Hierarchal Model for Solving Resource-Constrained Project Scheduling Problem under Minimum Makespan Constraint

Khaled Z. Ramadan
University of Petra, department of civil engineering, Amman 11196 Jordan,
E-mail: khzera66@gmail.com

***Abstract*-- In this paper a hierarchal model for Resource-Constrained Project Scheduling (RCPS) problem is presented. The model consists of two steps; step 1 applies the longest path integer programming model to determine the critical path activities and the free floats of the non-critical path activities. In step 2, Genetic algorithm exploits the information found in step 1 to determine the start times of the non-critical activities such that the total cost of resources needed to perform the project is minimized. This proposed model differs from the conventional RCPS problem in that the proposed model seeks to minimize the total cost of resources to perform the project within the minimum makespan of the project while the conventional RCPS seeks to minimize the total cost of resources usually by extending the duration of the project beyond its minimum makespan. The effectiveness of the model were demonstrated using two illustrative examples. The results of these examples showed that the proposed model is effective in scheduling the non-critical activities within the minimum makespan of the project such that the total cost of resources needed to perform the project is minimized. This proposed model can be effectively used in projects that are very sensitive to durations such as steel bridge construction projects, multi-level steel building projects, air craft maintenance, any other major construction projects, or where ever it is applicable.**

***Index Term*-- RCPS, Genetic Algorithm, Project Scheduling, Longest Path Model, Makespan.**

INTRODUCTION

Construction projects nowadays became very sensitive to the total project duration. Altering the duration of any project may have a great impact on the total cost of the project. Project scheduling is the tool to determine the start times of the project activities under precedence and resources constraints. The precedence relations determine the order in which the activities must be performed. Generally speaking, the objective of project scheduling is to minimize the makespan of the project. Critical path method, CPM, is usually used for this purpose but the project scheduling problem can also be formulated and solved as longest path problem. The Resource-Constrained Project Scheduling RCPS is an extension for project scheduling under which it is assumed that the activities need certain quantity of resources such as materials, equipments, funding, or manpower to start. It is further assumed that these resources are of limited capacities. Due to these limitations some of these activities may be delayed, which in turn delay the duration of the whole project. Blazewicz et.al(1983) shows that RCPS problem can be seen as a generalized form of job shop scheduling problem and it belongs to NP-hard optimization problems.

Since Kelly (1963) proposed a schedule generation scheme, many heuristic algorithms have been suggested. Three major methods were extensively used in the literatures for solving RCPS; implicit enumeration with branch and bound by Dorndorf et al. (2000) and Brucker et al. (1998), dynamic programming by Battersby and Carruthers (1966), and zero-one programming by Patterson and Huber (1974) and Patterson and Roth (1976). Lately, evolutionary algorithms find its path into RCPS problem solving. Bartchi (1996), used genetic algorithm to solve the multi-model RCPS problem and Jozefowska et al. (2001) used simulated annealing to solve the same problem. Selcuk et al. (2013) discussed an approach for multi-mode resource-constrained project scheduling using two non-greedy heuristics for mode selection. Hartmann (1998) used a combination of local search and genetic algorithm to solve the RCPS problem while Merckle et al. (2002) used ant colony analogy. Lova et al. (2009) used hybrid genetic algorithms with multiple execution modes for the problem. Chen and Chyu (2009) proposed four memetic algorithms that differ in initial population, restart strategy, and double variable neighborhood search algorithm to solve for RCPS problem. They compared their model with Hartmann and Briskorn (2010) model.

The underlying assumption of RCPS problem is that the process of each activity requires a predefined amount of resources from limited amounts of resources available at any epoch. This means that if the total amount of a certain type of resource is not enough in any epoch to support the activities in that epoch, then some of the activities cannot be processed simultaneously and must be processed sequentially, which in turn may lead to an increase in the makespan of the project. This makes RCPS problem solving techniques resort to extending the project beyond its minimum makespan.

In many situations, the duration of the project is much more important than the cost of the extra resources needed to perform the project within its minimum makespan. In these cases the liquidated damage cost exceeds the cost of adding more resources to complete designated activities. Construction of long-span steel bridge projects and multi-level steel building projects are applicable examples that can be used to demonstrate the present concept. For example, it is estimated that the cost of the downtime per crane is about $15,000, while the liquidated damage due to one day delay is estimated to be about $100,000.

Moreover, the bidding process on some construction projects depends heavily on the duration of the project such that the low duration is considered an advantage. These

examples and many other examples call for scheduling such projects in a way giving the duration of the project higher priority over the resources cost. In this paper a model for RCPS is proposed under minimum makespan constraint.

In the proposed model, the project activities are scheduled such that the total cost of resources is minimized under precedence and minimum makespan constraints. The total resources cost is calculated based on the optimum number of resource units needed for each resource type to complete the project. The minimum makespan is found using longest path integer programming model whereas genetic algorithm is used to find the minimum total cost of resources within that makespan. The start times of the non-critical activities will be scheduled according to their early start times and free floats such that the minimum total cost of resources is achieved under minimum makespan and precedence constraints.

The proposed model is a hierarchal model with two steps involved; the first involves in calculating for the minimum makespan and determining the critical path activities whereas the second involves in determining the start times for the non-critical path activities. The first step will be achieved using longest path integer programming model and the second step will be achieved using genetic algorithm with the aid of the free floats of the non-critical path activities from determined in the first step.

MODEL FORMULATION

Consider a project that consists of:

a. A set of n activities, $\boldsymbol{a} = \{a_1, a_2, \dots, a_j, \dots, a_n\}$, where a_1 and a_n are two dummy activities denote the start and the end of the project respectively.

b. A set of durations $\boldsymbol{d} = \{d_1, d_2, \dots, d_j, \dots, d_n\}$ such that d_j is the duration of activity a_j.

c. A set of $\boldsymbol{N_j}$ predecessors for each activity; $\boldsymbol{P_j} = \{P_{j1}, P_{j2}, \dots, P_{jp}, \dots, P_{jN_j}\}$ such that P_{jp} is the p^{th} predecessor activity of activity a_j.

d. A set of K renewable resources, $\boldsymbol{R_j} = \{R_{j1}, R_{j2}, \dots R_{jk}, \dots, R_{jK}\}$, for each activity a_j such that R_{jk} is the number of units needed from resource k for activity a_j.

Let s_{a_j} and f_{a_j} denote the start and the end times for activity a_j respectively, and let c_k denotes the cost of acquisition of one unit of resource R_{jk}. Moreover, let $r_{t,k}$ denotes the number of units needed from resource k at any epoch t for all activities. The acquisition cost of resource k for the project can be calculated as:

$$c_k \max_{t\in\{0:f_{a_n}\}}\{r_{t,k}\}, \quad (1)$$

where $\max_{t\in\{0:f_{a_n}\}}\{r_{t,k}\}$ gives the number of units needed from resource k in the minimum makespan f_{a_n}. Mathematically the total acquisition cost of the resources for the project is calculated as:

$$\sum_{k=1}^{K} c_k \max_{t\in\{0:f_{a_n}\}}\{r_{t,k}\}, \quad (2)$$

where c_k is the acquisition cost of one unit of resource k. Equation (2) represents the objective function of the proposed mode.

The longest path linear programming model is well documented in Taha (2007). To find the longest path for the project, a one unit of flow is assumed to enter from node 1 and leaves at node W where nodes 1 and W represent the start and the end nodes respectively.

Let x_{uw} be the amount of flow in the activity a_j defined by the $arc\ (u,w)$ between nodes u and w such that $x_{uw} = 1$ if the activity a_j is on the longest path and zero otherwise. Moreover, let $D_{uw} = d_j$ which denotes the duration of the activity a_j defined by the $arc\ (u,w)$. The linear programming for the longest path can be modeled as:

$$\max \vartheta = \sum_{\substack{all\ defined \\ arcs\ (u,w)}} D_{uw} x_{uw} \quad (3)$$

s.t

$$\begin{pmatrix} External\ input \\ into\ node\ w \end{pmatrix} + \sum_{\substack{u \\ all\ defined \\ arcs\ (u,w)}} x_{uw} = \begin{pmatrix} External\ outpu \\ from\ node\ w \end{pmatrix} + \sum_{\substack{l \\ all\ defined \\ arcs\ (w,l)}} x_{wl} \quad (4)$$

$$x_{uw} \geq 0 \quad (5)$$

where Equation (3) is the objective function that finds the minimum makspan of the project which is the longest path of the network. Equation (4) represents the conservation of flow equation at each node and it enforces the precedence relationship between the activities.

The full formulation of the proposed model can be written as:

$$\min \sum_{k=1}^{K} c_k \max_{t\in\{0:f_{a_n}\}}\{r_{t,k}\}$$

s.t

$$f_{a_n} = \vartheta$$
$$f_{P_{jp}} \leq s_{a_j} \; \forall \, j = 2, 3, \dots, n \; and \; p = 1, 2, \dots, N_j$$
$$f_{a_j} \geq 0 \;\; \forall \, j = 1, 2, 3, \dots, n,$$

Genetic Algorithm

Genetic Algorithm GA is an evolutionary algorithm based on Darwin's theory. The principle is simple; the strong individuals fit better to their surrounding and hence they survive and pass their genes to the next generations while the weak individuals, those who fit less to their surrounding, extinct with time and hence cannot pass their genes to the next generations. This evolutionary process suggests that the new individuals are better than their ancestors as the new individuals are fitter to their surroundings.

GA, as an evolutionary algorithm, is characterized by a population that evolves over generations. GA has two operators; crossover and mutation. The crossover operator is used to explore the sample space for promising areas as this operator causes a big change in the chromosome while mutation operator is used to exploit the promising regions found by the crossover operator as it causes a limited change in the chromosome.

GA uses chromosome representation to encode the objective function of the optimization model into a genotype representation. In the proposed model, the chromosome representation is a position-based chromosome which consists of n genes. Each gene carries two pieces of information; the gene location represents the activity number and the gene value represents the start time of that activity. Figure 1 shows a chromosome for 5-activity project. As the figure shows, the chromosome contains 5 genes. For example the first gene represents the first activity and its value denotes the start time of that activity which is 4.

4	7	3	5	11

Fig. 1. Chromosome representation for 5-activity project

Because the chromosome representation for this GA is position-based chromosome, the crossover operator will not be used since this operator will create many infeasible offspring. The crossover operator will be replaced by a heavy-mutation operator. The heavy-mutation operator will have the same effect as the crossover operator but will conserve the location of the gene and thus will generate feasible offspring. The heavy-mutation operator works as follows:

1- For each chromosome in the population, select $ceil\left(\frac{n}{2}\right)$ random genes from the chromosome.
2- For these selected genes, replace the values of the genes as follows: for each selected gene, select a random number between the early start time of the corresponding activity and the sum of the early start time and the free float of that activity. Replace the value of the gene with this number. This way of mutation guarantees that the offspring is feasible.

The mutation operator works the same as the heavy-mutation operator but it works only on one gene from the whole chromosome and its purpose is to do a fine search or exploitation. The heavy-mutation operator will be the main operator used in the early generations to explore the sample space adequately and the mutation operator will be used extensively used toward the end of the generations to exploit the accumulated information over the generations about the promising regions.

Experimentations

Consider a 9-activity project with 3 different resources. Table 1 shows the duration, precedence, and the resources needed for each activity. Figure 2 shows an Activity-on-Arrow network diagram for this project. Activity 8 is a dummy activity and hence it has zero duration and resources.

Table I
Pertaining data for the 9-activity project

Activity	Duration	Precedance
Start	0	-
A	5	-
B	6	-
C	3	A
D	8	A
E	2	C, B
F	2	C, B
G	1	D
H(dummy)	0	D
I	12	H, E
End	0	G, F, I

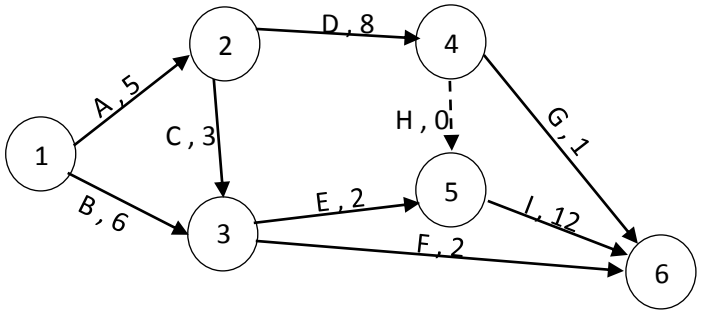

Fig. 2. Network diagram for the 9-activity project

The critical path for this project was found using step 1 of the proposed model using the longest path linear programming model. Table 2 shows the output of step 1 of the propose model. The table shows the critical path activities in ***italic bold***, duration, precedence, early start, late start, free float, total float, and Start Range for each activity. The minimum makespan for this project was found by step 1 to be 25 units of time.

Table II
Start Range for the 9-activity project

Activity	Early start	Late Start	Total Float	Free Float	Start Range
A	0	0	0	0	0
B	0	0	5	2	0-2
C	5	5	3	0	5
D	5	5	0	0	5
E	8	11	3	3	8-11
F	8	11	15	15	8-23
G	13	13	11	11	13-24
H	13	13	0	0	13
I	13	13	0	0	13

Table II shows the range of start-up times for the non-critical activities B, E, F, and G. For example activity B can be started anywhere between zero and 2 without causing any delay in the project makespan or a delay in its successors. These ranges in start times always the scheduler, in some occasions, to schedule the activities that need the same resources sequentially i.e. without overlapping; therefore, reducing the maximum number of resources needed to accomplish the project. This reduction in the number of resources reduces the total cost of resources calculated by Equation (2).

Genetic algorithm is used to schedule these non-critical activities within their start ranges. For this particular project, one possible chromosome is shown in Figure 3.

0	1	5	5	10	20	15	13	13

Fig. 3. One possible chromosome for the 9-activity project.

In this chromosome each gene represents an activity, for example gene number 3 represents activity C and its value i.e., 5 represents the start time of this activity. Note that this chromosome uses the Start Range of the activity to assign its start time. For example activity G is scheduled to start at 15 which is in the Start Range of this activity.

To do a crossover for this chromosome, 5 random genes $\left(ceil\left(\frac{9}{2}\right)\right)$ from the chromosome is selected. Assume that these genes are {3, 1, 5, 8, 2}. These genes corresponds to the following activities {C, A, E, H, B}. The genes values for these activities will be replaced by a randomly selected numbers from the corresponding start ranges of these activities. The new genes values can be {5, 0 ,11 ,13, 2}. These values will replace the values in the parent chromosome in Figure 3 to produce the offspring. Figure 4 shows the offspring generated by the heavy-mutation operator. Note that the heavy-mutation operator preserve the feasibility of the offspring as the genes will only assume feasible values in their start ranges.

0	2	5	5	11	20	15	13	13

Fig. 4. Offspring of parent chromosome generated by the heavy-mutation operator.

The mutation operator works exactly as the heavy-mutation operator but on one gene only. For example if gene number 6 is selected randomly, then the value of this gene is replaced by a random value from the Start Range of activity F say 9. The mutated offspring is then given by Figure 5.

0	2	5	5	11	9	15	13	13

Fig. 5. Mutated offspring generated by mutation operator.

Let us assume that there are three types of resources needed in this project with unit cost of $1000, $100, $1000 respectively. Table 3 shows the activities along with their required resources for this scenario.

Table III
Resources needed for the activities in the first scenario of the 9-activity project

Activity	Resources	Start Range
A	R1, R3	0
B	R2	0-2
C	-	5
D	R1	5
E	R3	8-11
F	R1, R2	8-23
G	R2, R3	13-24
H	-	13
I	-	13

Table III shows that activities A, D, and F need resource R1. Since activity A and activity D are on the critical path, there will be no overlap between them and hence, only one resource is enough to do both activities. On the other hand, F can be scheduled anywhere between 8 and 23. If this activity is scheduled anywhere between 8 and 13, two units of R1 is needed since activity F and activity D will overlap. If activity F is scheduled anywhere between 14 and 23 only one unit of R1 is needed since no other activity in this range needs R1. Moreover if activity F is scheduled such that it overlaps with activity G, then two units of R2 are needed. This way of scheduling is possible for small number of activities, but it can be very tedious for large number of activities.

Step 2 of the proposed method is aimed to do this tedious job using GA. The best solution found for this scenario is represented by the chromosome in Figure 6 and the best value for this situation is $11100 which is the optimal value as one unit only of each resource is needed in the project.

0	0	5	5	11	20	13	13	13

Fig. 6. Optimal solution found for the first scenario.

Figure 7 shows a Gantt chart for this solution.

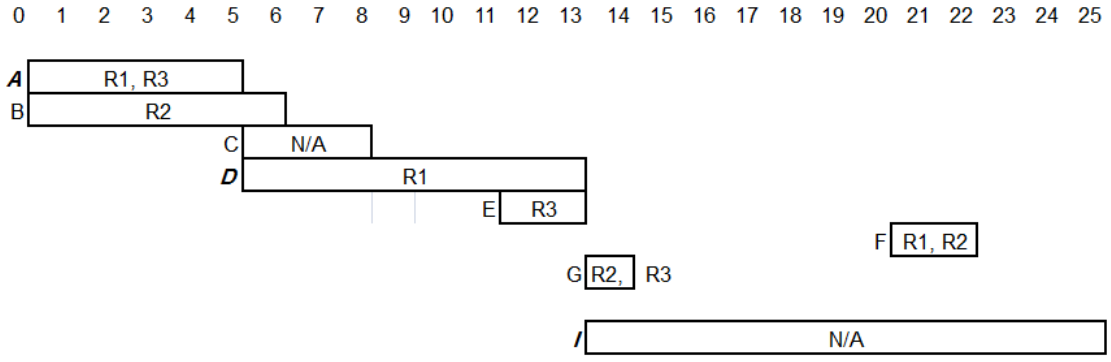

Fig. 7. Gantt chart for the first scenario

It should be noticed from the Gantt chart that the GA scheduled the activities such that there is no overlap between the activities that use the same resources. The GA scheduled activity G to start from 13, activity F to start from 20, and activity E to start from 11 to avoid this overlap. Activity B was started at 0 as there are no common resources between this activity and activity A. the overall cost of resources is the minimum cost of $11100.

Consider another scenario for the resources such that activity I needs resources R2 and R3 instead of activity G. Table IV shows the pertinent data for this scenario

Table IV
Resources needed for the activities in the second scenario

Activity	Resources	Start Range
A	R1, R3	0
B	R2	0-2
C	-	5
D	R1	5
E	R3	8-11
F	R1, R2	8-23
G	-	13-24
H	-	13
I	R2, R3	13

The best solution found for this scenario is represented by the chromosome in Figure 8 and the best value for this scenario is $11200 which is the optimal solution.

0	0	5	5	8	13	14	13	13

Fig. 8. Best chromosome found for the second scenario.

Figure 9 shows the Gantt chart for this solution.

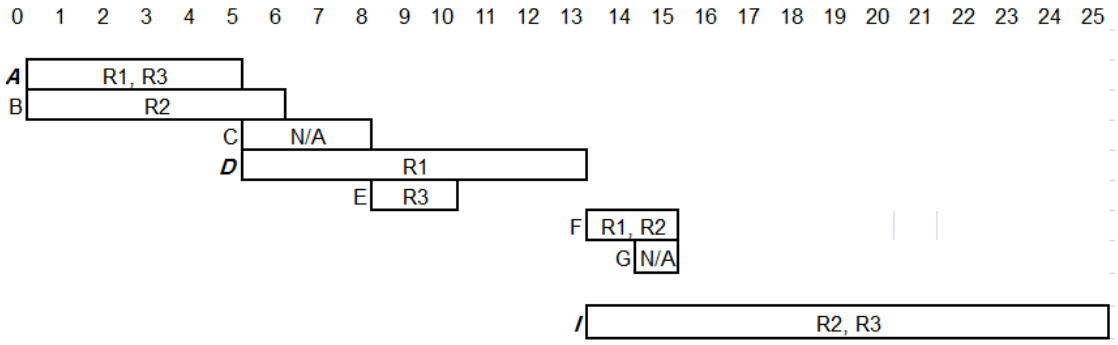

Fig. 9. Gantt chart for the second scenario

Note that activity F must overlap with either activity D with the resource R1 or activity I with the resource R2. Since the cost of one unit of R2 is cheaper than one unit of resource R1, GA chose to overlap activity F with activity I and to have 2 units of resource R2 instead of 2 units of R1 such that the total resources cost is minimum i.e., $11200. This shows the effectiveness of this model and how the model responds to the different scenarios.

To explore the effect of the unit resource cost, the third scenario increased the cost of resource R2 from $100/unit to $50000/unit. The same data in Table 4 was solved again. The best solution found for this scenario is represented by the chromosome in Figure 10 which is the optimal solution with optimal value of $71000.

0	2	5	5	11	11	13	13	13

Fig. 10. Optimal solution found for the third scenario.

Figure 11 shows the Gantt chart for this solution.

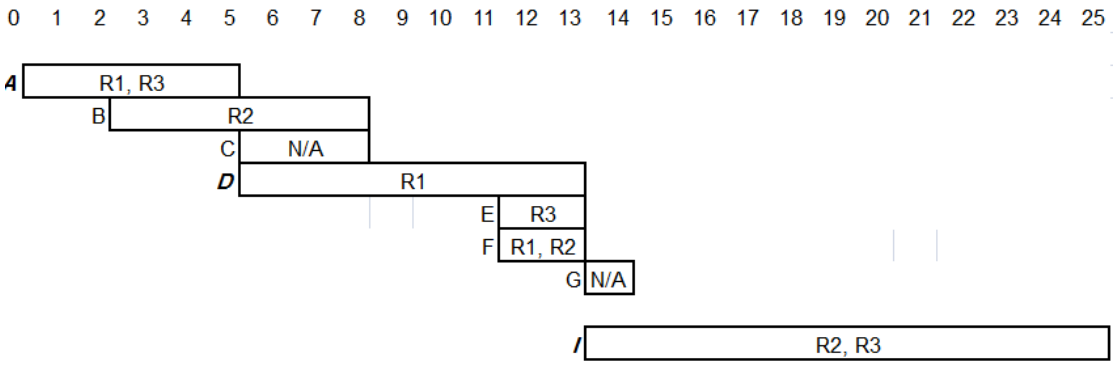

Fig. 11. Gantt chart for the third scenario

Note that the GA started activity F at 11 to avoid the overlap in R2 with activity I as in this scenario R2 costs more than R1. Again this shows the effectiveness of this model and how it adapts with different scenarios.

The last scenario for this example is when all of the 8 activities need some types of resources in different quantities. Table V shows the pertinent data for this scenario.

Table V
Resources needed for the activities in the fourth scenario

Activity	Resources	Start Range
A	3×R1, R3	0
B	R2	0-2
C	R1, 2×R2	5
D	R1	5
E	R3	8-11
F	R1, R2	8-23
G	2×R3	13-24
H	-	13
I	R2, R3	13

The best solution found for this scenario is represented by the chromosome in Figure 12 which is the oprimal solution with optimal values of $31300.

0	1	5	5	8	11	13	17	13

Fig. 12. Optimal solution found for the fourth scenario.

Figure 13 shows the Gantt chart for this solution.

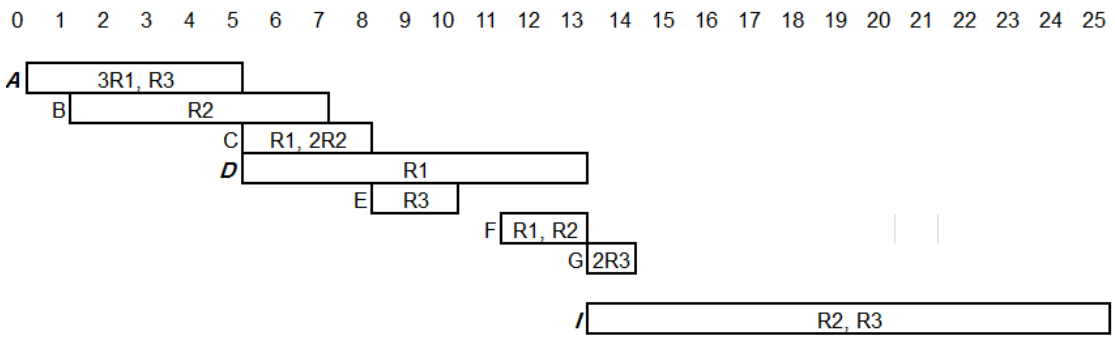

Fig. 13. Gantt chart for the fourth scenario

Note that activity A needs 3 units of resource R1 and this number cannot be avoided since it is needed in one activity. The GA started activity F at 11 to avoid the overlap in R2 with activity I even though it overlaps in R1 with activity D. The overlap between activities F and D does not matter in this case because the project must have three units of resource R1 for activity A anyways, hence the sequential scheduling of activities F and D does not help in reducing the overall number of units of resource R1. This shows again how this model adapt with the different scenarios

The second example is a 27-activity project as shown in Figure 14. The figure shows that activities D, K, Q, I, G, I, O, and P are dummy activities with zero duration and resource.

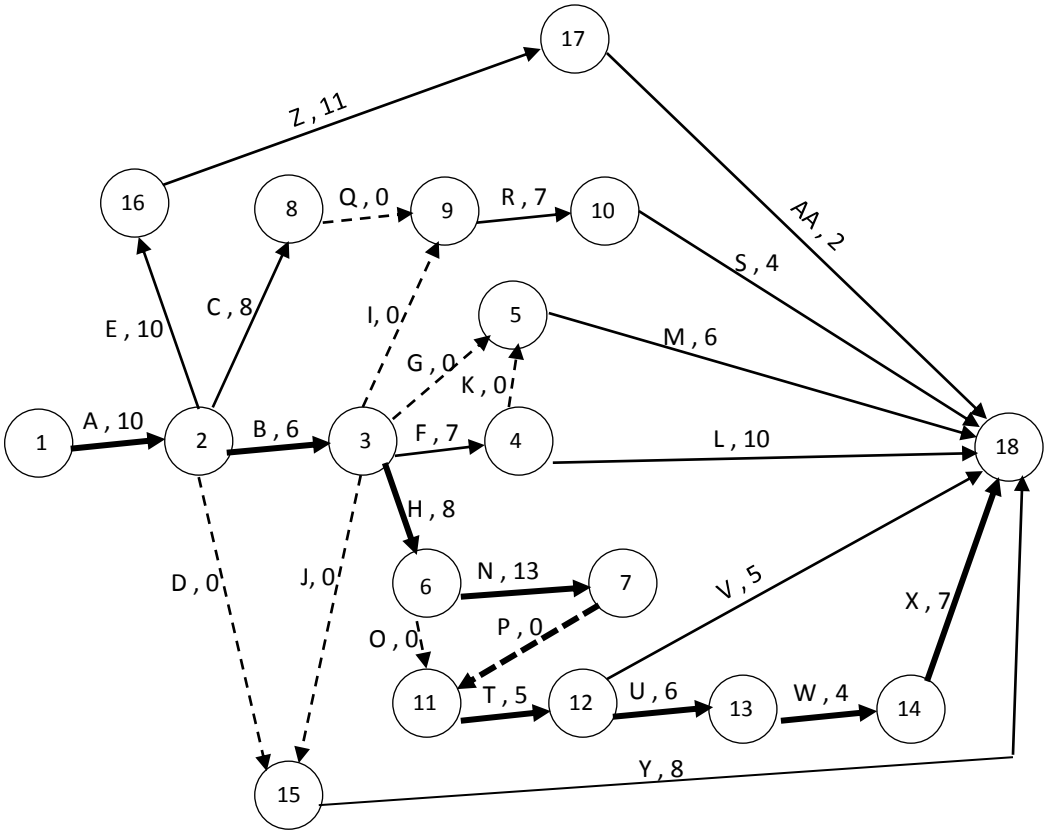

Fig. 14. Network diagram for 27-activities project.

Applying step 1 of the proposed model for this project gives the start ranges for the activities along with the free floats and the total floats as shown in Table VI. The minimum makespan for this project was found by step 1 to be 58.

Table VI
Start Range for the 27-activity project evaluated by step 1 of the proposed model

Activity	Early Start	Late Start	Total Float	Free Float	Start Range
A	0	0	0	0	0
B	10	10	0	0	10
C	10	10	29	0	10
D	10	15	40	5	10-15
E	10	10	25	0	10
F	15	15	26	0	15
G	15	22	37	7	15-22
H	15	15	0	0	15
I	15	18	32	3	15-18
J	15	15	35	0	15
K	22	22	30	0	22
L	22	48	26	26	22-48
M	22	52	30	30	22-52
N	23	23	0	0	23
O	18	36	13	13	18-31
P	36	36	0	0	36
Q	18	18	29	0	18
R	18	18	29	0	18
S	25	54	29	29	25-54
T	36	36	0	0	36
U	41	41	0	0	41
V	41	53	12	12	41-53
W	47	47	0	0	47
X	51	51	0	0	51
Y	15	50	35	35	15-50
Z	20	20	25	0	20
AA	31	56	25	25	31-56

Let the first scenario be that 9 of the activities need resources. Table 7 shows the resources needed for these activities.

Table VII
Resources needed in the activities for the first scenario of the 27-activity project

Activity	Start Range	Resources	Activity	Start Range	Resources
A	0	R2, 3×R3	***N***	23	R3
B	10	2×R2	O	18-31	-
C	10	-	***P***	36	-
D	10-15	-	Q	18	-
E	10	-	R	18	-
F	15	-	S	25-54	3×R3
G	15-22	-	***T***	36	-
H	15	-	***U***	41	-
I	15-18	-	V	41-53	-
J	15	-	***W***	47	2×R1, R2
K	22	-	***X***	51	-
L	22-48	R1, 3×R2, R3	Y	15-50	R1
M	22-52	2×R2, R3	Z	20	-
			AA	31-56	R1, 2×R2

The best solution found for this scenario is represented by the chromosome in Figure 15 which is again the optimal solution with optimal value of $ 33300.

0	**10**	**10**	**12**	**10**	**15**	**21**	**15**	**18**	**15**	**22**	**33**	**23**

23	**36**	**36**	**18**	**18**	**26**	**36**	**41**	**46**	**47**	**51**	**43**	**20**	**54**

Fig.15. Optimal solution found for the first scenario of the 27-activity project.

Figure 16 shows the Gantt chart for this solution. Note that the dummy activities are not included.

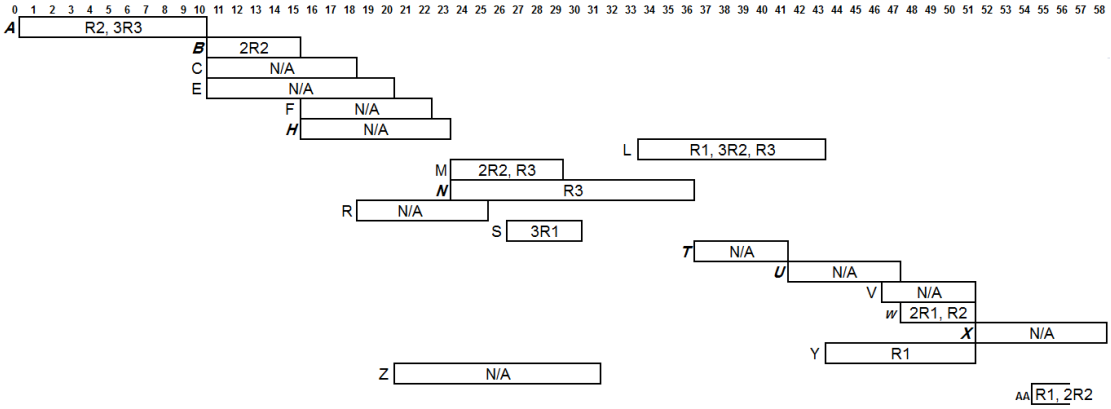

Fig. 16. Gantt chart for the first scenario of the 27-activity project

The best solution calls for having three resources from each type of the resource. Note that this solution is the optimal solution as activity S needs 3 units of R1, activity L needs 3 units of R2, and activity A needs 3 units of R3 thus 3 units of each resource are needed and cannot be avoided under any schedule as the project cannot be performed with less than three units of each type of the resources. The total cost of resources is $33300 which is the optimal value found by the model.

Note that activity S was scheduled to start from 26 even though its Start Range is between 25-54. The model chose this start time because if the activity started beyond 33, the number of units needed for R1 will be 4 or even 5 units. Moreover, the model chose to start activity L, that needs 3 unit of R2, from 33 even though its Start Range is between 22-48 to avoid the overlap with activity W that starts at 47 and need 1 units of resource R2. If there was an overlap between activities L and W, the total number of units needed for R2 will be 4 units but the model scheduled the activities such that they did not overlap.

Let us consider the scenario with the resources needed given in Table 8. The difference between this scenario and the previous scenario is that all the activities needed at most one unit of each type of the resources.

Table VIII
Resources needed for the activities in the second scenario of the 27-activity project

Activity	Start Range	Resources	Activity	Start Range	Resources
A	0	R2, R3	***N***	23	R3
B	10	R2	O	18-31	-
C	10	-	***P***	36	-
D	10-15	-	Q	18	-
E	10	-	R	18	-
F	15	-	S	25-54	R3
G	15-22	-	***T***	36	-
H	15	-	***U***	41	-
I	15-18	-	V	41-53	-
J	15	-	***W***	47	R1, R2
K	22	-	***X***	51	-
L	22-48	R1, R2, R3	Y	15-50	R1
M	22-52	R2, R3	Z	20	-
			AA	31-56	R1, R2

The best solution found for this scenario is represented by the chromosome in Figure 17 which is the optimal solution for this scenario with an optimal value of $ 12100.

0	10	10	12	10	15	21	15	18	15	22	33	23

23	36	36	18	18	26	36	41	46	47	51	15	20	56

Fig. 17. Optimal solution found for the second scenario of the 27-activity project.

Figure 18 shows the Gantt chart for this solution. Again, the dummy activities are not included.

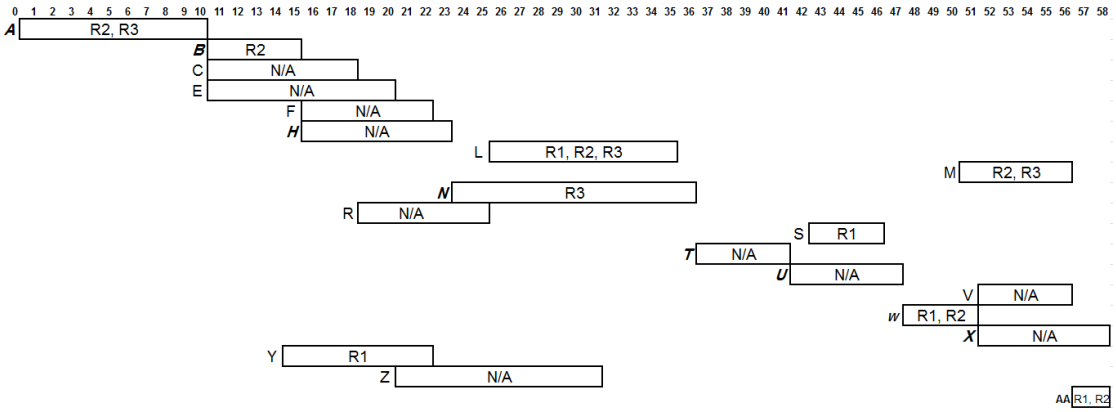

Fig. 18. Gantt chart for the second scenario of the 27-activity project

This Gantt chart is different than the one in Figure 16. The changes in the start times come to reduce the overall number of units of resources needed. As a consequence of this change, the optimal value was dropped from $33300 to $12100. Table 9 summarizes the differences in activities start times between the two charts.

Table IX
The start times for the different activities between first scenario and second scenario of the 27-activity project

Activity	First scenario start time	Second scenario start time	Start Range
L	33	33	22-48
M	23	23	22-52
S	26	26	25-54
V	46	46	41-53
Y	43	15	15-50
AA	54	56	31-56

From the different scenarios discussed, one can see that the model is effective in choosing the proper start times for the non-critical activities to avoid the overlap between the activities that demands the same resources when possible. When there must be an overlap, the model chooses the cheapest overlap. This shows that the proposed model is effective in scheduling the non-critical activities with respect to their resources costs.

CONCLUSIONS

In this paper a hierarchal model for RCPS problem is presented. The effectiveness of the model was demonstrated using two illustrative examples with six different scenarios. The results of these examples showed that the proposed model is effective in scheduling the non-critical activities within the minimum makespan of the project such that the total cost of resources needed to perform the project is minimized. Moreover, the GA was able to detect the optimal solutions considering all possible scenarios which in turn demonstrated the overall effectiveness of the proposed model. The heavy-mutation operator used in the GA was proven to be very efficient in producing feasible offspring by eliminating the need for chromosome-correction step. The proposed model can be used in projects that are very sensitive to the total duration of the project such as bridge construction projects or any heavy civil engineering construction projects.

ACKNOWLEDGMENT

The author would like to acknowledge the University of Petra for their all kinds of support.

REFERENCES

[1] Blazewicz, J., Lenstra, J., and Rinnooy Kan, A. (1983). Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics 5, 11-24.
[2] Kelley, J. (1963). The critical-path method: Resources planning and scheduling. In Industrial scheduling, J. Muth and G. Thompson, Eds. Prentice-Hall, New Jersey, pp. 347-365.
[3] Dorndorf U, Pesch E, and Phan-Huy T. (2000). A branch-and-bound algorithm for the resource constrained project scheduling problem, Mathematical Methods of Operations Research 52, 413-439.
[4] Brucker P, Knust S, Schoo A, and Thiele O., (1998). A branch & bound algorithm for the resource constrained project scheduling problem. European Journal of Operational Research 107(2), 272–88.
[5] Battersby and J Carruthers (1966). Advances in critical path methods. Operational Research Quarterly, 17, 4.
[6] Patterson J.H., and Huber W.D., (1974). A horizon-varying, zero-one approach to project scheduling. Management Science 20, 990–8.

[7] Patterson J.H., and Roth G.W., (1976) . Scheduling a project under multiple resource constraints: a zero one programming approach." AIIE Transactions 8, 449–55.
[8] Bartschi M. ; (1996); "A Genetic Algorithm for Resource-Constrained Scheduling" ; Master Thesis; Massachusetts Institute of Technology; Recovered from: http://lancet.mit.edu/ ~mwall/phd /thesis /thesis.pdf.
[9] Hartmann S. (1998). A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. Recovered from: http://halfrunt.bwl.unikiel. de/bwlinstitute/ Prod/mab/hartmann/ ga_sm4.pdf.
[10] Merkle D. , Middendorf M. , Schmeck H. (2002) "Ant Colony Optimization for Resource-Constrained Project Scheduling"; Recovered from: pacosy.informatik.uni-leipzig.de/pv/Personen/ middendorf / Papers/RCPSPGECCO.ps
[11] Zhi-Jie Chen, Chiuh-Cheng Chyu (2012). Solving the Resource Constrained Project Scheduling Problem to Minimize the Financial Failure Risk. International Journal of Advanced Research in Artificial Intelligence, Vol. 1, No. 1.
[12] Chyu C., and Chen J., (2009). Scheduling jobs under constant period-by-period resource availability to maximize project profit at a due date", Int. J. Adv. Manuf. Technol., vol. 42, pp. 569-580.
[13] Lova, A., Tormos, P., Cervantes, M, and Barber, F.(2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. International Journal of Production Economics 117 (2) 302 -316.
[14] Selcuk Colak, (2013).Multi-Mode Resource-Constrained Project-Scheduling Problem With Renewable Resources: New Solution Approaches. Journal of Business & Economics Research. Volume 11, Number 1.
[15] Jozefowska, J., Mika, M., Rozycki, R., Waligora, G. and Weglarz , J. (2001). Simulated annealing for multi-mode resource-constrained project scheduling. Annals of Operational Research,102(1):137–155.
[16] Hamdy A. Taha (2007). Operations research: an introduction. Eighth Edition, Prentice Hall, Upper Saddle River, N.J.