 | 1190945 | 1105315 | 1010545 | 1035225 | 1017225 | 959535 | 945235 | 930935 |
| 0.8 | tot | 1.99 | 1.94 | 1.93 | 1.9 | 1.89 | 1.89 | 1.98 | 1.86 | 1.97 |
| | num | 2 | 42 | 47 | 2 | 18 | 60 | 2 | 41 | 39 |
| | T | 194 | 195 | 193 | 191 | 186 | 182 | 175 | 180 | 152 |
| | ind | 1128690 | 1061685 | 1069565 | 1036045 | 1063295 | 1086795 | 1241035 | 1063665 | 1371110 |
| 1 | tot | 2.24 | 2.2 | 2.12 | 2.22 | 2.22 | 2.21 | 2.25 | 2.15 | 2.16 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | num | 11 | 6 | 2 | 2 | 2 | 13 | 2 | 4 | 2 |
| | T | 190 | 196 | 179 | 167 | 167 | 185 | 185 | 187 | 173 |
| | ind | 1148665 | 1053755 | 1094585 | 1312475 | 1302215 | 1157895 | 1197255 | 1069890 | 1188815 |
| 1.2 | tot | 2.41 | 2.45 | 2.36 | 2.45 | 2.45 | 2.41 | 2.49 | 2.41 | 2.44 |
| | num | 2 | 2 | 4 | 2 | 2 | 2 | 2 | 37 | 2 |
| | T | 166 | 173 | 174 | 162 | 163 | 159 | 162 | 175 | 178 |
| | ind | 1280105 | 1257395 | 1140785 | 1366285 | 1353035 | 1345145 | 1412355 | 1189920 | 1199785 |
| 1.4 | tot | 2.72 | 2.7 | 2.67 | 2.7 | 2.69 | 2.73 | 2.57 | 2.58 | 2.68 |
| | num | 21 | 39 | 26 | 2 | 2 | 32 | 2 | 2 | 12 |
| | T | 154 | 152 | 175 | 170 | 174 | 153 | 171 | 176 | 163 |
| | ind | 1515345 | 1518205 | 1226635 | 1318335 | 1264165 | 1534205 | 1153205 | 1109205 | 1366175 |
| 1.6 | tot | 2.92 | 2.92 | 2.84 | 2.9 | 2.9 | 2.93 | 2.8 | 2.81 | 2.9 |
| | num | 8 | 54 | 2 | 2 | 2 | 54 | 2 | 2 | 83 |
| | T | 157 | 170 | 150 | 151 | 150 | 163 | 172 | 174 | 156 |
| | ind | 1466815 | 1303795 | 1464085 | 1523505 | 1541105 | 1343985 | 1135845 | 1121785 | 1461305 |

As it can be seen in table 4, big $c_1$ and $c_2$ result in good solutions for big and small $IL_T$, and small $c_1$ and $c_2$ result in good solutions for medium $IL_T$. It reveals delay ranges in particles' initial positions are medium. In addition, for medium $IL_T$ particles get the solution sooner. It confirmes the discovered issue.

One of the combinations in table 4 (c1=2, c2=1.5 , $IL_T$=0.8) is studied to investigate the evolutionary process of solution improvement. Figures 2,3 and 4 demonstrate total objective, resource levelling index and project time of global optimal solution at different iterations respective. Figure 5 demonstrates the total objective of local optimal solution at different iterations.

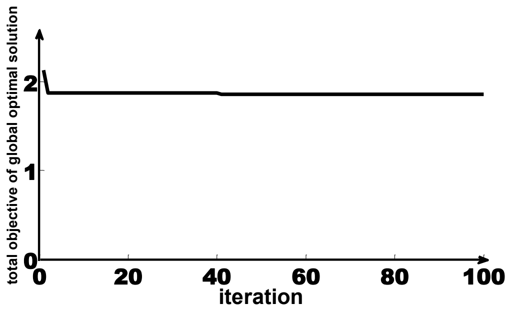



***Figure. 2. Total objective of global optimal solution at different iterations***

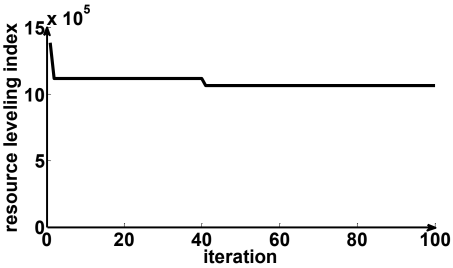



***Figure. 3. Resource levelling index of global optimal solution at different iterations***

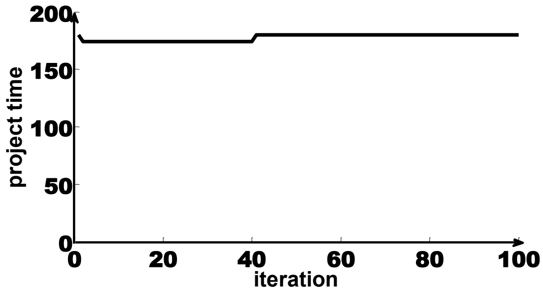



*Figure. 4. Project time of global optimal solution at different iterations*

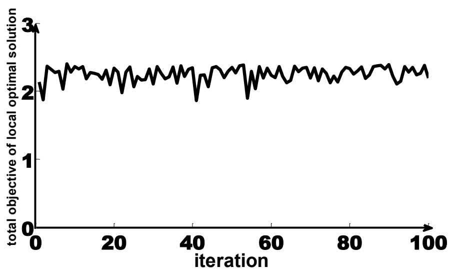



*Figure. 5. Total objective of local optimal solution at different iterations*

As it can be seen in figures 2,3,4 and 5, gradually total objective of global optimal solution decreases. This issue is evident. The total objective of local optimal solution have different value at different iterations and it's minimum amount is in iteration 41 from which total objective of global optimal solution does not change. Gradually project time of global optimal solution increases and resource levelling index of global optimal solution decreases. This issue reveals delay ranges in particles' initial positions are small for $IL_T$=0.8.

## 5-conclusion

- In this paper, we discussed how to solve a basic project scheduling problem using particle swarm optimization algorithm, initialized by some solutions. Essentially, we used different parameters value, to find optimal or near-optimal solutions. It was shown that the proposed method aims at completing the project in the shortest duration while extracting the most efforts from the resources available. In the other words, it could find the best optimal makespan and resource levelling index; therefore, It is practical and useful for scheduling under resource and time constraints at both planning and execution stages. In the proposed method, the resulting deterministic schedule has a very big benefit from managerial aspects, ie levelling resource, and improving the situation of the manager, to negotiating for managing resources in a more robust way even in a construction program which involves multiple subprojects [3] [14].
- In project planning, for creating a desirable deterministic schedule, the method considers delay ranges of activities as decision variables to minimize project duration and resource levelling index. The project duration is determined by precedence relationships and delay ranges. By using the "as soon as possible" approach and delay ranges of activities. The feature and effectiveness of the proposed particle swarm optimization have been addressed through a computational experiment on the project examples. It substantiated that the proposed algorithm outperforms models claimed to work very well [4] [6].
- In comparison with the other models presented in other research which are of improved solution quality and lower runtime [1] [24], the method presented by the paper is simpler and more efficient. The proposed method guarantees the optimal solution as it examines a big part of the feasible solutions, however, its computational requirements are small.
- We have applied the method and MATLAB language to a number of test cases with varying project structure and size (number of activities) and activity time–resource options (the computational efficiency of the model depends

mainly on these factors). Results from the evaluation show that the method can be reliably applied to actual engineering projects in terms of accuracy and solution efficiency. In other words, the proposed model can be used to solve actual construction scheduling problems of varying size and complexity and, it is hoped that practical application of the proposed model will prove to be so attractive to practitioners for project scheduling and resource levelling in construction that they abandon common project management software packages [2].

- Usually, Project scheduling is a troublesome duty due to some reasons. But, an innovative framework which is easy of computerization and understanding can determine the best schedule for project implementation by relying on small amount of data. Usually, in the field of construction management and execution, the project managers aim is to operate within some thresholds that maximize or minimize some levels that also allows for a compromise by which some issues can be fulfilled. This fully addressing some aspects leads to added computational processing which results in the fact that large amount of data should be available. This, itself results in high complexity of the used model and needs huge computational resources. As a result, it is noteworthy that in the implementation process of most of the literature on project scheduling method, even under the simplifying assumptions, complex resource types and diverse resource attributes are involved [5] [20]. As a result, the applied model's complexity reduction is a determinant factor and requirement in complex projects such as construction ones. The number of parameters used in this study is small and the required knowledge is not much. Therefore, practical implementation of the proposed model which is an advanced computational optimization method is not time-consuming and the model is implemented on a computer system with a user-friendly interface which does not require considerable computational effort. With the help of novelties of the proposed model, one of enough expertise in scheduling and programming can correctly and easily extract some knowledge and consider and define some issues in real-world situations to simulate scheduling. All this effort is made without much expense in terms of collection and computation time which is of immense importance to both practitioners and researchers.
- During the course of a project, there is the possibility of the occurrence of interrupting events such as unexpected adverse environmental conditions, and it should be taken into account to ensure successful project completion. In fact, delay analysis should be done to deal with disruptions during the stage of project execution. In this way, the gap between practical needs and theoretical tools concerning project network methods will be closed in a more simple way than other proposed ways [29] [31]. This is because of the mentioned easiness of development of the proposed model. This causes the wide-range applicability of the proposed model to actual projects which provides project managers with additional degrees of freedom to reiterate the application of the model during project due to some issues such as human errors and managerial inefficiencies.
- In most practical cases just a little knowledge is available concerning each task of the project which causes variability of many issues such as starting time. For example, foreseeable and unforeseeable risks which causes unforeseen major design revisions require frequent changes in resource requirement. As we all know, proactive scheduling has been the subject of many research efforts that aim to generate robust baseline schedules that are protected against schedule disruptions. The more robust the baseline schedules are, the lower the adjustment costs will be during project execution. But, it is unavoidable that given the high degree of uncertainty in project environment, there are many decision making situations for which a project manager should find the optimal alternative from a number of feasible alternatives. It means that he/she needs to revise the initial developed model which was programmed to handle dynamic, real-time changes. The theoretical and practical issues of the presented method make this revision possible. This model’s ability which increases its practical relevance and cannot be found in other proposed methods [17], is the first major practical contribution of the present research. This decrease incurred costs when project managers adjust the starting times of the activities to deal with uncertainties which may be so powerful that change the project scope.
- Emerging risks cause changes, delays or accelerations in one task which influence the overall project timeline and costs, because activities are often interdependent, and these precedence relations among project activities are usually influenced by simplifying assumptions in modeling scheduling problems. But, needless to say, activities are different with respect to their type of dependencies. The precedence relations between activities are usually governed by some factors, such as construction methods, safety consideration, space constraints, resource availability, project logic and traffic management and are of hard or soft character. If a relation connected with two activities is hard, its logical sequence cannot be changed. soft logics are those relations that in certain circumstances allow the activities connected by them to be scheduled as per a variety of work sequences or performed simultaneously. For example, it is physically possible to perform a repetitive activity in some units in the sequence instead of one unit, and weaken this predefined relation to generate other optional sequences. But, this changing the work sequence may result in longer activity durations and additional costs, and the change of project time and cost between which there is a trade-off, follow these changes. As a result, before and during project execution, it is possible to examine the impacts and variability of some precedence relationships more effortlessly than other proposed ideas [19] [22].

- In the course of future research, surveys of the combination of the proposed method with other scheduling algorithms, such as time-cost trade-off, may be of importance to researchers and practitioners. Further application of particle swarm optimization to a more complicated resource levelling problem with consideration of uncertain activity duration or preemption is under study at present. Also, further revision and modification of the particle swarm optimization procedure code could possibly improve the speed and the efficiency of the optimization process, and enhance its ability to arrive at quick solutions.

## References:


[1] Koulinas, G.K., Anagnostopoulos, K.P., (2013). A new tabu search-based hyper-heuristic algorithm for solving construction levelling problems with limited resource availabilities, *Automation in Construction*, 31, 169–175.

[2] Kastor, A., Sirakoulis, K., (2009). The effectiveness of resource levelling tools for Resource Constraint Project Scheduling Problem, *International Journal of Project Management*, 27(5), 493–500.

[3] Liu, Y., Cheng, H., Liu, C., (2022). Optimization of Construction Program Schedule Plans Based on Multi-Objective Optimization Theory, *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 46(6), 4763–4782.

[4] Tseng, L.Y., Chen, S.C., (2006). A hybrid meta-heuristic for the resource-constrained project scheduling problem, *European Journal of Operational Research*, 175(2), 707–721.

[5] Afshar-Nadjafi, B., Rahimi, A., Karimi, H., (2013). A genetic algorithm for mode identity and the resource constrained project scheduling problem, *Scientia Iranica*, 20(3), 824-831.

[6] Sebt, M.H., Afshar, M.R., Alipouri, Y., (2015). An efficient genetic algorithm for solving the multi-mode resource-constrained project scheduling problem based on random key representation, *International Journal of Supply and Operations Management*, 2(3), 905-924.

[7] Gargiulo, F., Quagliarella, D., (2011). a hybrid genetic algorithm for resource constrained scheduling problem, *Proc. Int. Conf. on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*. Italy, 244-252.

[8] Buddhakulsomsiri, J., Kim, D.S., (2006). Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting, *European Journal of Operational Research*, 175(1), 279–295.

[9] Peteghem, V.V., Vanhoucke, M., (2010). A genetic algorithm for the pre-emptive and non-pre-emptive multi-mode resource-constrained project scheduling problem, *European Journal of Operational Research* , 201(2), 409–418.

[10] Shi, J.J., Deng, Z., (2000). Object-oriented resource-based planning method (ORPM) for Construction, *International Journal of Project Management*, 18(3),179-188.

[11] Nabipoor Afruzi, E., Najafi, A.A., Roghanian, E., Mazinani, M., (2014). A Multi-Objective Imperialist Competitive Algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations, *Computers & Operations Research*, 50, 80-96.

[12] Chen P.H., Weng, H., (2009). A two-phase GA model for resource-constrained project scheduling, *Automation in Construction*, 18(4), 485-498.

[13] Çakır, G., Subulan, K., Yildiz, S.T., Hamzaday, A., Asılkefeli, C., (2022). A comparative study of modeling and solution approaches for the multi-mode resource-constrained discrete time–cost trade-off problem: Case study of an ERP implementation project, *Computers & Industrial Engineering*, 169, 108201.

[14] Nabipoor Afruzi, E., Aghaie, A., Naja, A.A., (2020). Robust optimization for the resource-constrained multi-project scheduling problem with uncertain activity durations, *Scientia Iranica*, 27(1), 361-376.

[15] Li, J., Liu, W., (2005, Sept 19-22). A Hybrid Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem, *IEEE Conference on Emerging Technologies and Factory Automation,*. United States, 414-420.

[16] Moradi Vartouni, A., Mohammad Khanli, L., (2014). A hybrid genetic algorithm and fuzzy set applied to multi-mode resource-constrained project scheduling problem, *Journal of Intelligent & Fuzzy Systems*, 26(3), 1103–1112.

[17] Sharma, A., Tiwari, S., (2021). Fuzzy Multi-Objective Genetic Algorithm Based Resource Constrained Time-Cost Trade-Off Model under Uncertain Environment, *International Journal for Research in Applied Science & Engineering Technology*, 9(8), 1126-1133.

[18] Hosseinian, A.H., Baradaran, V., (2023). A two-phase approach for solving the multi-skill resource-constrained multi-project scheduling problem: a case study in construction industry, *Engineering, Construction and Architectural Management*, 30(1), 321-363.

[19] Selvam, G., Tadepalli, T.C., (2019). Genetic algorithm based optimization for resource levelling problem with precedence constrained scheduling, *International Journal of Construction Management*, 1-10.
[20] Ma, Z., He, Z., Wang, N., Yang, Z., Demeulemeester, E., (2018). A Genetic Algorithm for the Proactive Resource-Constrained Project Scheduling Problem With Activity Splitting, *IEEE transactions on engineering management*, 66(3), 1-16.
[21] Li, H., Demeulemeester, E., (2016).A genetic algorithm for the robust resource levelling problem, *Journal of Scheduling*, 19(1), 43-60.
[22] DAMCI, A., POLAT, G., (2014). Impacts of Different Objective Functions on Resource Levelling in Construction Projects: a Case Study, *Journal of civil engineering and management*, 20(4), 537–547.
[23] Pana, N.H., Hsaiob, P.W., Chena, K.Y., (2008). A study of project scheduling optimization using Tabu Search algorithm, *Engineering Applications of Artificial Intelligence*, 21(7), 1101–1112.
[24] Zhang, Y., Wang, S., Ji, G., (2015). A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications, *Mathematical Problems in Engineering*, 2015(1), 1-38.
[25] Casadei, R., (2023). Artificial Collective Intelligence Engineering: a Survey of Concepts and Perspectives, *Artificial Life*, 29(4), 433–467.
[26] Xia, X., Gui, L., Yu, F., Wu, H., Wei, B., Zhang, Y.L., Zhan, Z.H., (2020). Triple Archives Particle Swarm Optimization, *IEEE TRANSACTIONS ON CYBERNETICS*, 50(12), 4862-4875.
[27] Gad, A.G., (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review, *Archives of Computational Methods in Engineering*, 29, 2531–2561.
[28] Pu, H., Song, T., Schonfeld, P., Li, W., Zhang, H., Hu, J., Peng, X., Wang, J., (2019). Mountain railway alignment optimization using stepwise & hybrid particle swarm optimization incorporating genetic operators, *Applied Soft Computing Journal*, 78, 41-57.
[29] Turkoglu, H., Arditi, D., Polat, G., (2024). Augmented Time–Cost Trade-Off Optimization Using Particle Swarm Optimization, *Journal of Construction Engineering and Management*, 150(5), 04024031.
[30] Gong, Y.J., Zhang, J., Chung, H., Chen, W., Zhan, Z., Li, Y., Shi, Y.H., (2012). An Efficient Resource Allocation Scheme Using Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation*, 16(6), 1-47.
[31] Lin, J.C.W., Lv, Q., Yu, D., Srivastava, G., Chen, C.H., (2023). Adaptive particle swarm optimization model for resource leveling, *Evolving Systems*, 14, 593–604.
[32] Hossain, S.I., Akhand, M.A.H., Shuvo, M.I.R., Siddique, N., Adeli, H., (2019). Optimization of University Course Scheduling Problem using Particle Swarm Optimization with Selective Search, *Expert Systems with Applications*, 127, 9-24.
[33] Elbeltagi, E., Ammar, M., Kassab, M., (2015). Overall multiobjective optimization of construction projects scheduling using particle swarm, *Engineering, Construction and Architectural Management*, 23(3), 265 – 282.