ORIGINAL RESEARCH

Resource leveling scheduling by an ant colony-based model

Mohsen Garmsiri[*] and Mohammad Reza Abassi

Abstract

In project scheduling, many problems can arise when resource fluctuations are beyond acceptable limits. To overcome this, mathematical techniques have been developed for leveling resources. However, these produce a hard and inflexible approach in scheduling projects. The authors propose a simple resource leveling approach that can be used in scheduling projects with multi-mode execution activities. In the mentioned approach, an ant algorithm determines the execution mode of each activity so that resource leveling index and project time become optimum. In the model, some visibility functions (defined in accordance with problem characteristics) are utilized, and the best, which return the best result, is selected for the model.

Keywords: Ant colony optimization, Resource leveling, Visibility function

Background

Resource availability constraints and resource leveling are typical real-life scheduling problems. It is, thus, imperative that the development of project schedules includes not only activity precedence relations and duration assignments but also resource assignments for each activity in the network. This issue has been studied extensively. For example, Easa (1989) developed an improved tabu search model to solve resource leveling problems. Kastor and Sirakoulis (2009) evaluated the effectiveness of resource leveling tools of three popular packages by comparing the results when leveling two real construction projects as case studies. Leu et al. (2000) proposed an algorithm that can effectively provide the optimal or near-optimal combination of multiple construction resources as well as starting and finishing dates of activities subjected to the objective of resource leveling. Jiang and Shi (2005) presented a new algorithm called the enumerative branch-and-cut procedure for minimizing the total project duration of a construction project under multiple resource constraints based on an enumeration tree. Kim and Graza (2005) evaluated the resource-constrained critical path method, which some writers had proposed. Najafia et al. (2009) proposed a genetic algorithm method to solve a resource investment problem with discounted cash flows (a

project-scheduling problem in which (a) the availability levels of the resources are considered decision variables and (b) the goal is to find a schedule such that the net present value of the project cash flows optimizes).

Moumene and Ferland (2009) represented list representation for a generalization of the resource-constrained project scheduling problem. Chtourou and Haouari (2008) presented a two-stage algorithm for a robust resource-constrained project scheduling. Christodoulou et al. (2009) proposed a method for resource allocation and scheduling of resource-constrained projects by use of a measure of the projects' entropy. Ranjbar and Kianfar (2007) used a genetic algorithm-based model to schedule each activity in one of its modes in order to minimize the project makespan.

There are also some researches about resource scheduling under multi-project scheduling. For example, Kim et al. (2005) proposed a hybrid genetic algorithm with fuzzy logic controller to solve the resource-constrained multiple project scheduling problem.

There are also researches about project scheduling under situations where availability levels of the resources are variable. For example, Buddhakulsomsiri and Kim (2006) studied about the multi-mode resource-constrained project scheduling problem when activities can be split during scheduling under situations where resources may be temporarily not available.

* Correspondence: mohsen.garmsiri@yahoo.com
MBA Program, Payame Noor University, Tehran, Iran

There are also researches about project scheduling for various stages. For example, Shi and Deng (2000) developed a construction scheduling and planning method, named object-oriented resource-based planning method, for meeting the different requirements at various planning stages.

For many real-life projects, each activity can be performed using alternative modes that define the subset of the resources needed, and an activity may share different resources simultaneously. The problem is to select a single mode for each operation and, accordingly, to schedule the resources while optimizing a target. This issue has been studied extensively. For example, (Peteghem and Vanhoucke 2009) presented a genetic algorithm for the multi-mode resource-constrained project scheduling problem in which multiple execution modes are available for each of the activities of the project. Wuliang and Chengen (2009) used genetic algorithm to solve a multi-mode resource-constrained discrete time/cost trade-off problem. Chen and Weng (2009) proposed a two-phase genetic algorithm (GA) model in which both the effects of time–cost trade-off and resource scheduling are taken into account. Lova et al. (2009) proposed a hybrid genetic algorithm for finding the start times and execution modes for the activities of a project that minimize project makespan while verifying a set of precedence and resource constraints. (Damaka et al. 2009) proposed a differential evolution algorithm for solving multi-mode resource-constrained project scheduling problems. He et al. (2009) developed a model to assign activities' modes and progress payments so as to maximize the net present value of the contractor under the constraint of project deadline. Ranjbar and Kianfar (2007) used a genetic algorithm-based model to schedule each activity in one of its modes in order to minimize the project makespan.

This paper proposes a new heuristic approach for scheduling each activity of a project in one of its defined modes subject to precedence constraints in order to minimize the project makespan. Since employing time–cost trade-off analysis usually causes great fluctuations in the daily resource requirements by mixing compressed-duration activities of high resource demand with others of low resource demand, therefore, the need for resource management techniques becomes inevitable to ensure efficient utilization of resources. So in addition to project time, resource leveling index is considered as a part of scheduling objectives. Ant colony optimization algorithms are applied because they have been applied to many combinatorial optimization problems ranging from quadratic assignment to protein folding or routing vehicles, and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to produce near-optimal solutions to the traveling salesman problem. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time.

The rest of this paper starts with a definition of the project scheduling problem being solved. The focus then moves to an introduction of ant colony optimization algorithm 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.

Results and discussion

Resource leveling scheduling by an ant colony-based model

Objective function

The objective function includes two objectives:

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 leveling indexes (Elazouni and Metwally 2007; Leu et al. 2000). In this research, objective 2 implemented the minimum moment algorithm to make the daily resource consumption (r_{tj}) as smooth as possible. Also, objective 2 uses 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 represents 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

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

(1)

where r_{tj} is the consumption of type j at time t in the project, W_j is the weight of the resource j, n is the number of optimal project duration on the condition of the

resource constraints, and m is the number of total type resource project days (Pan et al. 2008).

Total objective

A compound of project cost index and resource leveling index makes the total objective:

$$\text{Total objective} = \text{IL}_T \times \frac{T}{T_{basis}} + \text{IL}_\sigma \times \frac{\sigma}{\sigma_{basis}} \quad (2)$$

where IL_T is the importance level of the project time index, IL_σ is the importance level of the resource leveling index, T is the project time, σ is the resource leveling index, T_{basis} is the basis amount of the project time index, and σ_{basis} is the basis amount of the resource leveling index.

Ant colony optimization system

Ant colony optimization (ACO) is a metaheuristic technique for hard discrete optimization problems that was first proposed in the early 1990s. In ACO, blind insects like ants can manage to establish the shortest route paths from their colony to feeding sources and back. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and quality of the food and carries some of the found food to the nest. During the return trip, the ant deposits a chemical pheromone trail on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other ants to the food source (Blum and Dorigo 2004; Gambardella et al. 1997). While an isolated ant moves essentially at random, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The collective behavior that emerges is a form of autocatalytic behavior where the more ants follow a trail, the more attractive that trail becomes for being followed. The process is characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. Also, pheromone concentration vanishes with time, and then, the less-used path will be much lower in pheromone concentration. The indirect communication between the ants via the pheromone trails allows them to find the shortest paths between their nest and food sources (Bonabeau et al. 1999).

ACO will have some major differences with a real (natural) one:

(1.) Artificial ants will have some memory.
(2.) They will not be completely blind.
(3.) They will live in an environment where time is discrete.

The idea is that if, at a given point, an ant has to choose among different paths, it does this based on two main components:

(1.) Pheromone trails that were heavily chosen by preceding ants (that is, those with a high trail level) are chosen with higher probability. Furthermore, high trail levels (pheromone values) are synonymous with short paths.
(2.) Heuristic, which is visibility, is defined as the inverse of the distance, and so, closer nodes have a higher chance of being next.

Two parameters (α and β) are introduced to control the relative weight of pheromone trail intensity and visibility. At run-time, ACO algorithms try to update the pheromone values in such a way that the probability to generate high-quality solutions increases over time. The pheromone values are updated using previously generated solutions. The reinforcement of solution components depending on the solution quality is an important ingredient of ACO algorithms. It implicitly assumes that good solutions consist of good solution components. To learn which components contribute to good solutions can help to assemble them into better solutions (Dorigo et al. 1996).

The basic ACO algorithm involves the following: ants choose the trail with the largest concentration of pheromone τ with a certain probability p. The shorter path will have more pheromone concentration. For a whole colony following this behavior, the shortest path will be much higher in pheromone concentration than the other paths. Then, the probability of choosing any other paths will be very small. Also, the less-used path will be much lower in pheromone concentration due to pheromone volatility with time. Let an ant k reside in a node i at time t. Assume that one of the possible trails for its next step is (i, j) connecting node i to node j. Then, the probability of choosing node j as the next node is given by

$$P_{(k,i,j)} = \begin{cases} \dfrac{\tau(i,j)^\alpha \eta(i,j)^\beta}{\sum\limits_{r \in \Gamma} \tau(i,j)^\alpha \eta(i,j)^\beta} & r \in \Gamma \\ 0 & r \notin \Gamma \end{cases} \quad (3)$$

where the pheromone trail update is given by $\tau(i,j)(t+1) = (1-v)^* \tau(i,j)(t) + v^* \delta(k,i,j)$.

$v \boxtimes (0, 1)$ is the evaporation coefficient. $\delta(k, i, j)$ is the pheromone released by the ant k on the trail (i, j).

$$\delta(k,i,j) = \begin{cases} Q/L & \text{if ant selected trail} = j \\ 0 & \text{if ant selected trail} \neq j \end{cases} \quad (4)$$

where Q is a positive constant, and L is the total path time. Γ is a tabu list, which contains all the trails that

Figure 1 Execution mode selection and route establishment by ants.
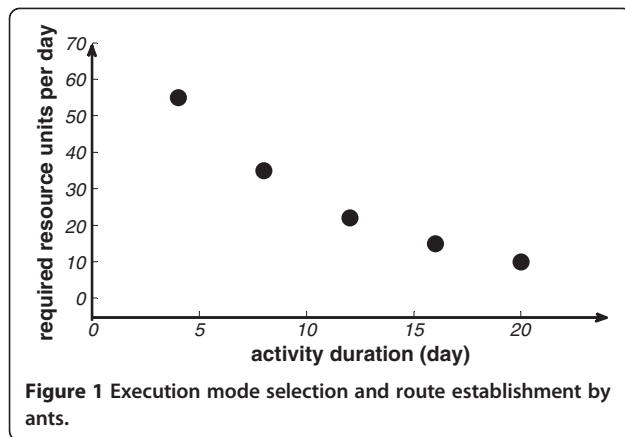
the ant has already passed and must not be chosen again; this acts as the memory of the ant. $\eta(i, j)$ is the visibility function, which is usually selected as the inverse of the distance for i to j, i.e.,

$$\eta(i,j) = \gamma/d(i,j).$$

Where γ is the visibility parameter, α and β express the relative weight between the importance of τ and η (Abdallah 2009).

Proposed ACO technique for solving resource leveling scheduling

The ant selects an execution mode (from available execution modes) for each activity and establishes a route from the initial activity until the last. A schedule is established based on the route and as-soon-as-possible scheduling approach. A route that has optimal objective function with respect to all other routes is the best route. Figure 1 demonstrates the execution mode selection and route establishment by ants.

In the algorithm proposed here, each feasible solution is demonstrated by the mode string. The mode string *ms* can be regarded as a vector. There are as many parts in the vector as there are activities. The situation that an activity a_j performed in mode m can be represented by $Ms(j) = m$, where $1 \leq j \leq N$ and $1 \leq m \leq M_j$, respectively.

Figure 2 demonstrates a feasible solution determined by ant 1. In this figure, the third number of the mode string is 1, indicating that the execution mode of activity 3 is 1.

In the proposed algorithm, the visibility function $\eta(i, j)$, which is usually selected as the inverse of the distance from i to j, is proposed here to be proportional to the (1) inverse of the cost, (2) inverse of the cost multiplied by the duration, (3) inverse of the duration, (4) duration, and (5) inverse of the number of required resource per day. So, some models for different visibility functions are established.

For the first run, each ant is given a random guide to the execution modes for each activity. In other words, the ants choose one of the available execution modes for each activity. They put their pheromone trace on their path. A tabu list, which contains all the execution modes that the ant has already selected, is made. The tabu list acts as the memory of the ant. The pheromone level is also updated. For other runs, ants choose one of the available execution modes for each activity guided by the pheromone trail and visibility level, putting its pheromone trace on this path. The pheromone level is also updated. The mentioned procedure is repeated, until certain stopping criterion is met (e.g., a fixed number of iterations).

Computational study

A simple project is used to demonstrate the proposed model. The data of each activity and the relationship between activities are shown in Table 1. Resource cost rates are 20 and 15 units per day.

We use procedure with ACO-related input data as follows: number of ants = 50 and iteration number = 1,000. The proposed algorithms were implemented in MATLAB language. The running time using Intel (R) Pentium D, 3 GHz, 160 MB RAM is approximately 60 s.

In the previous researches, the effects of some parameters, including v, α, and β, have been analyzed. In this research, in addition to afore-stated parameters, the impact of different definitions of visibility function is analyzed. The normalized different definitions of visibility

Activity **1**　　Activity **2**　　Activity **3**　　　　Activity **n**

Figure 2 A feasible solution.
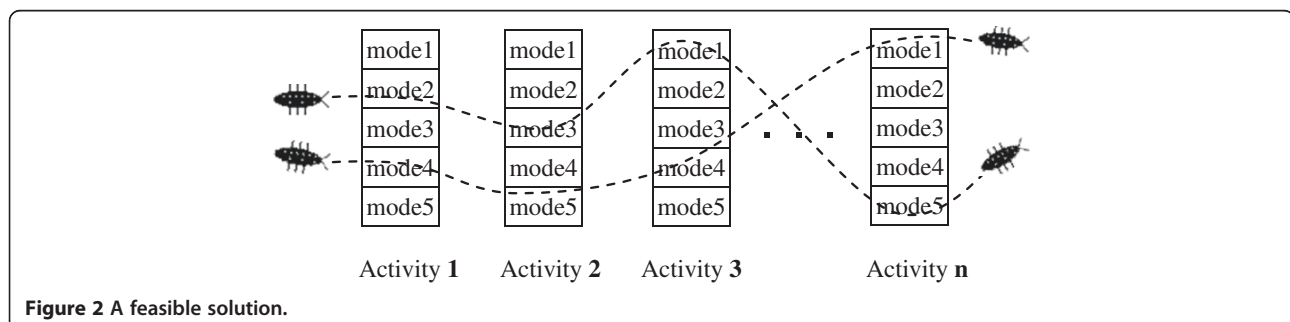
Table 1 The activity data

Activity	Precedence	Option	Duration	Resource 1 requirement	Resource 2 requirement
1	-	1	25	5	2
		2	15	7	4
		3	10	10	8
2	1	1	35	10	3
		2	22	15	7
		3	18	25	15
3	1	1	22	8	5
		2	15	15	9
		3	12	25	20
4	1	1	25	10	7
		2	18	18	15
		3	14	30	25
5	2	1	13	10	15
		2	10	15	28
		3	8	25	40
6	3, 4	1	20	18	20
		2	18	30	30
		3	14	40	45
7	2, 4	1	18	10	14
		2	12	20	18
		3	8	25	35
8	6	1	15	5	8
		2	12	10	15
		3	8	15	25
9	7	1	30	10	15
		2	18	1	20
		3	15	25	30
10	8	1	25	10	5
		2	20	15	10
		3	15	25	20
11	8	1	24	5	10
		2	20	10	15
		3	18	15	25
12	8	1	20	10	10
		2	16	15	20
		3	14	20	25

function for different execution modes of activities are presented in Tables 2, 3, 4, 5 and 6.

The values of w_T and w_σ are fixed to 0.5. The values of T_{basis} and σ_{basis} are equal to the project time and resource leveling index of the project schedule in which the first execution mode is selected for each activity.

With each parameter and visibility function combination, 100 runs of experiments on the project example

are performed so that the stochastic characteristic in ACO can be reflected. Results from the runs are shown in Table 7. The results of model running based on different values of ν, α and β and different definitions of visibility function are presented in Table 7.

In each cell of Table 7, the first number is the total objective function. The second number is the iteration number in which the ants select a route as the best, in other words, model result does not change after the mentioned iteration number. The third number is the project time of the solution, and the fourth number is the resource leveling index of the solution. The best solutions that have minimum total objective function are in bold.

Comparing different combinations that have different parameters values and definitions for visibility function, it is found that a big α and small β lead to better solutions because using extremely big α and small β encourages the ants to follow the preceding ants' pheromones. Using small α and big β encourages the ants to choose the next node as having the largest trail time. This is clear from Equation 3. In this case, the ants are not heavily attracted to the pheromones of preceding ants. This may lead to inaccurate identification of the best solution.

More than that, using high evaporation coefficient leads the ants to make all paths randomly, and the ant does not influence the succeeding ones, and evaporation coefficient leads to fast convergence to the preceding path and may lead to immature convergence due to the lack of exploration of other paths. However, results of the project show that changing the evaporation coefficients causes little difference in the ACO performance. It can be concluded that the initial adoption of the ants are good solutions.

As it can be seen in Table 7, the first definition for visibility function gets good result with respect to all other definitions. Project time and resource leveling index optimization are the objectives. In addition, the first definition is an inverse of cost, and cost is a function of activity time and number of required resource per day. Thus, the first definition produces the best results.

Another consideration that is dependent on the model results is the rapidity of getting result. As it can be seen in Table 7, the combinations which get good result get the result more rapidly. As a conclusion of the mentioned fact, the amounts of ν, α, β and the definition for visibility function affect the quality and rapidity of getting result.

Conclusions

An ACO-based method including its corresponding framework is proposed for solving the project scheduling

Table 2 Normalized first definitions of visibility function for different execution modes of activities

(1)	Mode 1	Mode 2	Mode 3
A1	0.32	0.35	0.33
A2	0.38	0.37	0.25
A3	0.43	0.34	0.23
A4	0.44	0.32	0.24
A5	0.42	0.32	0.26
A6	0.52	0.25	0.23
A7	0.36	0.33	0.32
A8	0.44	0.29	0.27
A9	0.42	0.33	0.25
A10	0.43	0.33	0.25
A11	0.45	0.32	0.22
A12	0.42	0.31	0.27

Table 4 Normalized third definitions of visibility function for different execution modes of activities

(3)	Mode 1	Mode 2	Mode 3
A1	0.19	0.33	0.48
A2	0.22	0.35	0.43
A3	0.23	0.34	0.43
A4	0.24	0.33	0.43
A5	0.25	0.33	0.43
A6	0.28	0.31	0.4
A7	0.21	0.32	0.47
A8	0.24	0.3	0.45
A9	0.29	0.32	0.39
A10	0.26	0.32	0.43
A11	0.28	0.34	0.38
A12	0.27	0.34	0.39

problem in terms of minimizing project duration and resource leveling index. An ant representation of activity execution mode, based on which a feasible schedule (including duration and resource allocation for all activities and, in result, project duration and resource leveling index) can be definitely produced, is adopted for the ACO-based approach. The ACO artificial agent seems to provide a powerful alternative to traditional methods in terms of construction scheduling since it is able to provide an efficient and easy-to-implement alternative to analyze and achieve the project scheduling problem and, in result, efficiently construct the best solutions in acyclic (unidirectional) network topologies.

The proposed model is applied to a project comprising 15 activities. For the sample project, the proposed model has shown that despite the seemingly iterative approach of the ACO algorithm, the method exhibits quick convergence to the final solution and, thus, low

computational time. Furthermore, the model also assists project managers in making decision about resources.

Ongoing work on the applicability of the ACO artificial agent addresses the inclusion of resource-based scheduling techniques to account for resource leveling in multi-project scheduling. More than that, the proposed ACO method will be developed into a user-friendly system so that the practitioners can utilize it in solving the project scheduling problems.

Methods
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
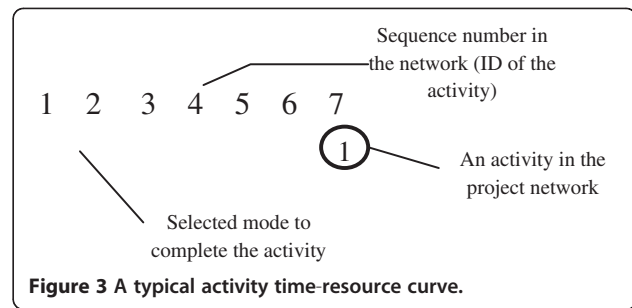
Table 3 Normalized second definitions of visibility function for different execution modes of activities

(2)	Mode 1	Mode 2	Mode 3
A1	0.19	0.34	0.48
A2	0.26	0.40	0.34
A3	0.32	0.37	0.31
A4	0.33	0.34	0.33
A5	0.33	0.33	0.34
A6	0.46	0.25	0.29
A7	0.23	0.31	0.46
A8	0.34	0.27	0.39
A9	0.37	0.33	0.3
A10	0.34	0.33	0.33
A11	0.4	0.34	0.26
A12	0.35	0.32	0.33

Table 5 Normalized fourth definitions of visibility function for different execution modes of activities

(4)	Mode 1	Mode 2	Mode 3
A1	0.5	0.3	0.2
A2	0.47	0.29	0.24
A3	0.45	0.31	0.25
A4	0.44	0.32	0.25
A5	0.42	0.32	0.26
A6	0.39	0.35	0.27
A7	0.47	0.32	0.21
A8	0.43	0.34	0.23
A9	0.38	0.34	0.28
A10	0.42	0.33	0.25
A11	0.39	0.32	0.29
A12	0.4	0.32	0.28

Table 6 Normalized fifth definitions of visibility function for different execution modes of activities

(5)	Mode 1	Mode 2	Mode 3
A1	0.49	0.32	0.20
A2	0.52	0.31	0.17
A3	0.55	0.30	0.16
A4	0.55	0.28	0.17
A5	0.51	0.30	0.20
A6	0.57	0.25	0.18
A7	0.50	0.30	0.20
A8	0.54	0.28	18
A9	0.46	0.33	0.21
A10	0.51	0.31	0.18
A11	0.51	0.3	0.19
A12	0.49	0.29	0.22

Figure 3 A typical activity time-resource curve.

resource planning will facilitate a reasonably stable level of the work force. Other advantages of resource leveling include simplified resource tracking and control, lower cost of resource management and improved opportunity for learning. For leveling resources, there are some tools such as Primavera and MS Project. Each tool uses a special approach to level resources.

These scheduling tools largely rely on heuristics in leveling resource in forward-pass scheduling; thus, in general, they do not yield the optimum solution in terms of the shortest project duration. In addition, the proposed resource leveling methods, though, do not necessarily optimize the allocation of resources so as to minimize the project duration. For example, a resource leveling 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 a more appropriate method, which considers project time, for leveling resources. In other words, both of the objectives, project time minimization and resource leveling, should be considered. To this purpose, a resource leveling index

Table 7 Result of the model

	$v = 0.5, \alpha = 1, \beta = 0.1$	$v = 0.5, \alpha = 0.1, \beta = 1$	$N = 0.5, \alpha = 0.5, \beta = 0.5$	$v = 0.9, \alpha = 1, \beta = 0.1$	$v = 0.9, \alpha = 0.1, \beta = 1$	$v = 0.9, \alpha = 0.5, \beta = 0.5$	$v = 0.1, \alpha = 1, \beta = 0.1$	$v = 0.1, \alpha = 0.1, \beta = 1$	$v = 0.1, \alpha = 0.5, \beta = 0.5$
1	1.3	1.56	1.53	1.28	1.51	1.47	1.37	1.53	1.42
	97	200	200	45	200	200	200	200	200
	66	69	50	59	63	66	70	56	64
	2,157,355	2,923,620	3,488,240	2,333,190	2,963,125	2,724,050	2,257,260	3,260,540	2,625,010
2	1.46	1.57	1.49	2	1.6	1.4	1.41	1.46	1.49
	194	200	200	1	200	200	200	200	200
	66	78	68	99	63	56	59	62	66
	2,694,100	2,636,525	2,735,690	3,391,750	3,270,925	2,824,710	2,757,290	2,814,450	2,779,960
3	1.39	1.63	1.52	2	1.57	1.65	1.36	1.49	1.64
	154	54	200	1	200	200	200	200	200
	59	**66**	**75**	**99**	**78**	**61**	**57**	**54**	**64**
	2,691,540	3,264,390	2,601,240	3,391,750	2,636,525	3,505,895	2,653,155	3,085,215	3,353,520
4	1.43	1.44	1.55	1.38	1.35	1.52	1.34	1.38	1.45
	37	200	200	96	200	200	200	200	200
	75	75	60	70	67	60	61	72	64
	2,266,535	2,303,975	3,215,840	2,274,415	2,289,725	3,109,700	2,452,995	2,208,825	2,739,430
5	1.37	1.46	1.46	1.3	1.38	1.48	1.34	1.46	1.46
	94	199	200	109	200	200	200	200	200
	70	70	61	70	61	67	75	61	60
	2,257,260	2,540,170	2,868,195	1,994,880	2,598,640	2,726,395	1,987,000	2,868,195	2,888,185

The best solutions that have minimum total objective function are in bold.

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.

For many real-life projects, it is possible to execute each activity in one of several alternative modes, which usually represent relations between the consumed resources and the duration of the activity, leading to some problems. The mentioned problems involve the selection of a set of execution modes in order to achieve a certain objective. The study involves the selection of a set of execution modes in order to level resources. In other words, each activity can be performed using alternative modes that define the subset of the resources needed. A typical activity discrete time-resource curve is shown in Figure 3.

In project scheduling, at first, the execution mode of each activity is selected so that objectives, project time minimization and resource leveling index optimization, are considered. Then, noncritical activities are shifted within their available slacks so that the resource usage curve became smoother. So, the execution mode of each activity determination is under the research problem.

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 be specified and may depend on several considerations. This requirement is such that any project completion later than this has 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 requirement optimization should arrive at the set of shifting amount of activity that results in a valid project completion date and have the minimum fluctuations in the daily resource requirements.

The project scheduling and resource leveling problem under study is based on the following assumptions: (1) some activities compose a project; (2) each activity can be performed using alternative modes that define the subset of the resources needed and its duration; (3) all predecessors must be finished before an activity can start (i.e., precedence constraints); (4) resources are renewable from period to period; and (5) activities are non-preemptive, that is, cannot be interrupted when in progress.

References

Abdallah H, Emara HM, Dorrah HT, Bahgat A (2009) Using ant colony optimization algorithm for solving project management problems. Expert Syst Appl 36:10004–10015

Blum C, Dorigo M (2004) The hyper-cube framework for ant colony optimization. Transactions on Systems, Man and Cybernetics – Part B. Cybernetics 34 (2):1161–1172

Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence from natural to artificial systems. Oxford University Press, Oxford

Buddhakulsomsiri J, Kim DS (2006) Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. Eur J Oper Res 175:279–295

Chen P-H, Weng H (2009) A two-phase GA model for resource-constrained project scheduling. Autom Constr 18:485–498

Chtourou H, Haouari M (2008) A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. Comput Ind Eng 55:183–194

Christodoulou S, Ellinas G, Pooyan A (2009) Entropy-based scheduling of resource-constrained construction projects. Autom Constr 18:919–928

Damaka N, Jarbouia B, Siarryb P, Loukil T (2009) Differential evolution for solving multi-mode resource-constrained project scheduling problems. Comput Oper Res 36:2653–2659

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B :29–41

Easa SM (1989) Resource leveling in construction by optimization. J Constr Eng Manag 115:302–316

Elazouni AM, Metwally FG (2007) Expanding finance-based scheduling to devise overall-optimized project schedules. Journal of Construction Engineering and Management 133:86–90

Gambardella LM, Taillard E, Dorigo M (1997) Ant Colonies for QAP. Technical Report Number: IDSIA 97–4, IDSIA, Lugano, Switzerland

He Z, Wang N, Jia T, Xu Y (2009) Simulated annealing and tabu search for multi-mode project payment scheduling. Eur J Oper Res 198:688–696

Jiang G, Shi J (2005) Exact algorithm for solving project scheduling problems under multiple resource constraints. Journal of Construction Engineering and Management 131:986–992

Kastor A, Sirakoulis K (2009) The effectiveness of resource levelling tools for resource constraint project scheduling problem. Int J Proj Manag 27:493–500

Kim K, de la Garza JM (2005) Evaluation of the resource-constrained critical path method algorithms. Journal of Construction Engineering and Management 131:522–532

Kim KW, Yun YS, Yoon JM, Gen M, Yamazaki G (2005) Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. Comput Ind 56:143–160

Leu S-S, Yang C-H, Huang J-C (2000) Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. Automation in Construction 10:27–41

Lova A, Tormos P, Cervantes M, Barber F (2009) An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. Int J Production Economics 117:302–316

Moumene K, Ferland JA (2009) Activity list representation for a generalization of the resource-constrained project scheduling problem. Eur J Oper Res 199:46–54

Najafia AA, Niaki STA, Moslem S (2009) A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. Comput Oper Res 36:2994–3001

Pan N-H, Hsaio P-W, Chen K-Y (2008) A study of project scheduling optimization using tabu search algorithm. Eng Appl Artif Intel 21:1101–1112

Ranjbar MR, Kianfar F (2007) Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms. Appl Math Comput 191:451–456

Shi JJ, Deng Z (2000) Object-oriented resource-based planning method (ORPM) for construction. Int J Proj Manag 18:179–188

Van Peteghem V, Vanhoucke M (2009) A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. Eur J Oper Res 201:409–418

Wuliang P, Chengen W (2009) A multi-mode resource-constrained discrete time–cost trade-off problem and its genetic algorithm based solution. Int J Proj Manag 27:600–609