Associate Professor Esmaeil MEHDIZADEH, PhD
E-mail: emehdi@qiau.ac.ir (Corresponding author)
Lecturer Hamidreza AKBARI, MSc
E-mail: hrakbari@yahoo.com
Department of Industrial Engineering, Faculty of Industrial & Mechanical Engineering, Qazvin Branch, Islamic Azad University
Qazvin, Iran

A NOVEL VIBRATION DAMPING OPTIMIZATION ALGORITHM FOR RESOURCE CONSTRAINED MULTI- PROJECT SCHEDULING PROBLEM

Abstract. *In this paper, we propose a Vibration Damping Optimization (VDO) algorithm with resonator loop as a meta-heuristic algorithm for solving resource constrained multi-project scheduling problem (RCMPSP). The objective is to determine the start time of the projects activities such that the total completion time of processes under the existing constraints would be minimized. This is the first attempt to develop a VDO algorithm for solving the RCMPSP. Also, a new solution representation scheme in a matrix form and special solution procedures are proposed. We explain the elements of the algorithm and solve some problems generated for this model including large size and small size instances. The performance of our proposed algorithm is evaluated by comparison with Simulated Annealing (SA) algorithm. The response surface methodology (RSM) is applied for tuning the parameters of the algorithms. The promising computational results validate the effectiveness of the proposed algorithm.*

Keywords: *multi-project; scheduling problem; resource constraints; project management; vibration damping optimization.*

JEL Classification: M11, C44, C61

1. Introduction

Resource constrained multi project scheduling problem (RCMPSP) is the generalization of resource constrained project scheduling problem (RCPSP) which is used to schedule project activities. The objective is to find an assignment of start time for activities with the given precedence and resource constraints such that makespan of the project minimized. RCMPSP is used in many large scale construction, transportation and manufacturing project management.

The RCMPSP is strongly NP-hard, that is, the time required to obtain a solution exponential increases with the size of the problem (Blazewicz et al, 1983). In the past, three approaches have been used for solving RCMPSP: exact, heuristic and meta-heuristic approaches. Exact approaches seek the optimum solutions. However they cannot find good solutions in reasonable computation time for large and complex projects. For instance, Pritsker et al. (1969) proposed a zero-one programming as an exact approach. Demeulemeester & Herroelen (1992) introduced a branch-and-bound procedure and Vercellis (1994) proposed a Lagrangean decomposition approach. More information about exact methods can be obtained from Kolisch et al, 1998.

Heuristic and Meta-heuristic method gives the near-best solution for large projects at a reasonable time and it is deployed more often but do not always give the best result for solving large size projects (Kolisch& Hartmann, 1998). Mize (1964) introduced a heuristic for scheduling model for multi-project organizations. Kurtulus and Davis (1982) proposed categorization of heuristic rules performance for multi-project scheduling. Their research provides a categorization process based on two powerful project summary measures. The first measure identifies the location of the peak of total resource requirements and the second measure identifies the rate of utilization of each resource type. Lova et al. (2000) developed a multi-criteria heuristic that lexicographically improved two criteria, mean project delay and project splitting. The multi-criteria heuristic algorithm consists of several algorithms based on the improvement of multi-project feasible schedules. Through an extensive computational study, they showed that their method improves the feasible multi-project schedule obtained from heuristic methods based on the priority rules as well as project management software. For other related studies in the heuristic, see Lova & Tormos, 2001. Kumanan et al. (2006) proposed the use of a heuristic and a meta-heuristic algorithm for scheduling a multi-project environment. Their method first identifies projects priority using meta-heuristic algorithm and then the priority of the activities are set by heuristic rules. Goncalves et al. (2008) presented a genetic algorithm for the resource constrained multi-project scheduling problem. The schedules were constructed using a heuristic that builds parameterized active schedules based on priorities, delay times, and release dates defined by the genetic algorithm. Chen and Shahandashti (2009) solved RCMPSP with a hybrid of genetic algorithm and simulated annealing (GA-SA Hybrid). The proposed GA-SA Hybrid is compared with the modified simulated annealing method (MSA), which is more powerful than genetic algorithm (GA) and simulated annealing (SA). Browning and Yassine (2010) addressed the static *resource-constrained multi-project scheduling problem* (RCMPSP) with two lateness objectives, project lateness and portfolio lateness. They conducted a comprehensive analysis of 20 priority rules on 12,320 test problems generated to the specifications of project, activity, and resource-related characteristics—including network complexity and resource distribution and contention. They found several situations in which widely advocated priority rules perform poorly. They also confirmed that portfolio managers and project managers will prefer different priority rules depending on their local or global objectives. Singh (2014) presented a hybrid algorithm that integrates the project priority (or criticality) with project schedule development for multi-project scheduling problem under resource constrained situation. The objective was to minimize the project make-span as well as the penalty cost when some projects carry higher priority. The project schedule is generated using a hybrid algorithm based on priority rules and AHP. Besikci et al. (2015) introduced a multi-project problem environment which involves multiple projects with assigned due dates; activities that have alternative resource usage modes; a resource dedication policy that does not allow sharing of resources among projects throughout the planning horizon and a total budget. The multi-project environment is modeled in an integrated fashion and designated as the resource portfolio problem. A two-phase and a monolithic genetic algorithm proposed as two solution approaches, each of which employs a new improvement move designated as the combinatorial auction for resource portfolio and the combinatorial auction for resource dedication.

In this paper, vibration damping optimization (VDO) algorithm with resonator loop has been applied as a novel VDO algorithm to solve resource constrained multi- project scheduling problem (RCMPSP) which is the first attempt to develop a VDO algorithm for solving the RCMPSP. The VDO algorithm is one of the new recent meta-heuristic algorithms which was introduced by Mehdizadeh and Tavakkoli-Moghaddam (2009) and extended by Mehdizadeh et al. (2015). For more information please see Hajipour et al, 2014. As a first attempt to VDO application for project scheduling problems, Mehdizadeh and Nezhad-Dadgar (2014) applied it to solve the resource constrained project scheduling problem (RCPSP) with weighted earliness-tardiness penalties.

The main innovations in this paper (to differentiate our efforts from those already published on the subject) are as follows:

- Applying a meta-heuristic algorithm named Vibration damping optimization (VDO) algorithm with resonator loop for solving RCMPSP.
- Designing a new solution representation scheme in a matrix form and special solution procedures are proposed.
- Tuning the parameters of the proposed algorithm with response surface methodology (RSM).

The rest of the paper is organized as follows: Problem definition and formulation are described in Section 2 in detail. The proposed meta-heuristic algorithm is given in Section 3. Tuning the parameters is provided in Section 4. Comparison of the algorithms and Analyzing is presented in Section 5. Finally, conclusions are given in Section 6.

2. Problem description

The problem consists of P projects. Project j includes N activities. There are K kinds of resources. The total availability of the resource k is R_k. Activity preemption is not allowed. Activity i can start when all its predecessors are completed. The objective function is to minimize the largest finish time of activities which is formulated as follows:

$$Minimize\ Max\ \{f(p,n) \mid p = 1 \dots P, n = 1 \dots N_p\} \tag{1}$$

Where

$f(p,n)$is the finish time of activity n of project p, and P and N_p are the number of projects and activities in project p, respectively.

The following assumptions are considered for the multiple resource allocation process.

1. Resources are positive integers.
2. Preemption is not allowed. Activities cannot be split.
3. Precedence relationships among activities should be identified. Precedence relationships are obtained using the critical path method (CPM).
4. Projects are independent from each other and there isn't any priority among them.
5. More than one project can start together.
6. Duration of activities is deterministic and specified.
7. All activities are executed in one mode and relations between them are finish to start with zero lag time.

8. The upper limit of available resources and quantity usage of any resource by any activity must be identified.

Based on aforementioned above, Minimizing the completion time in RCMPSP has been widely studied from both exact and heuristic points of view and is denoted by *m, 1|cpm|Cmax* (Demeulemeester & Herroelen,1992). Resource limitations makes project scheduling a combinatorial problem that is solved by exact methods only in the case of small projects or specialized structures. But companies frequently manage various projects including medium and large sized simultaneous sharing a pool of renewable resources. RCMPSP addresses this problem using mathematical model described and adapted by Christofides et al. (1987).

Parameters and decision variables

S_{ijt}: is 1 if activity $j$of project i starts at time t and 0 otherwise

M: The number of projects in the multi-project

J_i: The number of activities of project i

K*:* The number of renewable resource types

Model formulation:

$$Minimize\ f(time) \tag{1}$$

$$subject\ to \qquad \sum_{t} S_{ijt} = 1 \quad , \qquad i = 1, \dots, M, \quad J = 1, \dots, i_i + 1 \tag{2}$$

$$\sum_{t} t(S_{imt} - S_{ijt}) \geq d_{ij} \quad , \qquad (j, m) \in H_i, i \tag{3}$$

$$\sum_{i=1}^{M} \sum_{j=1}^{J_i} \sum_{q=t-d_{ij}+1}^{t} r_{ijk} S_{ijq} \leq R_k \ , k = 1, \dots, K, \ \ t \tag{4}$$

$$S_{ijt} \in (0,1) \tag{5}$$

The objective function (1) minimizes performance with respect to time criterion. Equation (2) indicates that every activity must start only one time.

Equation (3) is the precedence constraint where H_i is the set of pairs of activities with precedence constraints in project i. Activity $J_i + 1$ corresponds to the dummy activity where the feasible completion time of project i is obtained and d_{ij} is the duration of activity $j$of project i. Constraint (4) limits (for each resource type $k$and each time instant t) the resource demand of the activities which are currently being processed so that it does not exceed the availability R_k where r_{ijk} is the requirement of the resource k for the activity $j$of project $i$and $T$is an upper bound on the feasible completion time of the multi-project. Finally constraint (5) defines the decision variables as binary.

When talking about data clustering, there are a few basic concepts which need to be discussed, such as distance metric, similarity matrix and clustering algorithms. Conventional clustering methods mainly consist of two parts: the construction of a similarity matrix between documents and the construction of clusters using a clustering algorithm.

A distance metric (Li et. al., 2003) is defined as a function which establishes the distances between the elements of a data set X. Once a distance metric has been chosen for measuring the distances between the elements of a dataset, the similarity or distance matrix is computed, containing the distances among the *n* objects, taken two by two (Grünwald and Vitanyi, 2004). It is a symmetric *n x n* matrix containing positive real numbers, normalized between 0 and 1.

3. Vibration damping optimization algorithm

Vibration damping optimization (VDO) was initially proposed by Mehdizadeh and Tavakkoli-Moghaddam (2009) which is based on the damping process in vibration mechanics. In the following subsections, we will use this procedure to develop an efficient VDO to solve the model at hand.

3.1. Structure of coding

The solution representation is an important component of any meta-heuristic algorithm. It has to be designed such that it is easy to generate a neighbor and calculate the value of objective function quickly. It must also guarantee accessibility for the entire solution space. The activities could be shown as a one dimensional array of X numbers, where X the number of cells is equal to the number of activities for all the projects as shown in Eq (6).

$$X = \sum_{j=1}^{P} N_j \qquad (6)$$

Where N_j is the number of activities in the jth project and P is the number of projects. Each number in the string is dedicated to one activity. Representation of coding is displayed in Fig.1.

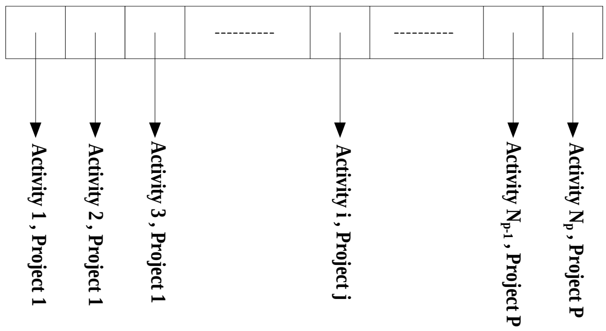

Figure1. Representation of coding

3.2. Generation of an initial solution

Schedule generation schemes (SGS) are the core of most heuristic solution procedures for the RCPSP. SGS starts from scratch and build a feasible schedule through stepwise expansion of a partial schedule. There are two schemes namely serial and parallel. The initial solution is generated randomly using a defined structure and applying the parallel SGS to it. For each expanding iteration g there is a schedule time t_g. Activities which have been scheduled up to g are either element of the complete set C_g or those of the active set A_g. The complete set comprises all the activities which have been completed up to t_g. $C_g = \{j \in J \,|F_j \leq t_g\}$, F_j denote the finish time of activity.The active set comprises all activities which are active at t_g,$A_g = A(t_g) = \{j \in J | F_j - p_j \leq t \leq F_j\}$ where $p_j$is duration of activity j. The eligible set $D_g$comprises all activities where all precedence activities are completed and have no precedent activity att_g

$D_g = \{j \in J | (C_g \cup A_g) | P_j \subseteq C_g \wedge r_{j,k} \leq, A_g,\ \tilde{R}_k(t_g)\ (k \in K)\}$. The remaining capacity at t_g is $\tilde{R}_k(t_g) = R_k - \sum_{j \in A_g} r_{j,k}$, an algorithmic description of the parallel SGS can be given as follows in Fig.2:

$\boldsymbol{Initialization}: g = 0, t_g = 0, A_0 = \{0\}, C_0 = \{0\}, \tilde{R}_k(0) = R_k$

$While\ |A_g \cup C_g| \leq n\ do$

(1) $g = g + 1$

$t_g = min_{j \in A_g \{F_j\}}$

$Calculate\ C_g, A_g, \tilde{R}_k(t_g), D_g$

(2) $While\ D_g \neq \emptyset \quad do$

$Select\ one\ j \in D_g$

$F_j = t_g + p_j$

$Calculate\ A_g, \tilde{R}_k(t_g), D_g$

$F_{n+1} = max_{h \in p_{n+1} \{F_h\}}$

Figure 2.The pseudo code of the parallel SGS algorithm

The initialization sets the schedule time to 0, assigns the start time of the activity for the active and completed set and sets the available capacity. Each iteration of parallel SGS involves two steps: (1) Determining the next schedule time t_g, the associated activity sets C_g, A_g, D_g and the available capacity $\tilde{R}_k(t_g)$. (2) Scheduling a subset of the eligible activities to start at t_g. This resource assignment algorithm is expanded for RCMPSP and finally calculates the value of objective function.

3.3. Generation of neighborhood solution

The neighborhood generation is performed as follow:

current feasible list is randomly selected and the positions of its latest predecessor and earliest successor is calculated for each project (Fig. 3).The selected activity can be moved anywhere within these two positions without disturbing the precedence constraints. The new position is also randomly chosen. When a move is possible, the new list is obtained by a cyclical shift of all the activities placed between the old and the new positions (Fig.4).

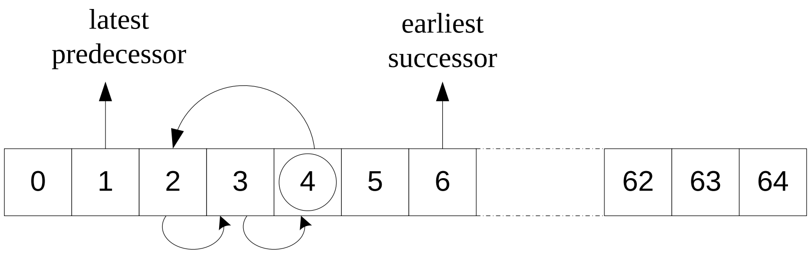

Figure3.Activity 4 and its new position are selected randomly (Swap)

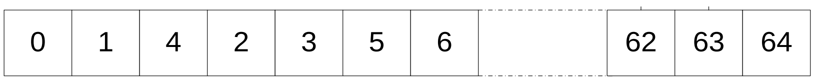

Figure 4. Generation of neighborhood solution

3.4. General framework of the procedures

Table1 shows the correspondence between the elements of optimum problem and the laws of physical vibration.

Table1. Analogy between COP and vibration damping optimization

Vibration damping optimization	Combinatorial Optimization Problem
Process states	Feasible solution
Energy	Cost
Change of state	Neighboring solution
Amplitude	Control parameter
Vibration damping	Heuristic solution
Degrees of freedom	Number of decision variables

The VDO algorithm repeats an iterative neighbor generation procedure and follows search directions that improve the value of the objective function. While exploring solution space to escape from local optimum, the VDO method provides the possibility of accepting the worse neighbor solutions. Each iteration of VDO for current solution calculates the objective function characterized by $f(x)$ and a neighbor x' is generated. $N(x)$, is the objective function for the set of all immediate neighbors. For each move the objective difference $\Delta E = N(x') - F(x)$is evaluated. For minimization problem x' replaces with$x$ whenever $\Delta E \leq 0$. Otherwise x'could also be accepted with a $P(A) = 1 - e^{-A^2/2\sigma^2}$probability. The acceptance probability is compared to a number $y_{random} \in [0,1]$generated randomly. $x'$is accepted whenever $P > y_{random}$. The factors that influence on acceptance probability are value of amplitude and the parameter σ.

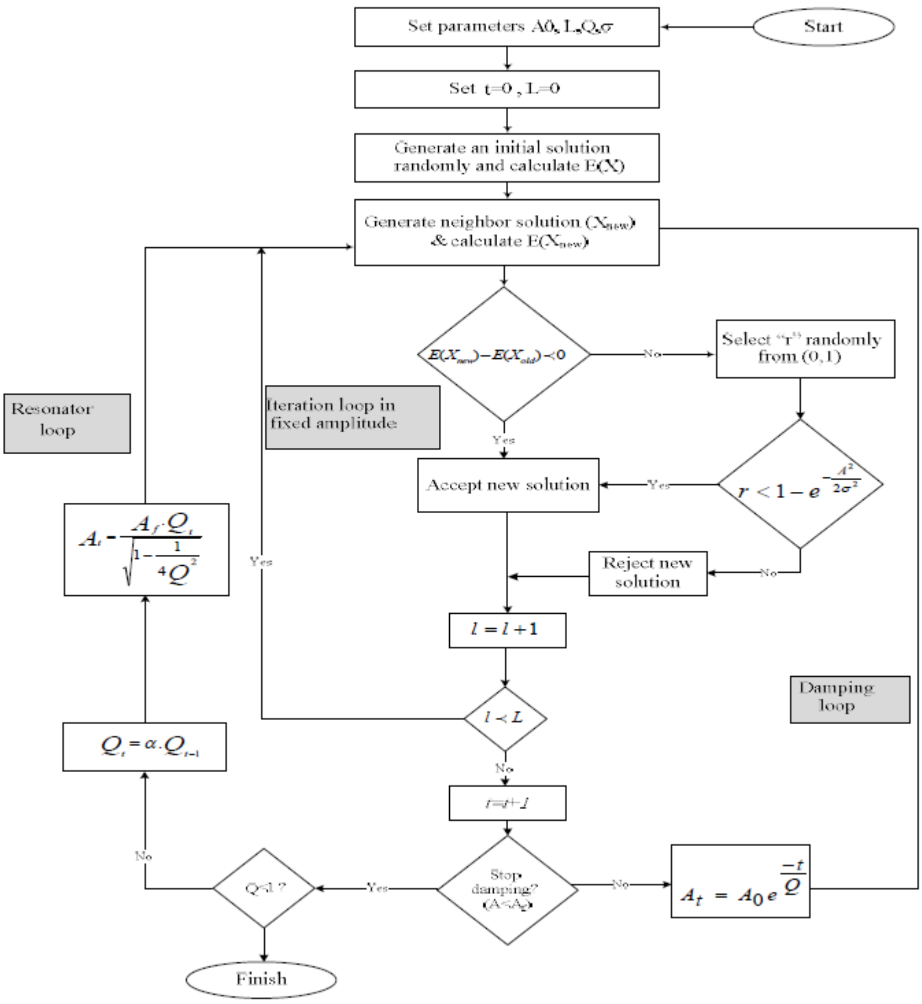

Figure 5. Flow of VDO with Forced Vibration

The value of amplitude can be controlled by damping strategy. Specifying how it should be progressively reduced to make the procedure more selective as the search progresses to neighborhoods of good solutions. There exist theoretical schedules guaranteeing asymptotic convergence toward the optimal solution. However, they require infinite computing time. In practice much simpler and finite computing time schedules are preferred even if they do not guarantee an optimal solution. The flow of the VDO is shown in Fig.5.

The loop resonator to force the algorithm has been added to escape the local optimum point. Toward the end of algorithm the amplitude decreases. The second loop makes for late stage of algorithm and increases the amplitude. This will increase the likelihood for probability function to accept bad solution and decreases the risk of falling in to local optimum solution. The pseudo code of the VDO algorithm is shown in Fig.6.

$t = 0\,, A = A_0\,, X_{best} = \emptyset\,, Generate\ X_0, X_{best} = X_0$

$Do\ (Damping\ Loop)$

$\quad n = 0$

$\quad Do\ (Force\ Loop)$

$\quad\quad Select\ a\ move\ randomly\ and\ run\ over\ X_n\ as{:}\ \Delta E$

$\quad\quad If \Delta E < 0\ then\ X_{best} = X_{new}\ and\ n = n + 1\ and\ X_n = X_{new}$

$\quad\quad\quad Else$

$\quad\quad\quad\quad Generator\ \rightarrow U[0,1]\ Randomly\,, Set\ z = 1 - e^{-A^2/2\sigma^2}$

$\quad\quad\quad\quad If\ r < z\ then\ n = n + 1\ \&\ X_n = X_{new}\,, End\ if$

$Loop\ \ while\ \ n < N$

$t = t + 1, A_t = A_0 e^{-t/Q}$

$Loop\ While\ (t < T\ \& A_t > 0)$

$Print\ \ X_{best}$

Figure 6.The pseudo code of the VDO algorithm

4. Tuning the parameters

Response surface methodology (RSM) is a collection of statistical and mathematical techniques used for optimization. It is useful particularly in situations where several input variables potentially influence on some performance measures or quality characteristics (Myers & Montgomery, 1995). RSM is applied to find the best VDO parameters as input variables such that the deviation from optimal solutions (Y_1 as a response) is optimized. As indicated previously primary amplitude, parameter of Rayleigh distribution, decrement coefficient, and Number of iterations per amplitude are considered as input variables. Table 2 presents parameters and the levels of the input variables. In the RSM application the values of A_0, σ, Q, and L are coded as X_1, X_2, X_3 and X_4 where each one is given the values of -1, 0, and1 for their low, middle and high levels respectively.

Table2. Parameters and their levels

Parameter	Range	Low	Middle	High
Primary amplitude(A_0)	60-200	60	130	
Parameter of Rayleigh distribution(σ)	15-50	15	32.5	
Decrement coefficient(Q)	80-300	80	190	
Number of iterations per amplitude(L)	50-100	50	75	

In order to generalize the statistical results, a set of 36 test problems were considered. Each problem contained either 2,5,10 or 20 and each project had either 30 and 90 or 120 non-dummy activities. Each test problem had four resource types. In order to eliminate the effects of different dimensions of the test problems and to obtain normal responses, a ratio of the lowest response with respect to all answers is obtained. These ratios change the nature of objective function from minimizing to maximizing. A central composite design (CCD) of 2^{4-1}a fractional factorial with four central points is chosen for the experiment. Also 2×4axial points are added to analyze the significant curvature on the response surfaces. Each level of testing is carried out 10 times and the normalized results are recorded (Table 3). The results of Table 3 are used to estimate the second order models for response. The fitted response are shown in Eq.(7).

$$Y = 0.99174 + 0.001444X_2 + 0.001699X_3 - 0.0015X_2X_2 + 0.002095X_4X_4 + 0.002323X_1X_4 \quad (7)$$

Table3. Results of the RSM experiments

RUN	Factors & Levels				Response variable
	$X_1 = A_0$	$X_2 = \sigma$	$X_3 =$	$X_4 = L$	Y
1	60	15	300	100	0.985163
2	60	50	300	50	0.994839
3	60	50	80	100	0.989896
4	60	15	80	50	0.992361
5	130	33	190	75	0.99088
6	130	32.50	190	75	0.988913
7	200	50	80	50	0.990388
8	200	15	300	50	0.993186
9	130	32.50	190	75	0.992361
10	200	15	80	100	0.993681
11	130	32.50	190	75	0.991867
12	200	50	300	100	0.996332
13	12.28	32.50	190	75	0.992361
14	247.73	32.50	190	75	0.995171
15	130	3.07	190	75	0.985001
16	130	61.93	190	75	0.992526
17	130	32.50	5	75	0.987769
18	130	32.50	375	75	0.999665
19	130	32.50	190	32.96	0.999845
20	130	32.50	190	117.05	0.997996

The analysis of variance results for the responses are given in Table 4. The Eq(7) is solved by Lingo and Finally the optimum values of the VDO parameters are obtained and presented in Table 5.

Table4. Analysis of variance for the accuracy performance index(Y)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Regression	11	0.000256	0.000256	0.000023	2.75	0.081
Linear	4	0.000092	0.000092	0.000023	2.72	0.106
Interaction	4	0.000107	0.000107	0.000027	3.14	0.079
Square	3	0.000057	0.000057	0.000019	2.25	0.16
Residual Error	8	0.000068	0.000068	0.000008		
Lack-of-Fit	5	0.000061	0.000061	0.000012	5.24	0.102
Pure Error	3	0.000007	0.000007	0.000002		
Total	19	0.000324				
S = 0.002914	R-Sq = 79.1%					

Table5. Optimum value of input variables

Parameters	Optimum value
Primary amplitude(A0)	200
Parameter of Rayleigh distribution(σ)	41
Decrement coefficient(Q)	300
Number of iterations per amplitude(L)	100

5. Runs and comparison

This section compares VDO and SA algorithms. All conditions and steps for design and implementation for both algorithms were same. RSM technique was also applied to obtain the parameters affecting the quality of SA algorithm answers. Both algorithms used the same standard instances. In the literatures we could not find any standard example for RCMPSP. Hence, this paper combines a certain number of single projects in PSPLIB (Kolisch & Sprecher, 1996) to generate the multi-project examples. Also, we construct three multi-project scheduling examples corresponding to the three types: J30 set, J90 set and J120 set in PSPLIB. Each multi-project example contains 5 projects that are randomly chosen from different single-project in each set type (Table6.).

Table6. Size standard issues produced

Size of problems	No. of projects	No. of activities
Small	2	32
	5	32
	10	32
Medium	5	92
	10	92
	20	92
Large	5	122
	10	122
	20	122

Activities are subject to finish-start precedence constraints with zero minimum time lags. Each activity has a single execution mode with fixed integer duration. Activities are only scheduled when all required resource types are available. The maximum capacity for the multi project is the capacity of a single project with highest resources. All problems were executed 10 times by both algorithms. The algorithms are implemented in C++ language on PC Pentium dual core 2 GHz and 4 GB RAM as operation system. Average objective function values and the best obtained response are shown in Table 7. The time stop condition is same for each test problem and different test problem collections have their own time stop condition.

For further accreditation of this result the appropriate statistical test is performed. Non-parametric test called ‘Wilcoxen test’ is selected to test the equality of average response obtained from both algorithms. Relationship (8) indicates zero & opposite hypotheses.

$$\begin{cases} H_0: \mu_{SA} = \mu_{VDO} \\ H_1: \mu_{SA} \neq \mu_{VDO} \end{cases} \quad (8)$$

Tests using the Minitab 13 software have been done. The results are shown in Table 8 and Table 9. It shows there are no reasons for rejecting zero hypotheses. We can conclude there is no significant difference between the performances of VDO algorithm and SA algorithm and the performances of VDO algorithm is similar to the performances of SA algorithm. Although, average results from both algorithm show that the VDO algorithm is superior to SA algorithm.

Table7.Example results using algorithms

Size of problems	Average response of objective function		Average of best responses	
	SA	VDO	SA	VDO
	68.9	68.35	68.93	67.6
Small	159.08	159.09	156.9	156.5
	294.66	294.87	291.4	291.4
	223.56	223.47	220.8	220.8
Medium	398.47	398.1	395	394.6
	758.04	757.78	753.7	752.5
	306.08	306.3	302.7	302.4
Large	544.42	544.65	540.9	541.7
	1154.51	1153.9	1150.08	1150.1

Table8. Result of 'wilcoxen' test on average responses

SA N = 9 Median = 306.1
VDO N = 9 Median = 306.3
Point estimation for ETA1-ETA2 is 0.1
95.8 Percent CI for ETA1-ETA2 is (-321.1;321.0)
W = 86.0
Test of ETA1 = ETA2 vs ETA1 not = ETA2 is significant at 1.
Cannot reject at alpha = 0.05

Table9. Result of 'wilcoxen' test on best responses

best SA N = 9 Median = 302.7
best VDO N = 9 Median = 302.4
Point estimation for ETA1-ETA2 is 0.3
95.8 Percent CI for ETA1-ETA2 is (-320.8;320.0)
W = 87.0
Test of ETA1 = ETA2 vs ETA1 not = ETA2 is significant at 0.9296
The test is significant at 0.9296 (adjusted for ties)
Cannot reject at alpha = 0.05

6. Conclusion

In this paper, a vibration damping optimization (VDO) algorithm was expanded for resource constrained multi-project scheduling problems (RCMPSP). The response surface methodology (RSM) was used to set the effective parameters. The performance of the proposed algorithm was evaluated by comparison with simulated annealing (SA) algorithm. Test problems with different dimension were run on both algorithm and the result were compared. Statistical test shows that there was no significant difference between these two algorithms when applied to resource constrained multi-project scheduling problem and the performances of VDO algorithm is similar to the performances of SA algorithm. Although, average results from both algorithm show that the VDO algorithm is superior. For future research project costs and income consideration can be added to the model. Other types of relationship between project activities would be possible. All activities would be carried out in multi-mode. Preemption of activities would be allowed.

REFERENCES

[1] **Beşikci, U., Bilge, U. and Ulusoy, G. (2015),** ***Multi-Mode Resource Constrained Multi-Project Scheduling and Resource Portfolio Problem*;** European Journal of Operational Research, 240(1), 22–31;

[2] **Blazewicz, J., Lenstra, J.K., Kan, A.& Rinnooy, H.G. (1983),** ***Scheduling Subject to Resource Constraints: Classification and Complexity*.** Discrete Applied Mathematics, 5, 11-24;

[3] **Browning, R. T. and Yassine, A. A. (2010),** ***Resource-Constrained Multi-Project Scheduling: Priority Rule Performance Revisited*;** International Journal of Production Economics, 126(2), 212–228;

[4] **Chen,P.H., Shahandashti, S.M. (2009),** ***Hybrid of Genetic Algorithm and Simulated Annealing for Multiple Project Scheduling with Multiple Resource Constraints*.** Automation in Construction, 18, 434–443;

[5] **Christofides, N., Alvarez-Valdés, R., Tamarit, J. M. (1987),** ***Project Scheduling with Resource Constraints: A Branch and Bound Approach*;** European Journal of Operational Research, 29, 262–273;

[6] **Demeulemeester, E., Herroelen, W. (1992),** ***A Branch-and-Bound Procedure for the Resource Constrained Project Scheduling Problem*.** Management Science, 38(12), 1803-1818;

[7] **Goncalves, J. F., Mendes, J. J. M. and Resend, M. G. C. (2008),** ***A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem*;** European Journal of Operational Research, 189(3), 1171–1190;

[8] **Hajipour, V. Mehdizadeh E. and Tavakkoli-Moghaddamand R. (2014),** ***A Novel Pareto-Based Multi-Objective Vibration Damping Optimization Algorithm to Solve Multi-Objective Optimization Problems*;** *Sharif University of Technology, Scientia Iranica, Transactions E: Industrial Engineering*, Vol. 21, No. 6, pp. 2368-2378;

[9] **Kolisch, R., Hartmann, S. (1998),** ***Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis*,** in: Project Scheduling: Recent Models, Algorithms and Applications, ed. J. W¸ eglarz (Kluwer Academic, Amsterdam, 1998), 147–178;

[10] **Kolisch, R., Sprecher, A. (1996), PSPLIB -** ***A Project Scheduling Library;*** European Journal of Operational Research, 96, 205-216;

[11] **Kolisch, R., Sprecher, A. (1998),** ***Benchmark Instances for Scheduling Problems.*** In J. Weglarz, ed. Handbook on recent advances in project scheduling, Kluwer, Amsterdam, 197-212;

[12] **Kumanan S., Jose, G. J., Raja, K. (2006),** ***Multi-Project Scheduling Using an Heuristic and a Genetic Algorithm,*** Journal of Advanced Manufacturing Technology 31 ,360–366;

[13] **Kurtulus, I. S., and Davis, E. W. (1982),** ***Multi-Project Scheduling: Categorization of Heuristic Rules Performance.*** Management Science 28 (2), 161-172;

[14] **Lova, A., Maroto, C., Tormos, P. (2000),** ***A Multi Criteria Heuristic Method to Improve Resource Allocation in Multi Project Scheduling*.** European Journal of Operational Research, 127, 408-424;

[15] **Lova, A., Tormos, P. (2001),** ***Analysis of Scheduling Schemes and Heuristic Rules Performance in Resource-Constrained Multi Project Scheduling.*** Annals of Operations Research, 102, 263–286;

[16] Mehdizadeh, E. and Nejhad Dadgar, S. (2014), ***Using Vibration Damping Optimization Algorithm for Resource Constraint Project Scheduling Problem with Weighted Earliness-Tardiness Penalties and Interval Due Dates*;** Economic Computation and Economic Cybernetics Studies and Research (ECECSR), 48(1), 331-343;

[17] **Mehdizadeh, E. and Tavakkoli-Moghaddam, R. (2009),** ***Vibration Damping Optimization Algorithm for an Identical Parallel Machine Scheduling Problem',*** in Proceeding of the 2nd International Conference of Iranian Operations Research Society, 20–22 May, Babolsar, Iran;

[18] **Mehdizadeh, E., Tavakkoli-Moghaddam, R. and Yazdani, M. (2015),** ***A Vibration Damping Optimization Algorithm for a Parallel Machines Scheduling Problem with Sequence-Independent Family Setup Times*.** Applied Mathematical Modelling, 39(22), 6845-6859;

[19] **Mize, H. H. (1964),** ***A Heuristic for Scheduling Model for Multi-project Organizations;*** Unpublished Ph.D. Thesis, University of North California, 1964;

[20] **Myers, R. H., Montgomery, D. C. (1995),** ***Response Surface Methodology: Process and Product Optimization Using Designed Experiments.*** *John Wiley & Sons Inc.;*

[21] **Pritsker, A., Allan, B., Watters, L. J., & Wolfe, P.M. (1969),** ***Multi Project Scheduling with Limited Resources: A Zero-One Programming Approach.*** Management Science, 16(1), 93-108;

[22] **Singh, A. (2014),** ***Resource Constrained Multi-project Scheduling with Priority Rules & Analytic Hierarchy Process*;** Procedia Engineering, 69, 725–734;

[23] **Vercellis, C. (1994),** ***Constrained Multi-Project Planning Problems: A Lagrangean Decomposition Approach*;** European Journal of Operational Research, 78, 267-275.