Resource-Constrained Scheduling of Construction Projects Using the Harmony Search Algorithm

Yaghob Gholipour*[1] and Mohammad Mehdi Shahbazi [2]
[1] School of Civil Engineering, College of Engineering, University of Tehran, Tehran, Iran
[2] Engineering Optimization Research Group, College of Engineering, University of Tehran, Tehran, Iran
(Received 20 November 2010, Accepted 2 May 2011)

Abstract

During the implementation, construction projects usually encounter situation that considerably affects the project scheduling and cost. This study aims at using an improved version of the harmony search algorithm (HSA) to schedule resource constrained construction projects. This model is formulated as a global optimization problem. It will determine the duration of each activity to minimize the project total cost. The algorithm tries to find the best duration for each activity so that it leads to the total consumption of the corresponding resource. This may cause some activities to start simultaneously. The improvements have been made to increase the convergence rate and to lower the cost and shorten duration of the project. A numerical example has been proposed to evaluate the efficacy of the algorithm. This algorithm also addresses issues such as multi-resource, resource combination, and resource limit. Two scenarios have been considered for the test problem. The former scenario shows the project scheduled using the minimum duration list and the latter scenario schedules the project using the optimization algorithm. A comparison between the two scenarios shows the effectiveness of the proposed algorithm in decreasing the total cost and duration of the projects.

Keywords: Project scheduling, Project cost flow optimization, Construction projects, Cash flow management

Introduction

Regarding project scheduling, resource constraints are generally considered the most important issue for contractors. Researchers have devoted considerable time and energy to investigate this issue over the past decades. Since adequate cash flow management is directly related to cost control and profit margin for contractors, numerous techniques have been recently developed for cash flow purposes. Due to the size, discreteness and complexity of construction projects, ineffective scheduling and design can result in a disastrous outcome.

In project scheduling, resources are utilized in accordance with activity requirement and implementation time relevant to the cash flow. In other words, the contractors are bound to follow a strict financial guideline to implement any phase of the project. Various financial factors, such as interest rate, payment condition, and credit limit, not only affect the project cost, but also influence the amount and timing of resource usage. Woodworth and Shanahan [1] have shown that time-oriented schedules usually exceed the targeted completion time by an average of around 38%. This is due to the fact that the successful completion of any activity is not only time but also resource dependent. The general resource-constrained project scheduling problem (RCPSP) arises when a set of interrelated activities (precedence relations) are given and when each activity can be performed. Questions arise regarding which resource-duration mode should be adopted, and when each activity begins in order to optimize the project cost. Slowinski and Weglarz [2] proposed a backtracking algorithm for solving duration minimization and net present value (NPV) maximization problem in a precedence and resource-

* Corresponding author: Tel: +98- 21- 61112804 Fax: +98- 21- 66461024 Email: ygpoor@ut.ac.ir

constrained network. Maximizing the NPV of a project was first suggested by Russell [3]. Grinold [4] observed that a simple variable substitution, converts Russell’s formulation into an equivalent linear program. Neumann and Zimmermann [5] extended Grinold’s algorithm to problems with generalized precedence. Elmaghraby and Herroelen [6] proposed an approximate solution procedure to the NPV maximization problem, while Schwindt and Zimmermann [7] developed an exact method based on the steepest ascent principle. Some efforts has been made to apply meta-heuristic algorithms to project scheduling optimization. Lucko [8] employed simulated annealing for optimizing cash flows, while Chen [9] has employed particle swarm algorithm. Geng et al. [10] presented an improved version of ant colony algorithm, which gives better results than the original version. Liao [11] presented the state of the art meta-heuristic algorithms for project scheduling.

1. Activities' relationship

Assume activity A_1 is a predecessor of A_2 and A_3. If the resources are unconstrained, both A_2 and A_3 can begin simultaneously, as soon as A_1 finishes. On the other hand, if there are not enough resources for both activities, one begins immediately and the other one would be delayed. This leads to a longer duration and larger project cost. The proposed algorithm, schedules the project based upon the activity relationships, resource constraints, and financial requirements.

In the proposed model, each activity can choose its own resource requirements and the corresponding duration. Activity duration d_i, is derived from a set of durations D_i; moreover, for each activity, the set D_i, consists of all possible durations complying with various resource requirement. This model provides schedulers with options to chose the best resource-duration mode considering the activities relationship. The finish-start (FS) relationship, which is typically used for construction projects, is shown as Eq. (1).

$$S_{i'} - S_i \geq d_i; \quad i = 1, 2, \ldots, P$$
$$i, i' \in A; \quad \forall d_i \in D_i \qquad (1)$$

where S_i denotes start time of activity i; $S_{i'}$ denotes the start time of activity i', which succeeds activity i; P represents last activity in the project network; d_i is the duration of activity i; A denotes the set of pairs of activities with precedence relationships; and D_i represents the set of all eligible duration for activity i.

2. Resource usage

A common feature of all heuristics is that they assign a resource-duration mode to each activity. For longer activity duration, there is the possibility that parallel activities may begin simultaneously.

The solution we discussed in this research distributes resources among activities so that they would start at the same time and progress in parallel until the resources are totally depleted. The total duration time in this case, may be shorter than the sum of the duration time of all activities if they were not started in parallel. As shown in Figure 1, the required resources decreases, when the duration of d_2 and d_3 increases. Consequently, the resource usage can be controlled to optimally fit the constraints. Eq. (2) summarizes requirements of resource type R_j on day k.

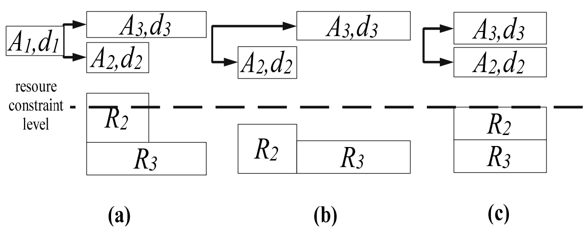

Figure 1: (a): original activity sequences. (b): *A3* begins after *A2* to satisfy the resource constraints. (c): *d2* and *d3* were altered to fit the total resource usage to the constraints

$$\forall k, \quad \sum_{i=1}^{SE_k} R_{ij} \le RL_j \qquad (2)$$
$$\forall i \in SE_k; \quad k = 1,2,\ldots,T; \quad j = 1,2,\ldots,J$$

where R_{ij} represents the required resource for type j and activity i during d_i; RL_j denotes the limit of resource type j per day; SE_k is the set of in progress activities on day k; T denotes project duration; and J is the number of resource types. Resources available for completing tasks can be classified as either renewable or non-renewable. Renewable resources typically have the same amount of availability in every period for an unlimited number of periods, while non-renewable resources are depleted after a certain amount of consumption (or number of periods). In a real construction project, both resource types are employed. For this research we are considering the later type.

3. Project cost flow

Minimizing project duration often carries with it simultaneous increase in project net present value. In other words, early completion of any activity, implies early cash payments. This generally will result in an increase in the amount of net present value [2]. The model formulation complies with the financial terminology employed by Elazouni and Metwally [12] and Au and Hendrickson [13], where payments for the completed activities are made at the end of each period. Typical cash out flows on a construction project include interest, material, labor cost, etc. [14]. Numerous approaches have been developed to investigate the cash flow in construction projects. Barbosa and Pimentel [15] constructed a linear programming model for cash flow management in the Brazilian construction industry. Chiu and Tsai [16] developed a heuristic searching rule to gain a near-optimal solution and incorporated penalty and bonus into the multi-project scheduling problems with a discounted cash flow. Elazouni and Gab-Allah [17] applied integer programming to establish a finance-based scheduling model for minimizing project duration, and presented project schedules with various credit limits. Elazouni and Metwally [12] considered credit limit and scheduled a construction project using improved genetic algorithms for total profit maximization under different credit limit settings. A project schedule, which does not consider cash flow, will overlook the costs associated with interest and payment conditions, and may lead to budget overruns and project failure [14]. In this research, only cash out flows have been considered in calculation of project cost that is equal to the sum of the payments during construction, including the cost of each resource on each day. Regarding the payment for each resource P_{jk}, Eqs. (3) and (4) show the total daily cost flow and total payment at period t, respectively.

$$P_k = \sum_{j=1}^{J} P_{jk} \qquad (3)$$

$$P_t = \sum_{k=1}^{m} P_k \qquad (4)$$

where P_k denotes the total payment of direct cost on day k for resource j; J is the number of resources; P_t denotes the total payment at the end of period t; and m is number of days per period. Eq. (6) shows the overall expenses at the end of period t (E_t), which consists of DC_k, the total direct cost on day k; E_t is summarized expense during period t; O_t is expenses of overheads, mobilization costs, taxes, and bonus at period t; and I_t is the interest charges at the end of period t. Eq. (7) shows the interest charges for each period. As for Eq. (7), it calculates interest until period t-1 and approximates the interest from the end of period t-1 to period t, which is generated from direct costs and other expenses during period t.

$$E'_t = \sum_{k=1}^{m} DC_k + O_t \qquad (5)$$

$$E_t = E'_t + I_t \qquad (6)$$

$$I_t = \left(R \times N_{t-1}\right) + \frac{R}{2} E'_t \qquad (7)$$
$$t \ge 1; \quad k = 1,2,\ldots,T$$

where R represents the interest rate per period; and N_t is the net cash out flow including interest charges at the end of period t. The cumulative cash out flow at the end of period t (F_t) is presented in Eq. (8), and the relationship among net cash out flow, cumulative cost flow, and total payment is as Eq. (9).

$$F_t = N_{t-1} + E_t \tag{8}$$

$$N_{t-1} = F_{t-1} + P_{t-1} \quad t = 1, 2, ..., L \tag{9}$$

where F_t denotes the cumulative cash out flow including interest charges at the end of period t; N_{t-1} is net cash out flow including interest charges at the end of period $t-1$; E_t is summarized expense during period t; and P_{t-1} denotes the total payment at the end of period $t-1$; and L is the last period in the project. Minimizing net cash out flow (N_L) without considering cash inflow, will lead to maximizing profit. Net cash out flow is considered at the end of last period of the project, L, and the objective is to minimize the net cash out flow at the last period, N_L, defined by Eq. (10)

$$\begin{aligned} &Minimize \quad f(d) = N_L \\ &Subject\ to \quad \begin{cases} d_i \in D_i, i = 1, 2, ..., n \\ \forall k, \quad \sum_{i=1}^{SE_k} R_{ij} \le RL_j \end{cases} \\ &\forall i \in SE_k; \quad k = 1, 2, \ldots, T; \quad j = 1, 2, \ldots, J \end{aligned} \tag{10}$$

where $f(d)$ is the objective function; d_i is optimization variable (duration of activity i); D_i is the set of possible range of i^{th} activity duration; and n is the number of decision variables.

4. Optimization algorithm

Many approaches, such as meta-heuristic algorithms (genetic algorithm, neural networks, etc.), mathematical programming, and CP can be employed to address resource-constrained scheduling problems. Meta-heuristic algorithms focus on solving problems in a short time, and provide approximate solutions. At times they can be imprecise and user dependent. Mathematical programming and CP techniques are applied to find an exact solution. Compared with mathematical programming, CP is not restricted by any particular model formulation, such as linear equations [18], and has been widely and successfully applied to complex problems, such as scheduling problems in various fields [19, 20]. As mentioned above, the proposed algorithm prolongs the activity durations to reach the optimum resource usage and minimum project cash out flow. An improved version of HS is employed here to find the optimum duration for each activity.

The original Harmony Search (HS) algorithm is a replication of a musical performance process. A musician's search to find a better state of musical harmony (a perfect state) [21] is similar to optimization process that seeks to find a global solution (a perfect state) as determined by an objective function. The pitch of each musical instrument determines the aesthetic quality; similarly, the set of values assigned to each decision variable determines the value of the objective function. The original HS algorithm consists of the following steps:

1. *Initializing the optimization problem and algorithm parameters*: The parameters of HS algorithm, that is, HMS, HMCR, and PAR are also specified at this step. The Harmony Memory Size (HMS) is the number of solution vectors in the Harmony Memory (HM) and Harmony Memory Considering Rate (HMCR) and Pitch Adjusting Rate (PAR) are parameters that are used to improve the solution vector. HM is a memory location where all the solution vectors (sets of decision variables) are stored which is similar to the genetic pool in GA.
2. *Initializing HM*: HM matrix shown in (11) is filled with as many solution vectors as the value of HMS. These solutions are randomly generated and sorted according to the values of the objective function, $f(d)$.

$$HM = \left[d^1, d^2, \ldots, d^{HMS} \right] \tag{11}$$

3. *Improvise a new harmony from HM*: Memory considerations and pitch

adjustments determine if the new harmony vector $d' = (d'_1, d'_2, \ldots, d'_n)$ should be generated from HM. That is, d'_i (the value of i^{th} variable for the new vector) is chosen from values in HM or D_i .The value of HMCR which varies between 0 and 1 will determine where to choose the possible new value as indicated by Eq. (12).

$$d'_i \leftarrow \begin{cases} d'_i \in \{d_i^1, d_i^2, \ldots, d_i^{HMS}\} & HMCR \\ d'_i \in D_i & (1 - HMCR) \end{cases} \quad (12)$$

The HMCR is the probability of choosing one value from the stored values in HM and (1-HMCR) is the probability of randomly choosing one feasible value, d_i, not limited to those stored in the HM. For example, a HMCR of 0.95 indicates that the HS algorithm will choose the duration of i^{th} activity from stored values in the HM with a 95% probability and from the entire feasible range with a 5% probability. A HMCR value of 1.0 is not recommended because of the possibility that the solution may be improved by values not stored in the HM. This is similar to the reasoning for using mutation rate in the selection process of Genetic Algorithm. Every component of the new harmony vector, $d' = (d'_1, d'_2, \ldots, d'_n)$ is examined to determine whether it should be pitch-adjusted. This procedure uses the PAR parameter that sets the rate of adjustment for the pitch chosen from the HM as shown in Eq. (13).

$$d'_i \leftarrow \begin{cases} YES & w.p. \quad PAR \\ No & w.p. \quad 1 - PAR \end{cases} \quad (13)$$

The pitch adjusting process is performed only after a value is chosen from the HM. The value (1-PAR) implies doing nothing. A PAR of 0.3 indicates that the algorithm will choose a neighboring value with $30\% \times HMCR$ probability. The HMCR and PAR parameters introduced in Harmony Search help the algorithm find globally and locally improved solutions.

4. *Update the HM*: The new harmony vector replaces the worst solution if a better objective function value is obtained. HM is then resorted.

5. *Check if the termination criterion is satisfied*: When the termination criterion is not satisfied steps 3 and 4 are repeated.

HS preserves the history in HM vectors similar to that of Tabu Search and is able to change the adaptation rate (HMCR) through out the computation period, which resembles Simulated Annealing. It also considers several vectors simultaneously in a manner similar to GA. However, the major difference between GA and HS algorithm is that the latter generates a new vector from all the existing vectors (all harmonies in HM), while GA generate a new vector from only two of the existing vectors (parents). For further information about the HS algorithm, see Refs [21-26]. Figure 2 illustrates the procedure of HS algorithm. An improvement has been applied to the original algorithm. In this improvement, D_i is confined in each step according to the last result as shown in Eq (14), Numerical studies shows the effectiveness of this improvement.

$$\begin{aligned} & D'_i = (D'_{i1}, D'_{i2}) \\ & D'_{i1} = \max\{d_i - \frac{\Delta d}{2}, d_i^*\} \\ & D'_{i2} = \min\{d_i + \frac{\Delta d}{2}, D_{in}\} \\ & \Delta d = 2, 4, 6, \ldots; \qquad d'_i \in D'_i \end{aligned} \quad (14)$$

where Δd is the eligible duration range which should be an even integer value; d_i^* is the minimum eligible duration of activity i according to the resource constraints; D_{in} is the upper bound of eligible activity duration list D_i; d_i is the duration of activity i assigned in the previous step; and the d'_i is the new duration of activity i chosen randomly from the list D'_i.

Project profit has been defined as the sum of all expenditures (direct costs and other expenses) and payments for completed activities. Cost flow minimization at the last period, L, leads to the profit maximization of the project. The enumeration procedure starts with the activities scheduled to complete with feasible resource and precedence with left-shifted or early finish times. It examines right-shifting activities in later time periods to permit examination of the resource and objective function trade offs that occur during such scheduling construction with the constraints as shown in Eqs. (1) and (2). The advisable point of the HS algorithm is to find a project schedule, which is close to the d^* (i.e. a vector containing the minimum possible activity durations).

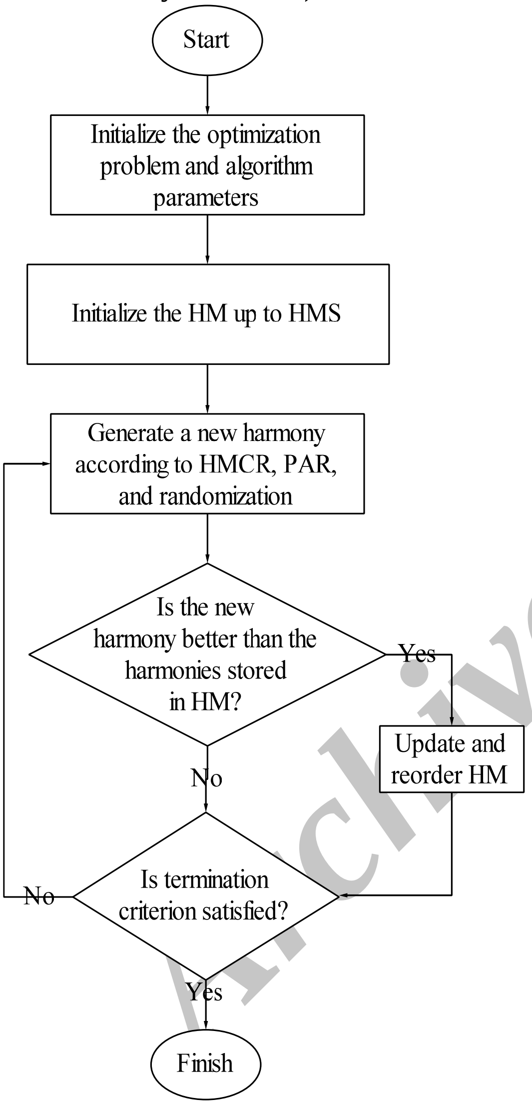

Figure 2: Harmony Search Algorithm optimization procedure

5. Computational experiment

To evaluate the performance of the proposed algorithm in detail, two computational experiment has been conducted. In both examples, $D_i=\{1,2,\ldots,50\}$; $\Delta d=4$; $I_t=0.01$; and HMS=10 are assumed as the initial values. Numerical experiments showed that HMCR=0.50 and PAR=0.90 lead to better results.

5.1. Case study 1

This case is a small project, which contains nine activities and seven resources. It is presented here to show how the algorithm works. Tables **Error! Reference source not found.** and **Error! Reference source not found.** show the activities and resources of forming and pouring concrete of a roof, respectively. Figure 3 shows the corresponding activity-on-node (AON) diagram, in which each node shows an activity A_i and its duration d_i ($i=1,2,\ldots,p$). The resource requirements and predecessors are shown in **Error! Reference source not found.**.

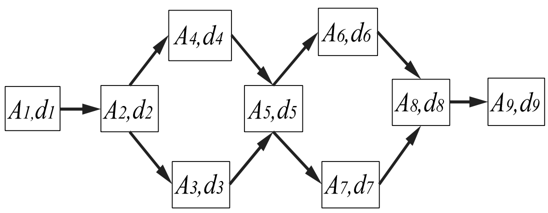

Figure 3: Form and pouring concrete of the first roof: activity-on-node diagram

Table 1: Form and pouring concrete of the first roof: definition of activities and resource requirements for case study 1

Activity	Description	Predecessors	d_i^*	Resources
A_1	Strip column forms	-	1	$2R_2$
A_2	Install rebar of the main beams	A_1	5	$12R_2$,$5R_3$,$0.4R_4$
A_3	Forming the beams	A_2	6	$6R_1$,$8R_2$,$0.2R_4$
A_4	Install secondary beams	A_2	2	$6R_2$,$0.6R_4$
A_5	Putting the roof blocks	A_3,A_4	3	$6R_2$,$800R_5$
A_6	Install rebar of the roof	A_5	2	$5R_2$,$0.5R_3$,$0.5R_4$
A_7	Install stages	A_5	2	$6R_2$
A_8	Pour roof slab	A_6,A_7	5	$8R_2$,$2R_4$,$48R_7$
A_9	Cure roof slab	A_8	1	$3R_2$,R_4,$4R_6$

Table 2: Form and pouring concrete of the first roof: definition of resources and costs for case study 1

Resource	Description	Constraint	Cost
R1	Rough carpenter crew	1	25
R2	Labor crew	4	18
R3	Rebar	1 Ton	800
R4	Engineer	1	30
R5	Roof block	300 Pieces	0.60
R6	Water	10 m3	0.20
R7	Concrete	10 m3	50

Two scenarios have been considered here. Scenario 1 schedules the project using the least possible duration d_i^* (Eq. (15) and **Error! Reference source not found.**).

$$d_i^* = max\{\frac{RL_j}{R_{ij}}\ j = 1,2,\ldots,J\} \qquad (15)$$

where RL_j is the limit of resource type j; d_i^* is the least duration of activity i; R_{ij} is the required resource type j for activity i; and J is the number of resource types. An interest rate of $I_t=0.01$ is assumed for both scenarios. For scenario 1, the total cost was obtained $Cost_1=9566$ units. Figure 4 shows the corresponding Gantt chart.

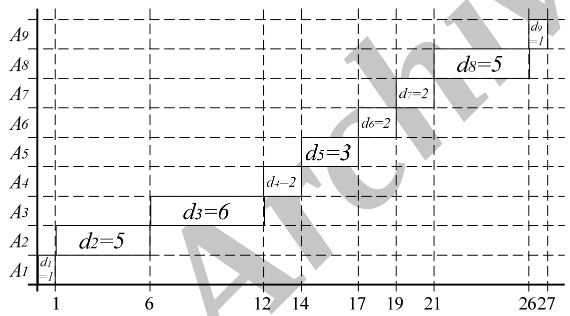

Figure 4: Gantt chart of the scenario 1 - project example 1

Scenario 2 is the optimal schedule obtained from the proposed algorithm, which schedules the project with $Cost_2=9456$ units. Figs 4 and 5 show the corresponding Gantt chart diagram of scenarios 1 and 2, respectively. Although the algorithm increased the duration of activities A_4, A_6, and A_7, but the total cost and duration decreased. This demonstrates the efficiency of the algorithm.

Figure 6 shows the convergence diagram of the improved algorithm in comparison to the original algorithm. The convergence rate for the proposed algorithm is has been greatly improved.

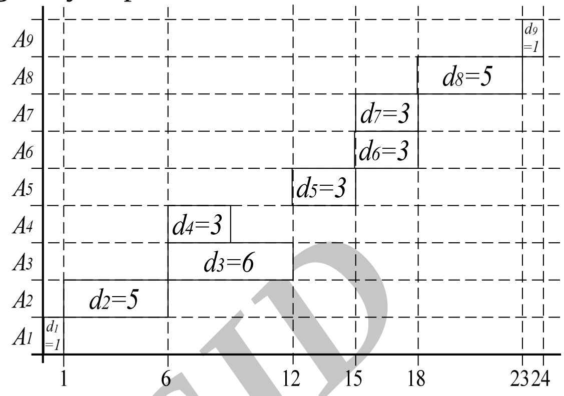

Figure 5: Optimal schedule Gantt chart corresponding to the scenario 2 - project example 1

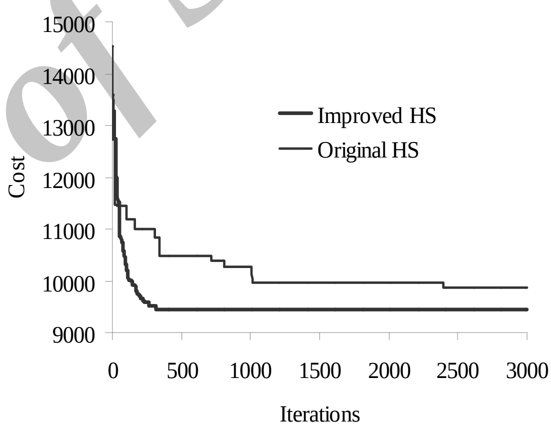

Figure 6: Convergence diagram of the improved algorithm in comparison to the original algorithm – Example 1

5.2. Case study 2

This case is based on a real construction project, which shows the efficiency of the algorithm in large and real projects. It is containing 32 activities and 7 resources. **Error! Reference source not found.** and Figure 7 show the project definition and obtained results. It should be noticed that the improvement has a better influence on larger and more complex projects.

Table 3: Project definition and result of original and improved HS algorithm for case study 2

Activity	Description	Predecessors	d_i^*	d_i (Original HS)	d_i (Improved	Resources
A_1	General excavation	-	4	4	5	$4R_2,8R_7$
A_2	Excavation of additional 2 m thick marine mud	A_1	2	2	2	$2R_2,4R_7$
A_3	Deposition and compaction of 2 m thick additional rock fill materials	A_2	2	3	2	$2R_2,R_3,4R_7,2R_8$
A_4	Placing and compaction of 400 mm thick rock fill	A_3	2	8	2	$R_2,4R_7,2R_8$
A_5	Laying of 75 mm thick blinding concrete	A_4	3	26	9	$R_3,2R_5,6R_7$
A_6	Fixing of steel reinforcement for base slab & side walls (lower part)	A_5	4	20	8	$2R_1,4R_7$
A_7	Erection of formwork for base slab & side walls (lower part)	A_5	2	5	8	$R_4,4R_7$
A_8	Concreting of base slab & side walls (lower part)	A_6,A_7	3	13	3	$R_3,3R_5,4R_7$
A_9	Erection of false work for top slab	A_8	3	5	5	$6R_7$
A_{10}	Fixing of steel reinforcement for top slab & side walls (upper part)	A_9	4	7	4	$2R_1,R_3,3R_7$
A_{11}	Erection of formwork for top slab & wide walls (upper part)	A_9	2	16	10	$R_4,4R_7$
A_{12}	Concreting of top slab & side walls (upper part)	A_{10},A_{11}	3	4	8	$R_3,3R_5,4R_7$
A_{13}	Placing and compaction of 400 mm thick rock fill	A_1	2	3	5	$R_2,3R_7,2R_8$
A_{14}	Laying of 75 mm thick blinding concrete	A_{13}	2	3	3	$R_3,2R_5,4R_7$
A_{15}	Fixing of steel reinforcement for base slab & side walls (lower part)	A_{14}	4	13	5	$2R_1,3R_7$
A_{16}	Erection of formwork for base slab & side walls (lower part)	A_{14}	2	14	7	$R_4,4R_7$
A_{17}	Concreting of base slab & side walls (lower part)	A_{15},A_{16}	3	5	4	$R_3,3R_5,4R_7$
A_{18}	Erection of false work for top slab	A_{17}	3	10	6	$5R_7$
A_{19}	Fixing of steel reinforcement for top slab & side walls (upper part)	A_{18}	4	5	6	$2R_1,R_3,R_7$
A_{20}	Erection of formwork for top slab & wide walls (upper part)	A_{18}	2	25	7	$R_4,4R_7$
A_{21}	Concreting of top slab & side walls (upper part)	A_{19},A_{20}	3	16	4	$R_3,3R_5,3R_7$
A_{22}	Placing and compaction of 400 mm thick rock fill	A_{21}	1	4	4	$R_2,2R_7,R_8$
A_{23}	Laying of 75 mm thick blinding concrete	A_{22}	2	7	8	$R_3,2R_5,2R_7,R_8$
A_{24}	Fixing of steel reinforcement for base slab & side walls (lower part)	A_{23}	2	13	3	$R_1,2R_7$
A_{25}	Erection of formwork for base slab & side walls (lower part)	A_{24}	2	9	6	$R_4,4R_7$
A_{26}	Concreting of base slab & side walls (lower part)	A_{25}	3	6	8	$R_3,3R_5,2R_7,R_8$
A_{27}	Erection of false work for top slab	A_{26}	2	29	6	$4R_7$
A_{28}	Fixing of steel reinforcement for top slab & side walls (upper part)	A_{27}	4	7	12	$R_3,2R_7$
A_{29}	Erection of formwork for top slab & wide walls (upper part)	A_{27}	2	24	12	$R_4,4R_7$
A_{30}	Concreting of top slab & side walls (upper part)	A_{28},A_{29}	3	35	4	$R_3,3R_5,2R_7,R_8$
A_{31}	Positioning of pre-cast concrete pipes at end wall	A_{30}	3	31	42	$R_3,R_6,6R_7$
A_{32}	Backfilling & compaction	A_{12},A_{31}	4	8	14	$4R_2,8R_7,4R_8$
Total Cost			7877	8676	7544	
Total Duration			87	151	100	

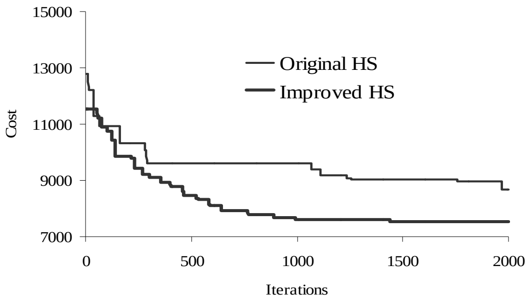

Figure 7: Convergence diagram for project example 2

The results show efficiency of the improvement on real and large size projects. As can be seen, the optimum result of the improved version has shorter project duration but lower cost, which indicates that the algorithm chosen a better resource-duration mode than the mode corresponds to d_i^*. in other words, the minimum duration list, d_i^*, may not lead to the minimum cost.

6. Conclusion

This research aims at decreasing the computational cost of a meta-heuristic algorithm by confining the eligible activity duration in each step and decreasing the project cost and duration that are major concern for contractors. The advantages of the proposed model lies in engaging the issues of both resource constraints and cash flow consideration, and thus providing an overall performance assessment while minimizing project cash out flow. This is done by finding the optimum resource-duration mode using an improved version of HS algorithm, which shown it had better performance than the original version. The improvement is to confine the eligible duration for each activity in each step Numerical investigations showed that the improved version had higher convergence rate and lower total cost compared to the original version. The efficacy of the algorithm has been evaluated in detail.

Acknowledgements

The authors acknowledge gratefully the Supports of the College of Engineering of University of Tehran.

References:

1- Woodworth, B.M. and Shanahan, S. (1988). "Identifying the critical sequence in a resource constrained project." *International Journal of Project Management*, Vol. 6, No. 2, PP. 89-96

2- Patterson, J.H., Brian Talbot, F., Slowinski, R. and Weglarz, J. (1990). "Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. "*European Journal of Operational Research*, Vol. 49, No. 1, PP. 68-79

3- Russell, A.H. (1970). "Cash Flows in Networks." *Management Science*, Vol. 16, No. 5, PP. 357-373

4- Grinold, R.C. (1972). "The payment scheduling problem." *Naval Research Logistics Quarterly*, Vol. 19, No. 1, PP. 123-136

5- Neumann, K. and Zimmermann, J. (2000). "Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints." *European Journal of Operational Research*, Vol. 127, No. 2, PP. 425-443

6- Elmaghraby, S.E. and Herroelen, W.S. (1990). "The scheduling of activities to maximize the net present value of projects." *European Journal of Operational Research*, Vol. 49, No. 1, PP. 35-49

7- Schwindt, C. and Zimmermann, J. (2001). "A steepest ascent approach to maximizing the net present value of projects." *Mathematical Methods of Operations Research*, Vol. 53, No. 3, PP. 435-450

8- Lucko, G. (2011). "Optimizing cash flows for linear Schedules modeled with singularity functions by simulated annealing." *J. Constr. Eng. Manage*.-ASCE, Vol. 137, No. 7, PP. 523-535

9- Chen, R.M. (2011). "Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem." *Expert Syst. Appl*., Vol. 38, No. 6, PP. 7102-7111

10- Geng, J.Q., Weng, L.P. and Liu, S.H. (2011). "An improved ant colony optimization algorithm for nonlinear resource-leveling problems." *Comput. Math. Appl*., Vol. 61, No. 8, PP. 2300-2305

11- Liao, T.W., Egbelu, P.J., Sarker, B.R. and Leu, S.S. (2011). "Metaheuristics for project and construction management - A state-of-the-art review. "*Automation in Construction*, Vol. 20, No. 5, PP. 491-505

12- Elazouni, A.M. and Metwally, F.G. (2005). "Finance-based Scheduling: tool to maximize project profit using Improved genetic algorithms. "Finance-Based Scheduling: Tool to Maximize Project Profit Using Improved Genetic Algorithms. Vol. 131. ASCE. 400-412

13- Au, T. and Hendrickson, C. (1986). "Profit Measures for Construction Projects." *Profit Measures for Construction Projects,* Vol. 112. ASCE. 273-286

14- Liu, S.-S. and Wang, C.-J. (2008). "Resource-constrained construction project scheduling model for profit maximization considering cash flow." *Automation in Construction*, Vol. 17, No. 8, PP. 966-974

15- Barbosa, P.S.F. and Pimentel, P.R. (2001). "A linear programming model for cash flow management in the Brazilian construction industry. "*Construction Management and Economics*, Vol. 19, No. 5, PP. 469-479

16- Huan Neng, C. and Deng Maw, T. (2002). "An efficient search procedure for the resource-constrained multi-project scheduling problem with discounted cash flows." *Construction Management & Economics*, Vol. 20, No. 1, PP. 55-66

17- Elazouni, A.M. and Gab-Allah, A.A. (2004). "Finance-based Scheduling of construction projects using integer programming. "Finance-Based Scheduling of Construction Projects Using Integer Programming. Vol. 130. ASCE. 15-24

18- Heipcke, S. (1999). "Combined modelling and problem solving in mathematical programming and constraint pro-gramming. "Combined Modelling and Problem Solving in Mathematical Programming and Constraint Pro-gramming

19. Fruewirth, T. and Abdennadher, S. (2003). "Essentials of constraint programming." Essentials of Constraint Programming. Springer-Verlag New York, Inc. 146

20- Tsang, E. (2003). "Constraint based Scheduling: applying constraint programming to Scheduling problems." *Journal of Scheduling*, Vol. 6, No. 4, PP. 413-414

21- Lee, K.S. and Geem, Z.W. (2004). "A new structural optimization method based on the harmony search algorithm." *Computers & Structures*, Vol. 82, No. 9-10, PP. 781-798

22- Mahdavi, M., Fesanghary, M. and Damangir, E. (2007). "An improved harmony search algorithm for solving optimization problems." *Applied Mathematics and Computation*, Vol. 188, No. 2, PP. 1567-1579

23- Geem, Z.W. (2009). "Music-inspired harmony search algorithm: theory and applications." Music-Inspired Harmony Search Algorithm: Theory and Applications. Springer Publishing Company, Incorporated

24- Zong Woo Geem, Joong Hoon Kim and Loganathan, G.V. (2001). "A new heuristic optimization algorithm: harmony search. "SIMULATION, Vol. 76, No. 2, PP. 60-68

25- Lee, K. and Geem, Z. (2005). "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice." *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 36-38, PP. 3902-3933

26- Geem, Z. (2008). "Novel derivative of harmony search algorithm for discrete design variables." *Applied Mathematics and Computation*, Vol. 199, No. 1, PP. 223-230