

Activity Prioritization Under Resource Constraints Using a Utility Index Method

P. Aslani¹, S. Christodoulou^{*,2}, F.H. Griffis¹, G. Ellinas³ and L. Chiarelli¹

¹Department of Civil Engineering, Polytechnic University, Brooklyn, NY 11201, USA

²Department of Civil and Environmental Engineering, University of Cyprus, Nicosia 1678, Cyprus

³Department of Electrical and Computer Engineering, University of Cyprus, Nicosia 1678, Cyprus

Abstract: Resource availability constraints are a typical real-life construction scheduling problem; a problem that limits a constructor's ability to execute and deliver a project as originally planned. It is, thus, imperative that developed project schedules should have not only well-thought project logic networks (successor/predecessor information and activity durations) but also resource assignments (including cost) for each activity in the network so that the effects of resource constraints can effectively be accounted for. The paper presents a new approach to resource-constrained scheduling that allows for activity prioritization when a project is subject to limited resources. The methodology proposed is based on a utility index, hereby defined as the ratio of the number of required resources for a specific activity to the total number of required resources among competing activities. A dynamic programming technique is adopted to maximize the utility value for each activity so that the resource allocation among competing activities, as suggested by the method, results in the minimum overall project duration.

INTRODUCTION

Projects the activities of which are assigned a resource (or a set of resources) with limited capacity or of limited availability fall under a category of scheduling problems termed Resource-Constrained Scheduling Problems (RCSP). These projects are constrained in their capacity to meet the constructor's predefined objectives, usually to complete all necessary tasks in the minimum total finish time. Since the parameters defining an RCSP could vary in nature and quantity, the resulting solution optimization is an NP-hard problem the complexity of which increases substantially with the project's network size.

Unlike resource-constrained scheduling problems encountered in typical day-to-day operations though (such as employee shifts, manufacturing, bus routing, etc.) resource-constrained construction schedules are also governed by network precedence relationships. This adds to the complexity of the problem since in searching for the minimum project duration under resource constraints, activity precedence constraints should also be taken into account. In general, construction tasks require different amounts of resources of which only a certain amount is available at each time step (Lenstra 1983) [1]. A solution to RCSP can be sought through a number of possible methodologies. A very common approach is to utilize implicit enumeration and backtracking (the methods used by most commercially available exact solvers) such as branch and bound methods, as proposed in Brucker and Knust [2],

Demeulemeester and Herroelen [3], Mingozzi *et al.* [4], Crawford [5], and Garey *et al.* [6]. Other methods utilize intelligent branching and evaluation techniques such as mathematical programming [7], dynamic programming [8], zero-one programming [9], or artificial agent techniques such as genetic algorithms [10] and ant colony optimization [11-13]. Brucker has also developed a variety of heuristic approaches for RCSP which can also solve realistic problems in reasonable time [14].

This paper presents a new resource allocation methodology [15] the goal of which is to help a planner and/or constructor prioritize the work at hand, optimize the allocation of resources available to him and minimize a project duration. The method makes use of the perceived value earned by the constructor upon completion of an activity in terms of resource utilization. Within this context, an activity's utility index may be defined as the ratio of the number of required resources to accomplish the specific activity to the total number of resources required by all competing activities at the time of resource allocation. A dynamic programming technique is adopted to maximize the utility index for each activity and subsequently optimize the allocation of resources and minimize the project's duration.

MATHEMATICAL FORMULATION

There are, in general, two categories of scheduling paradigms that can be used in addressing resource-constrained problems: serial scheduling and parallel scheduling. In the former paradigm, a priority list of activities is determined at time zero, with the order of activities in the priority list independent of the resource constraints. Given the priority list, activities are scheduled at the earliest possible time at which the precedence constraints are satisfied and the resources are available [16-20]. In the latter sche-duling

*Address correspondence to this author at the Department of Civil and Environmental Engineering, University of Cyprus, Nicosia 1678, Cyprus; Tel: +357-22892270; Fax: +357-22892295; E-mails: SCHRISTO@UCY.AC.CY or SC163@HOTMAIL.COM

paradigm, the order of activities is not determined at time zero and scheduling decisions are made when activities are planned to begin and resources are available.

If we map a construction project topology and precedence relationships as an activity-on-node network of the format $G(n,A)$, then we have a set of nodes (n) corresponding to activities and a set of directed edges (A) representing the direct precedence relations. We call this set of nodes and directed edges a directed graph. Each activity $i \in n$ has a specified duration d_i and cost c_i , with the duration derived by the volume of work (Q_i) included in the activity and the production rate (N_i) of the resources assigned to the activity. The relation between the three parameters is given by $d_i = Q_i / N_i$, with the resulting duration usually presumed to follow a triangular distribution as a result of variations in the production rate of the activity's resources during the time horizon of the activity. Traditional critical path method scheduling (CPM) is deterministic in its approach to activity durations (a single value is assumed for each activity's duration), whereas the assumption for activity durations following a triangular distribution can be utilized in stochastic approaches to construction scheduling (Program Evaluation and Review Technique, PERT).

In terms of parallel resource-constrained scheduling, the fundamental issue is which activity among competing activities, will receive the required resources first. In order to answer this question the construction planner has to dynamically evaluate his options at every junction in the construction process and decide on which activity would maximize the constructor's "return on investment", which usually means the activities that would minimize the overall project duration.

Typical commercial software, such as Primavera Project Planner® (P3) or Microsoft Project® (MSP), address resource-constrained scheduling by means of resource leveling. During resource leveling, P3 for example "only schedules an activity to occur when its resource demands can be met. To accomplish this, P3 may delay or advance activities to resolve resource availability conflicts. Typically, leveling is done on the forward pass through the project (and) P3 determines the earliest dates it can schedule an activity when sufficient resources will be available to perform the task." [21] The method P3 uses to forward-level a project involves several steps [21]:

The entire network is broken into a logical sequence of activities to find the first activity, or activities, without unsatisfied predecessor relationships. In the event of a tie, P3 uses the priority specification as the tiebreaker.

Normal resource limits are initially assumed and any resources that activities in progress are using are deducted from the available resource pool.

P3 checks that there will be sufficient resources available to satisfy the activity's demands for its entire duration. If so, P3 schedules the activity and returns to step 1. If not, the leveling process continues with step 4. Since the resource requirements are not always constant and continuous during every workperiod of an activity's duration, the software may split, stretch, or crunch certain types of activities during forward leveling based on the planner's instructions.

If the activity has no positive total float, P3 goes to step 5; otherwise, it delays the activity's start by one workperiod and re-examines the resource availability during each workperiod of the activity. If there are always sufficient resources available, P3 schedules the activity and goes back to step 1. If not, it repeats step 4 until the total float is exhausted.

If P3 reaches step 5 for the first time then the positive float of the activity had been exhausted without finding sufficient resources to schedule the activity under the normal resource limits. P3 tries to schedule the activity again, using the activity's original, early scheduled date, but this time assuming a higher resource availability, depending on whether or not resource smoothing is in effect. If P3 had previously reached step 5 with the particular activity, it skips this step and goes directly to step 6.

If the leveling process reaches step 6, P3 could not schedule the activity within its total float, even assuming the maximum resource availability. As a result, P3 delays the activity beyond its total float. This means that the activity will delay the project's completion. P3 continues to assume the maximum resource availability, and delays the activity one workperiod at a time, each time checking to see if sufficient resources are available during each workperiod of the activity.

Similarly, when MSP levels overallocated resources, "it goes through a series of decisions about each of the tasks in the schedule to determine whether they can be delayed or split in order to alleviate the resource overallocations. The following factors are examined to determine which tasks should be delayed or split" [22]: available slack, task duration, task's constraints, task ID, task priority, task interdependencies, scheduling dates. As seen, the prioritization is primarily based on the activities's total float and duration values.

Let us deviate from the above prioritization heuristics and reconsider the resource-constrained scheduling problem by focusing on resources. Let us, in turn, associate two metrics to each activity: (a) a certain resource requirement ($r_{i,j}$) for the activity and (b) a utility index ($UI_{i,j}$). The former is the number of resources of type j that had been scheduled for activity i , and in essence sets the productivity rate used in $d_i = Q_i / N_i$ to calculate the duration of each activity. The latter is a calculated score, which is hereby defined as the ratio of required resources ($r_{i,j}$) for a specific activity (i) over the total estimated resources for all competing activities (m) in a specific period of time (the time to finish all parallel activities).

$$UI_{i,j} = \frac{r_{i,j}}{\sum_{k=1}^m (r_{k,j})} \quad (1)$$

The resulting resource-constrained problem is to determine how many units of the required resource should be diverted to an activity in order to maximize the total utility index within the specified time-period, while there is a limited number of an available resource.

Assuming that,

w is the total number of units of resource type j available to the project,

Activity i is implemented at a level x_i (assuming x_i is an integer),

$r_{i,j} * x_i$ units of resource type j are used by activity i , then the objective of the first optimization to be performed is the maximization of the total utility index at the subject resource usage level,

$$\max \sum_{i=1}^n (UI_{i,j} * x_i) \quad (2a)$$

$$\text{s.t. } \sum_{i=1}^n (r_{i,j} * x_i) \leq w \quad (2b)$$

$$\text{and } x_i \geq 0 \text{ for resource type } j \quad (2c)$$

Since a schedule, S , is also defined by an n -vector of starting times, s_i , and implied finish times $f_i = s_i + d_i$ for each node i (with $i = 1, 2, 3, \dots, n$), then a subsequent optimization would be to minimize the finish time of the critical path,

$$\min \sum_{i=1}^n (f_i) \quad (3)$$

Should we assume that the productivity within an activity is directly proportional to the number of resources assigned to it then one can accelerate an activity by assigning more resources to the activity? Suppose we have a total of w available units of a resource, and we define $g(w)$ to be the maximum utility index that can be gained from w . To solve Eq.3 by dynamic programming, we define $g_i(w)$ to be the maximum utility index obtained from activities $i, i+1, i+2, \dots, n$ if:

$$g_{i+1}(w) = 0, \text{ for all } UI_i \quad (4)$$

$$g_{i+1}(w) = \max \{ UI_i + g_{i+1} [w - e_{i,j} * x_i] \}, x_i \geq 0 \quad (5)$$

for resource type j

where, i , must be an integer and, w_i , must satisfy $w_i \leq w$. The reasoning behind the above equations is that in order to use all available resources optimally one must begin by using some combination of resources. This will increase the resource utilization within the time-period of interest and produce the higher "return-on-investment" with respect to decreasing the overall project duration.

Furthermore, we define $x(w)$ the strategy that attains the maximum in Eq.5 and, $x(w) = 0$, to mean that no resource combination can be used ([8, 16]).

SAMPLE CALCULATIONS

Let us consider a small case-study network (Fig. 1) consisting of four activities scheduled in parallel, and requiring the same type of resource. The resource requirements for each activity (also shown in Fig. (1)) combine for a total of 18 resources required for completing the project.

In the absence of any resource constraints the project resources can be allocated and scheduled as shown in Fig. (2). If we suppose, though, that at time zero (the project start) the constructor has only 12 resources available to him and that the project completion time is of the essence, then the constructor's dilemma is how to best allocate his resources to the set of activities immediately following the "start" node so as to maximize their impact to the project ("finish" node). Thus, resource allocation and optimization is desired so as to warrant the shortest project duration.

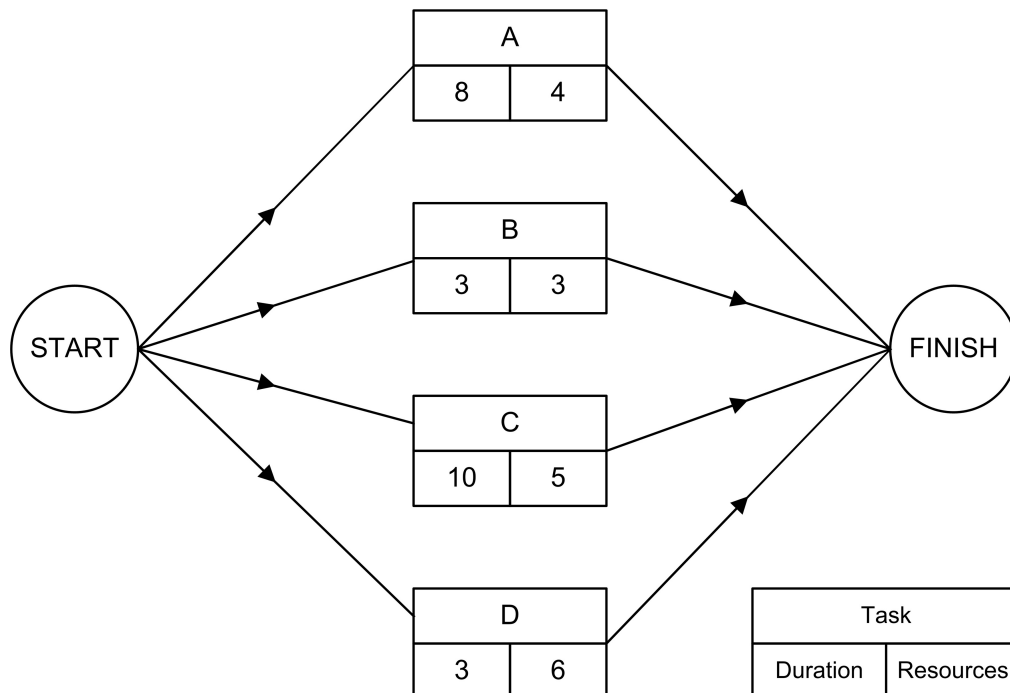


Fig. (1). Case study network.

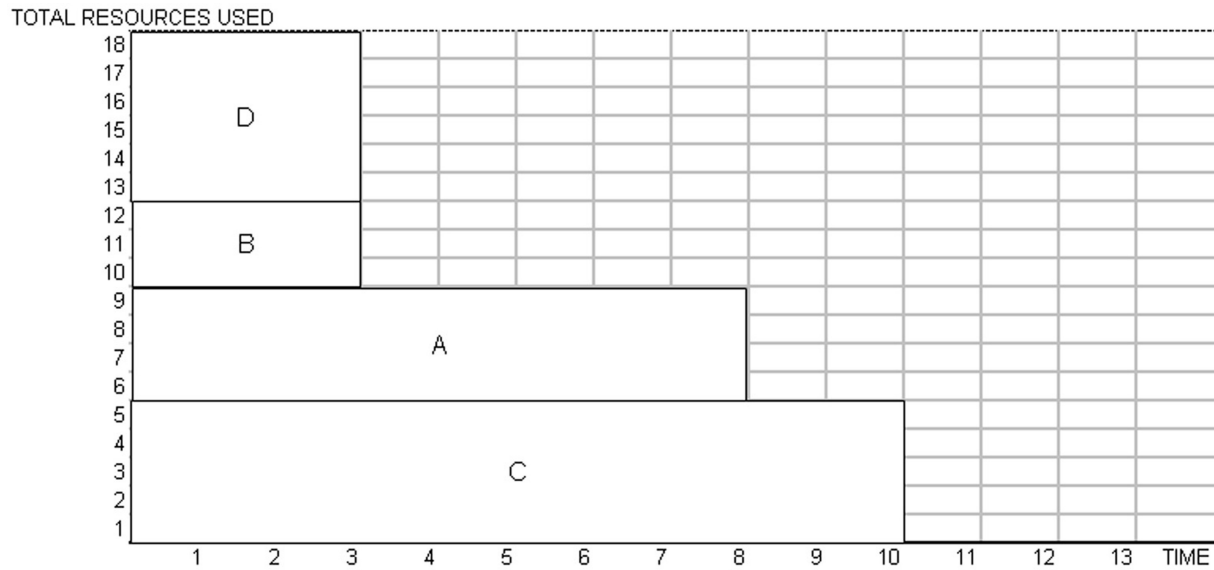


Fig. (2). Project schedule and resource allocation in the absence of resource constraints.

In terms of traditional CPM-based reasoning, the impulse would be to first divert all resources to the critical (or near-critical) activities and expedite them in the hope that the final overall project duration is shortened. In such a case, the possible strategy could be to allocate all possible resources to activity C. The result would be to cut the duration of activity C in half (by assigning 10 resources to it). Since only 2 resources would be left available for the other activities no other activity could be done during the same time period, and the project duration would increase to $10/2 + 8 = 13$ time-units, since the critical path would now run through activities C and A.

Let us now investigate the utility index strategy. The utility index for each activity, as previously defined, is obtained by dividing each activity's resource requirement by the total number of resources required to complete all activities. Thus, the utility index (UI) for activity A is $UI_A = 4/18$, for activity B is $UI_B = 3/18$ and for activities C and D are $UI_C = 5/18$ and $UI_D = 6/18$ respectively.

Table 1. Resource Requirements and Resulting Utility Indices (UI) for Case-Study Network

Activity Index (i)	Activity Title	Activity Duration	Resource Requirements	Utility Index (UI)
1	A	8	4	4/18
2	B	3	3	3/18
3	C	10	5	5/18
4	D	3	6	6/18

Since no activity can be done with only 0, 1, or 2 resources, then,

$$g(0) = g(1) = g(2) = 0, \text{ and } s(0) = s(1) = s(2) = 0.$$

For any number of resources higher than 2, at least one of the activities can be executed. For example, when the constructor has three resources at his disposal he can complete activity B, when he has four resources he can

complete activity C, when he has six resources he can either complete activity D at normal pace and duration or he can choose to complete activity B at half its duration by double-crewing the activity (2 crews of 3 resources each, for $3/2=1.5$ time-units of duration). In such cases, the constructor should decide which strategy (activity D in 3 time-units, or activity B in 1.5 time-units) is the most advantageous to the project and act accordingly. Thus, the constructor will have to use the utility index of each activity as a measure of its priority within the overall project schedule. One approach could be to consider the total float of the activity as an indication of its criticality and thus utility. Another approach could be to consider the opportunity gains (or losses) associated with the activity and act accordingly. For example, activities linked to contractual incentives or disincentives, or activities requiring expedition for facilitating other work should rank higher in a constructor's priority list.

Based on an assumption that activities requiring a higher number of resources to complete are bigger in scope and more complicated (thus more critical), we start the resource-constrained optimization algorithm from the activity that requires the higher number of resources (activity D) and we evaluate the maximum utility index that can be gained by implementing a chosen strategy. In the case of activity D, and assuming that the resources assigned to it (r_D) can only have an impact if they act as a crew (thus in groups of 6), then the utility gained (g_D) should be maximized by use of the equation

$$g_D(u) = \max_{r_D} \left\{ \frac{6}{18} r_D \right\} \quad (6)$$

where $6r_D \leq 12$ and $r_D \geq 0$.

Thus,

$$g_D(12) = 12/18$$

$$g_D(11) = g_D(10) = g_D(9) = g_D(8) = g_D(7) = 6/18$$

$$g_D(6) = 6/18$$

$$g_D(5) = g_D(4) = g_D(3) = g_D(2) = g_D(1) = 0$$

The corresponding resource-assignment strategies are:

$$s_D(12) = 2$$

$$s_D(11) = s_D(10) = s_D(9) = s_D(8) = s_D(7) = 1$$

$$s_D(6) = 1$$

$$s_D(5) = s_D(4) = s_D(3) = s_D(2) = s_D(1) = 0$$

We then turn our attention to activity C as the activity next-in-line in terms of resource requirements. Activity C is assigned r_C number of resources and the remaining resources are assigned to activity D. The utility gained (g_C) should be maximized by use of

$$g_C(u) = \max_{r_C} \left\{ \frac{5}{18} r_C + g_D(u - 5r_C) \right\} \quad (7)$$

where $5r_C \leq 12$ and $r_C \geq 0$.

The resource-assignment strategies (s_C) and resulting utility gains (g_C) are as follows (marked with an asterisk is the strategy that in each case results in maximum utility gains under the circumstances):

$$g_C(12) = \max \begin{cases} \frac{5}{18}(0) + g_D(12) = \frac{12}{18} & s_C = 0 \\ \frac{5}{18}(1) + g_D(7) = \frac{11}{18} & s_C = 1 \\ \frac{5}{18}(2) + g_D(2) = \frac{10}{18} & s_C = 2 \end{cases}$$

Similarly, for activity B the utility gained (g_B) is maximized by use of,

$$g_B(u) = \max_{r_B} \left\{ \frac{3}{18} r_B + g_C(u - 3r_B) \right\} \quad (8)$$

where $3r_B \leq 12$ and $r_B \geq 0$.

$$g_B(12) = \max \begin{cases} \frac{3}{18}(0) + g_C(12) = \frac{12}{18} & s_B = 0 \\ \frac{3}{18}(1) + g_C(9) = \frac{9}{18} & s_B = 1 \\ \frac{3}{18}(2) + g_C(6) = \frac{8}{18} & s_B = 2 \\ \frac{3}{18}(3) + g_C(3) = \frac{9}{18} & s_B = 3 \\ \frac{3}{18}(4) + g_C(0) = \frac{12}{18} & s_B = 4 \end{cases}$$

Finally, for activity A the utility gained (g_A) should be maximized by use of

$$g_A(u) = \max_{r_A} \left\{ \frac{4}{18} r_A + g_B(u - 4r_A) \right\} \quad (9)$$

where $4r_A \leq 12$ and $r_A \geq 0$.

$$g_A(12) = \max \begin{cases} \frac{4}{18}(0) + g_B(12) = \frac{12}{18} & s_A = 0 \\ \frac{4}{18}(1) + g_B(8) = \frac{12}{18} & s_A = 1 \\ \frac{4}{18}(2) + g_B(4) = \frac{8}{18} & s_A = 2 \\ \frac{4}{18}(3) + g_B(0) = \frac{12}{18} & s_A = 3 \end{cases}$$

Thus, based on the previous utility-based calculations the constructor arrives at the following options in terms of resource allocation strategies, subject to the immediate resource constraint of 12 resources in total:

- (1) All 12 resources assigned to activity D
- (2) All 12 resources assigned to activity C
- (3) All 12 resources assigned to activity B
- (4) 4 resources assigned to activity A, 8 resources assigned to activity B
- (5) All 12 resources assigned to activity A
- (6) 4 resources assigned to activity A, 3 resources assigned to activity B, and 5 resources assigned to activity C
- (7) 6 resources assigned to activity D, 6 resources assigned to activity B

It should be noted that, (i) the last two options are deduced by branching option 4 and option 1 respectively to make better utilization of the assigned resources, and (ii) options 2 and 4, even though deduced from the previous calculations, should eventually be excluded from consideration within the context of this paper since they involve non-integer crew assignments to activities C and B respectively (an assumption was made earlier in this paper that only integer multipliers of crew assignments would be considered for each activity).

Table 2. Resource Assignments and Resulting Activity Durations

Option	A		B		C		D	
	Res	Dur	Res	Dur	Res	Dur	Res	Dur
1	0		0		0		12	3/2
2	0		0		12	10/2	0	
3	0		12	3/4	0		0	
4	4	8	8	3/2	0		0	
5	12	8/3	0		0		0	
6	4	8	3	3	5	10	0	
7	0		6	3/2	0		6	3

By applying option 1, we would not start activities A, B and C at day one. Instead we would start activity D first and expedite it by using 12 resources (Fig. 3). By applying option 2 we would only start activity C at day 1 (using double the resources required by the activity), leaving all

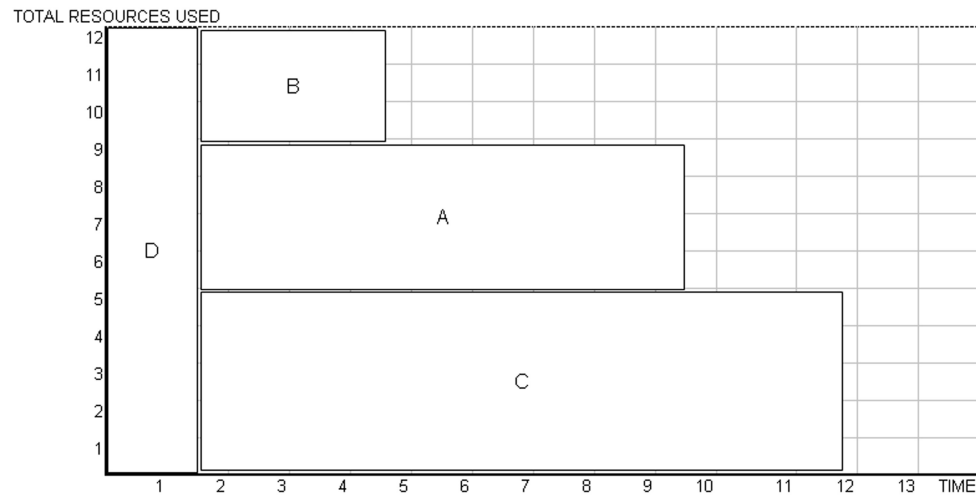


Fig. (3). Suggested optimized resource allocation (Option 1).

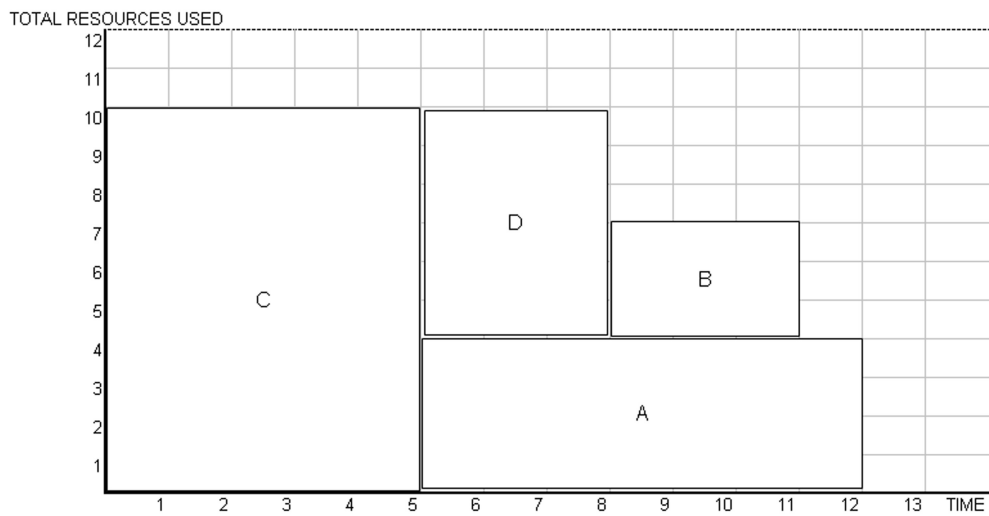


Fig. (4). Suggested optimized resource allocation (Option 2).

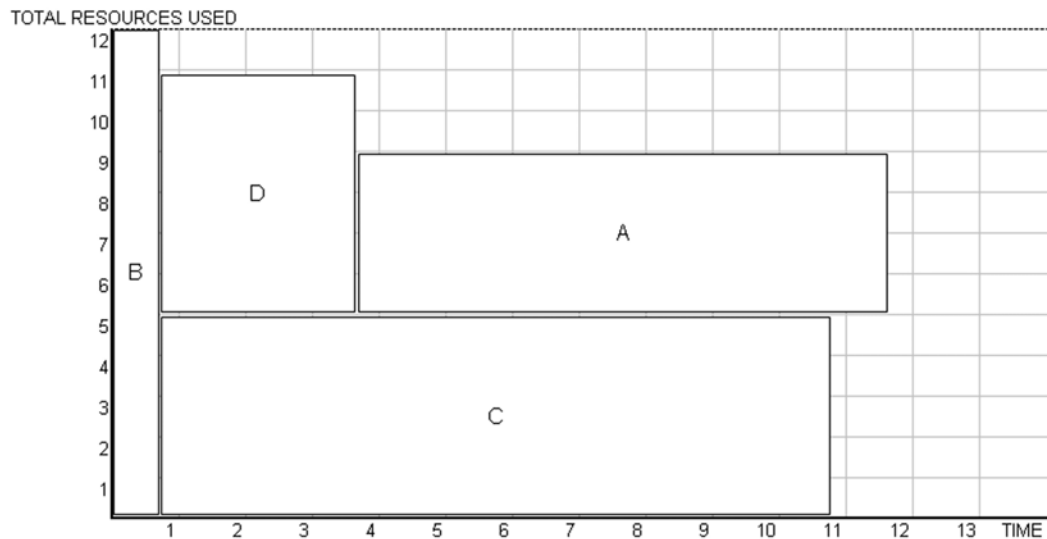


Fig. (5). Suggested optimized resource allocation (Option 3).

other activities until after C is completed (Fig. 4). Option 3 starts activity B first with 4 crews (Fig. 5), while option 4 has activities A and B starting concurrently with activities C and D following Fig. (6). Option 5 has all 12 resources

assigned to activity A (Fig. 7), and option 6 has A, B and C starting concurrently (Fig. 8). Finally, strategy 7 has activity B and D starting concurrently with B expedited by using 2 crews on it (Fig. 9).

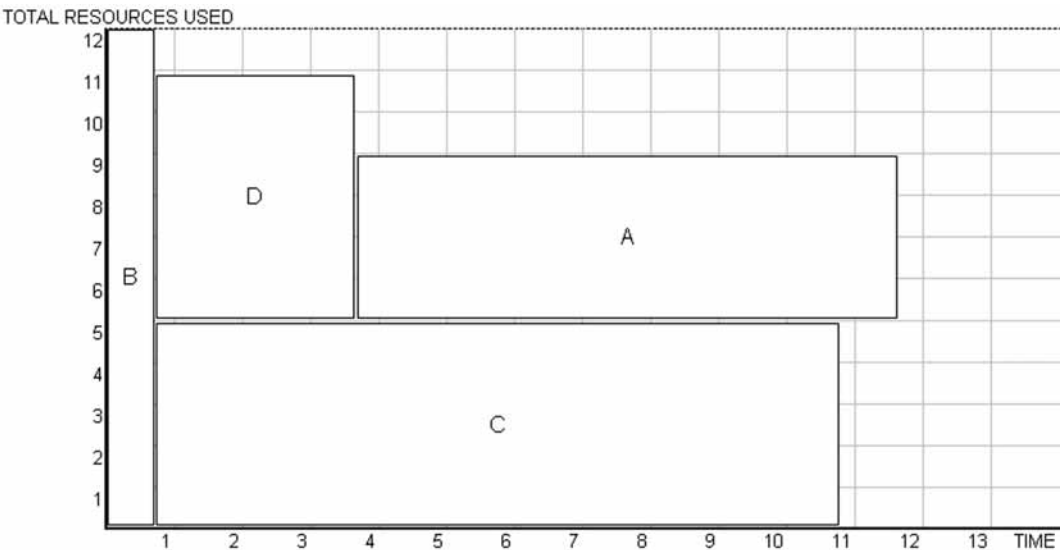


Fig. (6). Suggested optimized resource allocation (Option 4).

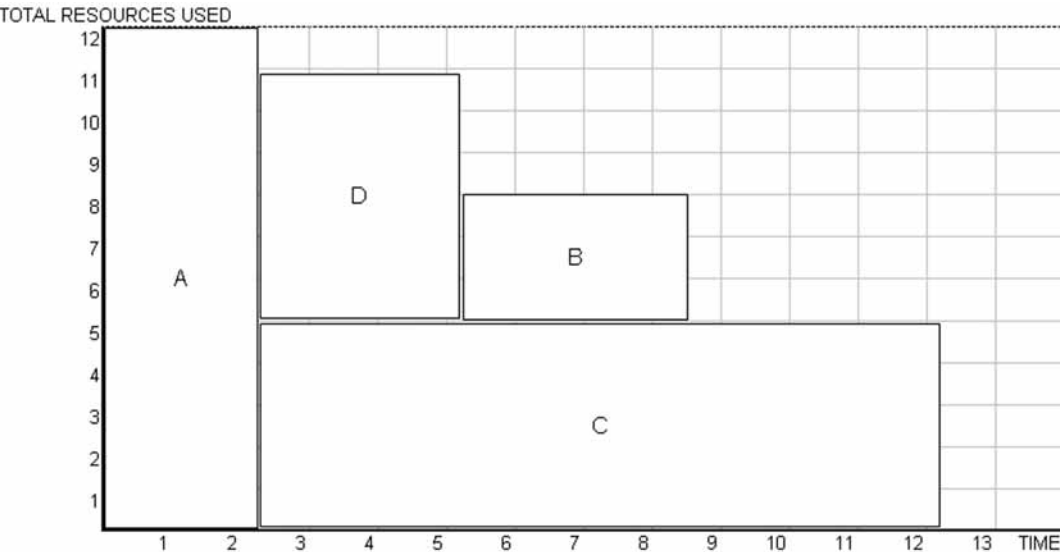


Fig. (7). Suggested optimized resource allocation (Option 5).

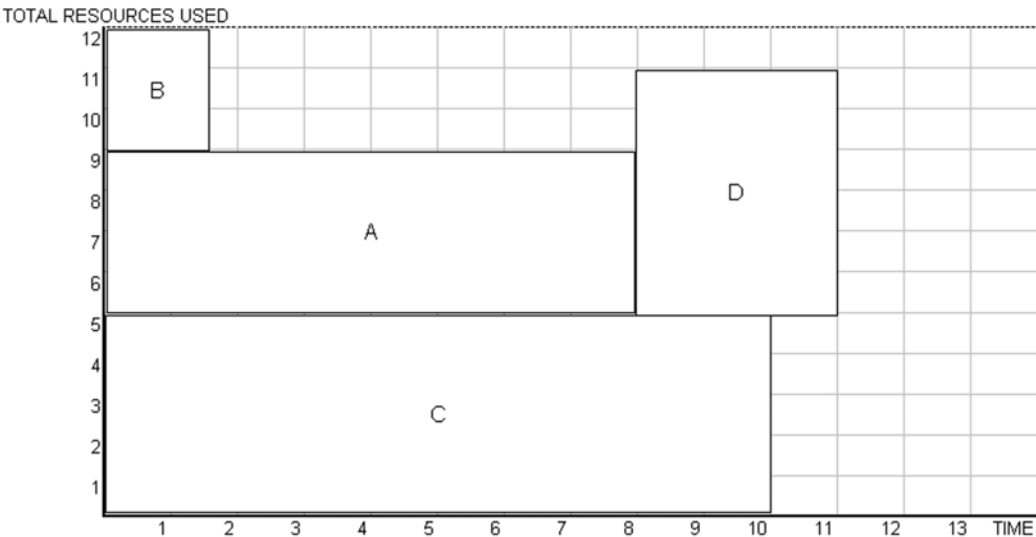


Fig. (8). Suggested optimized resource allocation (Option 6).

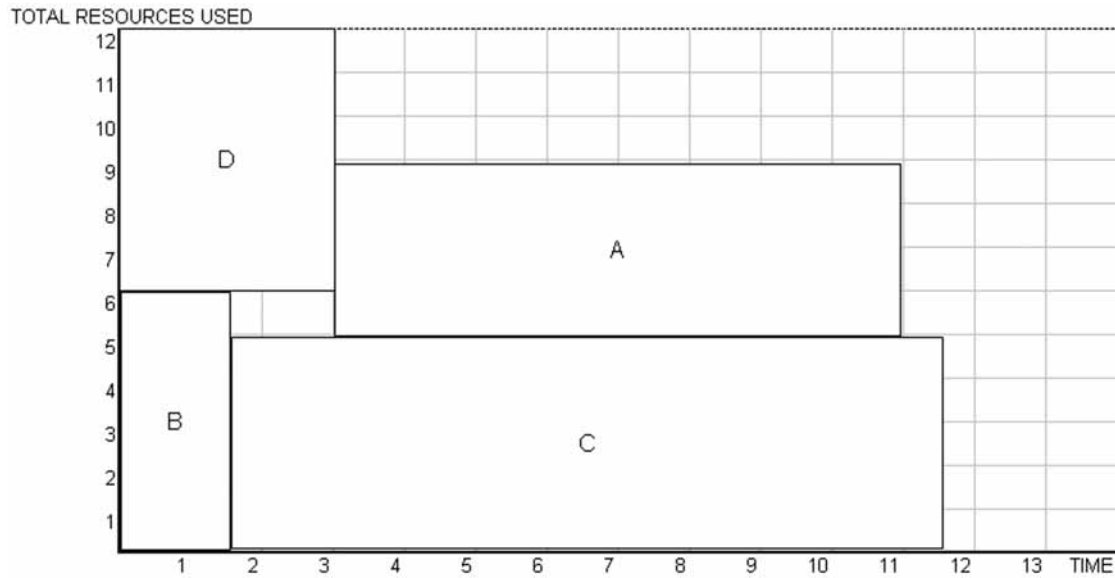


Fig. (9). Suggested optimized resource allocation (Option 7).

The resulting project durations and critical paths for each one of these cases are tabulated in Table 3. It can be seen that the utility-index approach indicates option 6 as the best strategy to follow in terms of allocating our limited resources to the predefined set of immediate activities in our network. This is in contrast to the CPM-based impulsive strategy to primarily divert all resources to critical activities (Activity C in this case) in the hope that the overall critical path duration would decrease.

Table 3. Resulting Alternative Project Durations and Critical Activities

Option	Critical Path	Project Duration
1	D-C	11.50
2	C-A	12.00
3	B-D-A	11.75
4	B-C	11.50
5	A-C	12.67
6	A-D	11.00
7	B-C	11.50

CONCLUSIONS

Construction is a complex process that requires the employment of many different resources (labor, material, equipment). Quite often, either in their attempt to simplify the job of developing a schedule or under time constraints to develop it quickly, construction planners prepare a construction schedule under the assumption that all the resources needed for each activity will be available to the constructor at any time during the construction process. In the presence of resource constraints, though, development of a construction schedule implies work prioritization mechanisms that can help divert resources to the activities that need them the most and that can eventually result in the shortest overall

project duration. When concurrent activities are competing for the same resources they may eventually need to be performed sequentially; a necessity which may have a negative effect on the project's duration if the activities are critical or near-critical. The paper presented a method to optimize the allocation of limited resources so as to minimize the impact of their unavailability on any proposed project schedule. The assignment of limited resources to individual activities is determined by means of an algorithm that maximizes the use of available resources, defined herein as maximizing the utility index in the shortest amount of time. The utility index of a given resource for any activity is the ratio of the amount of that required resources for that activity to the total required amount of that resource for all competing activities. If there are multiple limited resources required by two or more activities, the resource with the least production rate becomes the driving resource.

REFERENCES

- [1] J. Lenstra and A. R. Kan, "Scheduling subject to resource constraints classification and complexity", *Discrete Applied Mathematics*, vol. 5, pp. 11-24, 1983.
- [2] P. Brucker, S. Knust, A. Schoo and O. Thiele, "A branch and bound algorithm for the resource-constraint project scheduling problem", *European Journal of Operational Research*, vol. 107, pp. 272-288, June 1998.
- [3] E. Demeulemeester and W. Herroelen, "New benchmark results for the resource-constraint project scheduling problem", *Management Science*, vol. 11(43), pp. 1485-1492, November 1997.
- [4] A. Mingozzi, V. Maniezzo, S. Ricciardelli and L. Bianco, "An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation", *Management Science*, vol. 44, pp. 714-729, May 1998.
- [5] J. M. Crawford, "An Approach to Resource Constrained Project Scheduling", in *Proceedings of Artificial Intelligence and Manufacturing Research Planning Workshop*, 1996.
- [6] M. Garey, R. Graham, D. Johnson and C. Yao, "Resource constrained scheduling as generalized bin packing", *Journal of Combinatorial Theory*, vol. 21, pp. 257-298, 1976.
- [7] W. L. Winston and M. Venkatarmanan, *Introduction to mathematical programming*. Thomson, 4th ed., 2003.

- [8] J. A. Carruthers and A. Battersby, "Advances in critical path methods", *Operational Research Society*, vol. 17(4), pp. 359-380, December 1966.
- [9] J. H. Patterson, "A comparison of exact approaches for solving the multiple constrained resource project scheduling problem", *Management Science*, vol. 30, pp. 854-867, July 1984.
- [10] T. Hegazy, "Optimization of resource allocation and leveling using genetic algorithms", *Journal of Construction Engineering and Management*, vol. 125(3), pp. 167-175, June 1999.
- [11] S. Christodoulou, "Ant Colony Optimization in construction scheduling", in *Proceedings of ASCE International Conference on Computing in Civil Engineering*, 2005.
- [12] S. Christodoulou, "Scheduling Construction Activities Using Ant Colony Optimization", in *Proceedings of the Eighth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, 2005.
- [13] S. Christodoulou, "Resource-Constrained Scheduling Using Ant Colony Optimization", in *Proceedings of the Ninth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, 2007.
- [14] P. Brucker and S. Knust, "Lower bounds for resource-constrained project scheduling problems", *European Journal of Operational Research*, vol. 149, pp. 302-313, 2003.
- [15] P. Aslani, *Dynamic resource-constrained scheduling*, Ph.D. thesis, Department of Civil Engineering, Polytechnic University, New York, 2007.
- [16] S. Colak, A. Agarwal and S. Erenguc, "Resource constrained project scheduling: A hybrid neural approach", *International Series in Operations Research and Management Science*, Kluwer Academic, vol. 92, pp. 297-320, 2006.
- [17] Y. Liu, S. Zhao, X. Du and Q. Li, "Optimization of Resource Allocation in Construction using Genetic Algorithms", in *Proceedings of International Conference on Machine Learning and Cybernetics*, IEEE, 2005.
- [18] E. V. Denardo, *The Science of Decision Making*, John Wiley, 2004.
- [19] A. B. Senouci and N. Eldin, "Use of Genetic Algorithms in Resource Scheduling of Construction Projects", *Journal of Construction Engineering and Management*, ASCE, vol. 130(6), pp. 869-877, December 2004.
- [20] S. Sakellariopoulos and A. P. Chassiakos, "Project time-cost analysis under generalized precedence relations", *Advances in Engineering Software*, vol. 35(10-11), pp. 715-724, November 2004.
- [21] Primavera Systems, *Primavera Project Planner Electronic Manual*, Primavera Systems, Inc., 2001.
- [22] Microsoft Corporation, *Microsoft Project Electronic Manual*, Microsoft Corporation, 2000.

Received: September 29, 2008

Revised: December 04, 2008

Accepted: February 09, 2009

© Aslani et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.