A Resource Leveling Model Based On Genetic Algorithms: Activity Splitting Allowed

N. Razavi, N. Mozayani

Abstract— **The goal of resource leveling is to minimize the deviation between the resource requirements and the desired resource profile to prevent problems that caused by fluctuation of required resources. It is a common problem and arises frequently in real situations and therefore has been studied numerous times. Almost in all of these studies and the resulting solutions, there exist a common element, which is once an activity is started, it cannot be stopped and restarted again. That is, it cannot be split. In many instances in actual construction, there exist activities that can be split. So, it seems to be very useful to develop a model for resource leveling problems that allows certain activities to be split. An important interesting property that such a model should have is its simplicity. By simplicity we mean its ease of use by inexpert users without strong mathematical background. This paper presents a model based on genetic algorithms to level resources that permits selected activities to stop and restart. This splitting of activities results in improvement to the leveling solution that is traditionally achieved when splitting is not allowed. Examples are presented that illustrate the improvement in the solution obtained from the proposed model compared to models that do not allow splitting and compares the result to that obtained using commercially available software. The model presented here is very efficient and effective. Therefore, it can be used in real situations to automatically solve resource leveling problems efficiently to obtain effective results.**

Index Terms— **activity splitting, genetic algorithms, resource leveling.**

I. Introduction

The resource leveling problem comes from the project scenario in which the project duration is fixed. Most construction projects have a completion date, specified in the contract documents, which determines fixed duration for the project. If the project is not completed by that date, the owner may incur damages due to the non availability of the facility. Construction contracts frequently contain a clause that the contractor pays a penalty for each day of delay after the stated completion date. To meet this completion date, the contractor has to manage resources efficiently. He/she would like to reduce the peek demand and the fluctuation of the resource required. Therefore the objective of this problem is to minimize the peek demand and fluctuations in the pattern of resource usage. The peek demand and fluctuations of resource are undesirable for the contractor because they can cause the following problems [1]-[3]:

- It is expensive to hire and fire labor on short-term basis to satisfy fluctuating resource requirements.
- Resource can not be managed efficiently, if the schedule demands more output per day than possible with available resource.

The resource leveling problem can be defined as combinatorial nondeterministic polynomial complete (NP-complete) problem [4]. The growth of computational time of NP-complete problem is usually in the order of exp(n). Thus, the computational time grows exponentially as the size of the problem increases. Also, there is no polynomial bounded algorithm for an NP-complete problem.

Little research has been done to solve the resource leveling problem using numerical optimization methods [5],[6]. This is because the models are only suitable for small networks due to the nature of the combinatorial problem described earlier.

Also, numerous heuristic methods have been developed for resource leveling [1], [3], [7], [8]. The basic idea of these methods is as follows:

- Create a resource profile based on the early start position of activity calculated from the critical path method (CPM).
- Shift non critical activities according to proposed heuristic rules.

However, also the heuristic methods can handle very large projects, solution they provide still needs improvements in the areas of efficiency and optimality. Therefore, it seems to be very promising to use genetic algorithms to solve resource leveling problem. Although, some researchers have developed some methods using GA [9]-[11], but nearly all existing methods have been developed with one common assumption, which is that an activity once started will continue until it is finished.

Although this is not a necessary constraint for all activities, it is valid in many cases. The assumption is convincing because of the setup time and cost for restarting an activity. However, all the activities do not need the setup time and added cost to split the activity. There is one work in which some activities can be split, but it uses linear programming method [12]. As mentioned earlier, this method suffers from the large problem size. Some commercial project planning software's have a "splitting" function for resource leveling [13]. This splitting function allows an activity to be stopped and restarted. However, not all programs have this splitting function [14]. Additionally, the heuristic rules used to level resources

N. Razavi is PH.D student in Iran University of Science and Technology, Tehran, Iran (phone: +984113309664; e-mail: Razavi@iust.ac.ir).

N. Mozayani is with the Iran University of Science and Technology, Tehran, Iran (e-mail: mozayani@iust.ac.ir).

incorporated into the software do not guarantee an optimal solution.

In this paper we have developed a model to solve resource leveling problem based on genetic algorithms that allows some non critical activities to be split.

II. Overview of Resource Leveling Problem

Popescu and Charoenngam describe resource leveling as "the method of scheduling activities within their available float so as to minimize fluctuation in day-to-day resource requirements." [15]. An early start schedule of the project tends to create conflicts by requiring large number of resources on some days of project. An additional problem that can occur with an early start schedule is fluctuating requirements during the project. To fix this undesirable situation, the resource leveling method was introduced [7]. Noncritical activities are split within their available float to minimize the deviation between the desired daily resource requirement and the actual daily resource requirement.

III. Genetic Algorithms

A. Overview

Genetic algorithms were developed to simulate the genetic evolution process: *survival of the fittest*. The evolution process predicts the survival and characteristics of the offsprings on the basis of knowing the characteristics of their parents. A GA is an optimization procedure that operates on set of design variables. Each set is called a string and it defines a potential. Each string consists of a series of characters representing the value of the discrete design variables as defined by the objective function and the constraints. In its simplest form, a genetic algorithm consists of three operations: (1) reproduction, (2) crossover, and (3) mutation. Each of these operations are described below.

The reproduction operation is the basic engine of *Darwinian natural selection* by the survival of the fittest. The goal of the reproduction process is the information stored in string with good fitness value to survive into the next generation. Each string in the population is assigned a probability of being selected as a parent string based on the string's fitness. As such, reproduction does not change the features of parent strings. The next generation of the offspring's strings is developed from selected pairs of parent strings when exposed to the application of the explorative operators such as crossover and mutation.

Crossover is a procedure in which a selected parent string is broken into segments and some of these segments are exchanged with corresponding segments of another parent string. In this manner, the crossover operation creates variations in the solutions population by producing new solution strings that consist of parts taken from the selected parent strings.

The mutation operation is introduced as an insurance policy to enforce diversity in a population. It introduces random changes in the solution population by exploring the possibility of creating and passing features that are nonexistent in both parent strings to the offsprings. Without an operator of this type, some possible important regions of the search space may never be explored.

There are five steps to creating a GA: (1) formation of the chromosome structure suitable for the problem on hand, (2) selection of the evaluation criteria (objective function), (3) generation of an initial population of chromosomes (initial population), (4) selection of an offspring generation mechanism (process to generate new potential solutions), and (5) preparation of the procedure code to apply genetic operators to generate the next generation of solution strings.

Genetic algorithms are considered one of the more effective techniques for determining optimal solution especially when the problem domain is fairly large. As algorithms, they are different from traditional optimization methods in the following aspects:

1) Genetic algorithms operate on a coding set of variables and not with variables themselves;
2) They search for a population of solutions rather than improving a single solution;
3) They use objective function without any gradient information; and
4) Their transition scheme is probabilistic, whereas traditional methods use gradient information .

The genetic algorithm system can also be described by the following pseudocode:

BEGIN
INITIALIZE population with random candidate solutions;
EVALUATE each candidate;
REAPEAT UNTIL (*TERMINATION CONDITION* is satisfied) DO
1 *SELECT* parents;
2 *RECOMBINE* pairs of parents;
3 *MUTATE* the resulting offspring;
4 *EVALUATE* new candidates;
5 *SELECT* individuals for the next generation;
OD
END

In the following sections we will describe each component of the pseudocode in our model with more details.

B. Chromosomes Structure in our GA Model

The activities in a Critical Path Method (CPM) project network can be identified as critical activities or noncritical activities. Each noncritical activity k has duration D_k, the earliest start time ES_k, the earliest finish time EF_k, total float TF_k, free float FF_k and daily resource requirement R_k (k = 1, 2, …, n), where n is the number of noncritical activities. The critical activities are not shifted in the traditional resource leveling model because the project duration is fixed. Thus, noncritical activities are split within the extent of the activities' float.

Thus, in our chromosomes there is one gene for each activity that can be split. Each gene represents the possible time units that the corresponding activity can be done. The length of the gene corresponding to activity k is equal to $D_k + TF_k$.

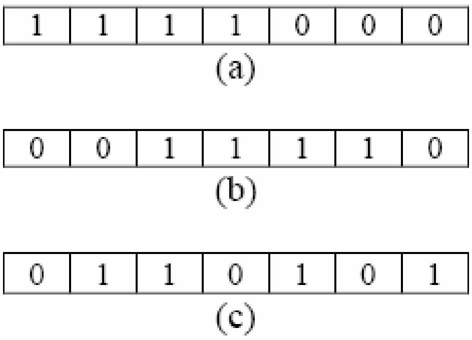


Fig. 1. Three possible scheduling for an activity. (a) Scheduling the activity based on early start. (b) The activity is shifted two units of time. (c) Splitting the activity.

For example, suppose we have an activity A that has 4 days of duration and 3 days of total float and it's early start time is equal to 5. In Fig. 1, the corresponding gene for activity A is represented. Fig. 1(a) states that A is done on days 5, 6, 7, and 8; and there is no shifting or splitting. In Fig. 1(b), A is shifted two days and thus it is done on days 7, 8, 9, 10. As can be seen in this figure there is no splitting. But in Fig. 1(c), A has one day of delay and two points of split (on days 8, 10); and so it is done on days 6, 7, 9, 11.

There is one constraint that should be satisfied in each gene: The number of 1's in each gene should be equal to the duration of its corresponding activity. Another constraint that should be considered in each chromosome is that in a network diagram the relationships of the activities can not change. In order to maintain this, a relationship constraint is needed. This relationship constraint says that there should be no overlap between an activity and its predecessors.

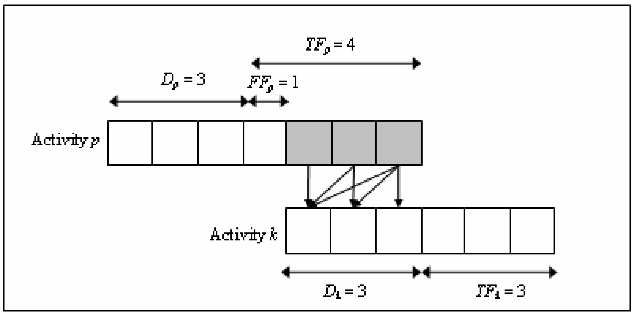


Fig 2. Example for relationship constraint

Consider the simple example shown in Fig. 2. A noncritical activity k has a predecessor activity p. each activity's duration is 3 days. FF_p is 1 day, TF_p is 4 days, and TF_k is 3 days. According to this figure, if each of the last three days in activity p (shaded in the figure) has value 1, then all of its corresponding days (shown with arrows) in activity k must have value 0.

C. Initialization

In our GA, each chromosome in the initial population is created randomly so that each gene constraint and relationship constraints have been satisfied.

D. Evaluating Chromosomes

The objective function for the resource leveling model is to minimize the deviation between the actual daily resource requirement y_i and the desired daily resource requirement d_i on day i. the minimization of the sum of the squared deviation between y_i and d_i is used as the objective function for our GA model. The objective function can be expressed as equation (1):

$$\text{Minimize } z = \sum_{i=1}^{T} (d_i - y_i)^2 \qquad (1)$$

Where T = project duration.

Here, a uniform resource level is chosen without loss of generality and is given by:

$$d_i = \frac{\left(\sum_{i=1}^{T} y_i\right)}{T}$$

E. Selecting Parents

In our GA, we have used tournament selection to select parents. In this method, at first we select k (an adjustable parameter to control selection pressure) chromosomes at random out of population with n chromosomes ($k \leq n$), and then we select the best chromosome (the chromosome with the lowest fitness value) among them. This process will be repeated n times to fill the intermediate population.

F. Recombination

In this operation two offsprings will be created by exchanging some possible genes between two selected parents. Stating from the first gene in chromosomes, for each gene say i, the possibility of exchanging the value of that gene in the parents is checked according to the relationship constraints and if possible the values of the gene i will be exchanged in the two offsprings. Otherwise the value of gene i in each parent will be copied directly in the offsprings. This process will be repeated for each gene separately.

G. Mutation

The mutation operator works on the new offsprings (after crossover) in a simple manner. For each gene, the activity that corresponds to that gene will be rescheduled separately with probability p_m so that no constraints will be violated. Rescheduling an activity is based on its predecessors and its successors. At first, according to the predecessors, the actual early start time of the activity is computed and then according to the successors, the actual latest finish time of the activity is computed. Then the activity is rescheduled within this extent randomly.

H. Survival Selction

Our GA model follows the simple Generational GA model in which all the chromosomes in the current population will be replaced with the new chromosomes (offsprings) in the mating pool. Also, we have used elitism in which some of the best chromosomes in the current population will be copied directly into the next population without undergoing crossover or mutation operators.

I. Termination Condition

Our GA has a simple criterion for termination. In each run, it will be terminated after a prespecified number of generations.

IV. Experiments

A. Example 1

Consider the CPM diagram in Fig. 3, which is a precedence network diagram of a project consisting of ten activities. The network diagram shows duration, daily resource requirement, and relationship of each activity. Table I shows the relevant activity information of example 1. Fig. 4 shows the bar chart schedule based on early start schedule. The gray areas are the possible occupying positions of the activities. Activities A, B, C, D, E, and F are the critical activities and their positions are fixed in order to maintain the original project duration. The daily resource profile corresponding to the early start schedule is shown in Fig. 5.

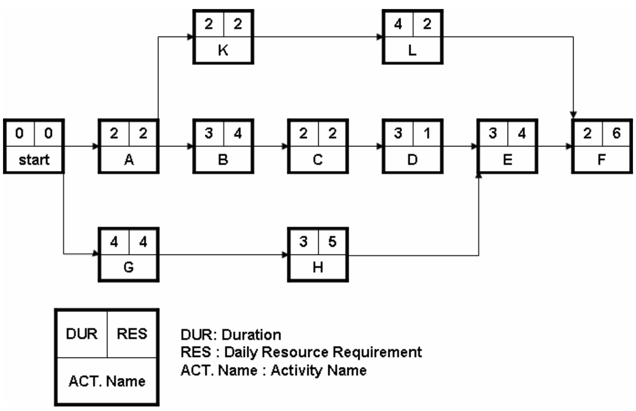


Fig. 3. Precedence diagram of example 1

Table I Schedule Data for Activities in Example 1

ACT	DUR	ES	EF	LS	LF	TF	FF
A	2	1	2	1	2	0	0
B	3	3	5	3	5	0	0
C	2	6	7	6	7	0	0
D	3	8	10	8	10	0	0
E	3	11	13	11	13	0	0
F	2	14	15	14	15	0	0
G	4	1	4	4	7	3	0
H	3	5	7	8	10	3	3
K	2	3	4	8	9	5	0
L	4	5	8	10	13	5	5

Note: ACT = activity; DUR = duration; ES = earliest start time ; EF = earliest finish time; TF=total float; and FF=free float

Activity	DUR	RES	Days														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	2	2	2	2													
B	3	4			4	4	4										
C	2	2						2	2								
D	3	1								1	1	1					
E	3	4											4	4	4		
F	2	6														6	6
G	4	4	4	4	4	4											
H	3	5					5	5	5								
K	2	2			2	2											
L	4	2					2	2	2	2							
y_i			6	6	10	10	11	9	9	3	1	1	4	4	4	6	6

Fig 4. Bar chart based on early start

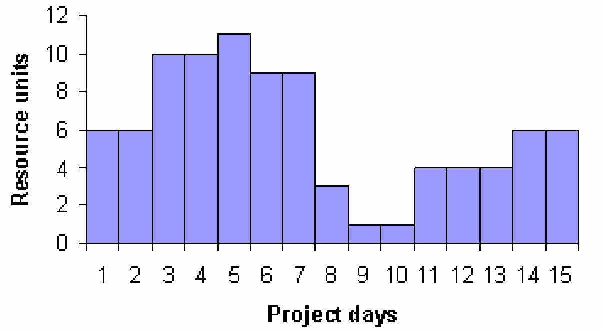


Fig. 5. Daily resource profile based on early start

The GA model is used to minimize the deviation between the desirable daily requirement and the actual daily requirement as discussed earlier. For the desired daily requirement, a uniform distribution is used. Thus, d_i is a constant that equals to the total resource rate divided by the project duration. The value of d_i is 90/15 = 6. The resulting bar chart schedule is shown in Fig. 6 and the daily resource profile corresponding to this scheduling is shown if Fig. 7. As can be seen in this figure, this scheduling is optimal in the way that there is no fluctuation at all. In Fig. 8, we have presented the result of scheduling with our model when splitting is not allowed to stress that splitting can produce better results in cases that some activities can be split without any extra cost due to splitting. In this case, the daily recourse profile is also presented for a better comparison in Fig. 9.

Activity	DUR	RES	Days														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	2	2	2	2													
B	3	4			4	4	4										
C	2	2						2	2								
D	3	1								1	1	1					
E	3	4											4	4	4		
F	2	6														6	6
G	4	4	4	4				4	4								
H	3	5								5	5	5					
K	2	2			2	2											
L	4	2					2						2	2	2		
y_i			6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Fig. 6. Bar chart after leveling (splitting allowed)

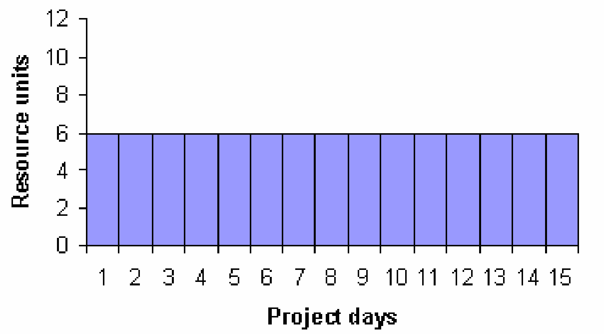
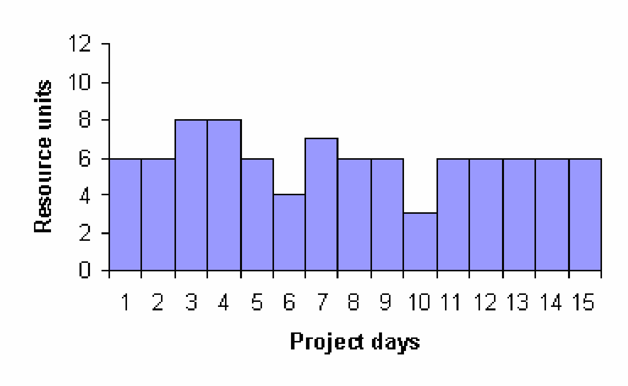


Fig 7. Daily resource profile after leveling (splitting allowed)

Activity	DUR	RES	Days 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	2	2	2	2													
B	3	4			4	4	4										
C	2	2						2	2								
D	3	1								1	1	1					
E	3	4											4	4	4		
F	2	6														6	6
G	4	4	4	4	4	4											
H	3	5							5	5	5						
K	2	2					2	2									
L	4	2										2	2	2	2		
y_i			6	6	8	8	6	4	7	6	6	3	6	6	6	6	6

Fig. 8. Bar chart after leveling (splitting not allowed)



Fg 9. Daily resource profile after leveling (splitting not allowed)

Our resource leveling model for the first example allows activity splitting. However, the user may want to apply activity splitting to specific activities instead of applying it to all the activities. Our model enables the user to do this easily and with no extra effort just by removing those specific activities from the list of the activities that can be split. As an example, we have provided the result for splitting when activity L cannot be split.

The bar chart schedule for the modified example is shown in Fig. 10. The daily resource profile corresponding to the modification is shown in Fig. 11. As can be seen, the value of the objective function is 4 which is not as good as when all activities ca be split, but is better than when no activities could be split. This is to be expected and is consistent with actual construction practices where only certain activities could be split.

Activity	DUR	RES	Days 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	2	2	2	2													
B	3	4			4	4	4										
C	2	2						2	2								
D	3	1								1	1	1					
E	3	4											4	4	4		
F	2	6														6	6
G	4	4	4	4				4	4								
H	3	5								5	5	5					
K	2	2			2	2											
L	4	2										2	2	2	2		
y_i			6	6	6	6	4	6	6	6	6	8	6	6	6	6	6

Fig 10. Bar chart after leveling (splitting not allowed for activity L)

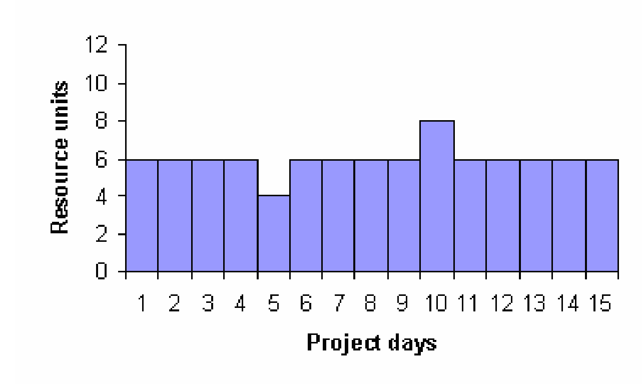


Fig. 11. Daily resource profile after leveling (splitting not allowed for activity L)

B. *Example 2*

Consider the project network shown in Fig 12. The project network has five critical activities (I, J, K, G, and H) and six noncritical activities (A, B, C, D, E, and F). Table II shows the relevant activity information of this example. Let us assume that activities A, C, D, and E can be stopped and restarted. That is, they can be split. Activities B and F cannot be split.

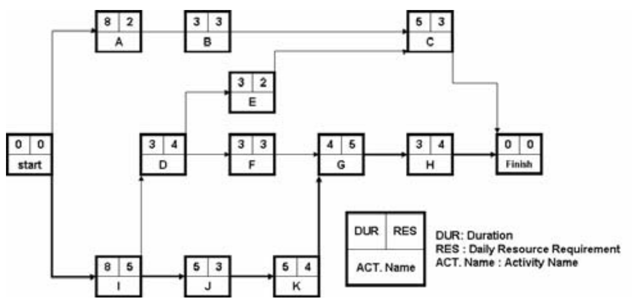


Fig 12. Precedence diagram for example 2

The bar chart schedule based on GA when splitting is not allowed is shown in Fig. 13 and the bar chart schedule based on GA when activities A, C, D, and E can be split is shown in Fig. 14. The resource profiles generated in both cases (with and without splitting) are superimposed in Fig. 15 to show the improvement. The objective function value from the GA when splitting is not allowed is 9. The objective function value from the GA when splitting is allowed is 3. As can be seen, there is some improvement when splitting is allowed.

Table II Schedule Data for Activities in Example 2

ACT	DUR	ES	EF	LS	LF	TF	FF
A	8	1	8	8	15	7	0
B	3	9	11	16	18	7	1
C	5	13	17	19	23	6	6
D	3	7	9	11	13	4	0
E	3	10	12	16	18	6	0
F	3	10	12	14	16	4	4
G	4	17	20	17	20	0	0
H	3	21	23	21	23	0	0
I	6	1	6	1	6	0	0
J	5	7	11	7	11	0	0
K	5	12	16	12	16	0	0

Note: ACT = activity; DUR = duration; ES = earliest start time ; EF = earliest finish time; TF=total float; and FF=free float

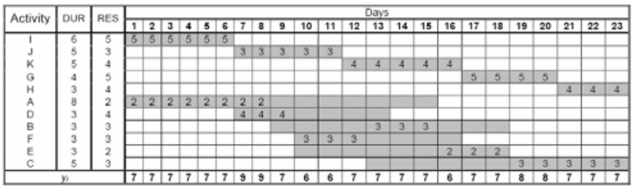


Fig. 13. Bar chart after leveling (splitting not allowed)

Fig. 14. Bar chart after leveling (splitting allowed to certain activities)

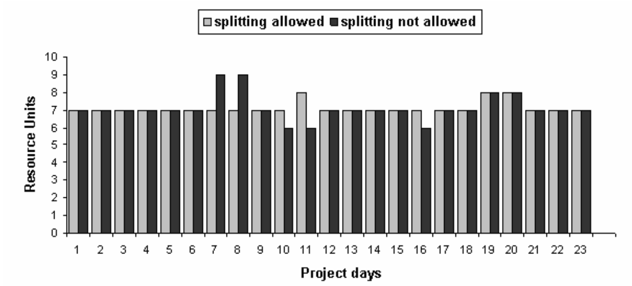


Fig. 15. Daily resource profiles

J. Son and K. Mattila have leveled this same network and associated resources using commercially available software. They have used SureTrak Project Manager version 3.0 and Primavera Project Planner (P3) version 3.0. We have used their report to compare the results of these programs with our model. Table III shows their report.

Table III SureTrak and P3's Objective Functions Values

Software	Resource limit	Objective function value
P3[a]	10	22.02
P3	8	22.27
P3	7	22.27
SureTrak[b]	10	22.02
SureTrak	8	14.35
SureTrak	7	27.99

[a]Primavera Systems Inc. (1999a).
[b]Primavera Systems Inc. (1999b).

The results are interesting in that the best result of 14.35 is obtained using SureTrak when the resource limit is 8. This occurs even though activities cannot be split in SureTrak. This compares to the objective function value of 3 when activities A, C, D, and E can be split and 9 when no activity can be split using the model presented. This improvement in the objective function when using an optimal solution is important for construction practitioners to be aware of.

V. Conclusion

This paper presented a GA model to level resources where activity splitting was allowed (Although it can be used when no activities can be split). Activity splitting in the traditional models is not permitted. By allowing activity splitting of certain activities the model can accurately represent the actual construction process.

The model is developed and tested on a CPM schedule and the results are compared to solutions obtained when activity splitting is not allowed. When activity splitting is allowed for all activities and when only certain activities can be split. As mentioned before, in this model the user can select which activities can be split and therefore this model is a more realistic approach to the resource leveling problem.

Additionally, the results using the model presented where compared to that using commercially available software. The model results were better than commercial software when using the default settings.

In summery, the model for resource leveling presented in this paper shows the applicability of GA to this problem for the first time, and also the results are very promising. However, there are some improvements that can be done on our model like using local search algorithms (hill climbing and simulated annealing) to fine tune the resulting solutions. Another possibility that should be addressed is to modify the model to handle construction projects that use multiple resources. Also, this model still needs to be tested on larger problem sizes to gain a better understanding of its performance.

References

[1] Harris, R. B. *Precedence and arrow networking techniques for construction*. Wiley, New York, 1978.

[2] Stevens, J. D. *Techniques for construction network scheduling*. McGraw-Hill, New York, 1990.

[3] Martinez, J., and Ionnou, P. "Resource leveling based on the modified minimum moment heuristic." *Proc, 5th Int. Conf., Computing in Civ. and Build. Engrg*. ASCE, Reston, Va., 1993, pp. 287-294.

[4] Savin, D., Alkass, S., and Fazio, P. "Construction resource leveling using neural networks". *Can. J. Civ. Engrg*., 23(3), 1996, pp. 917-925.

[5] Easa, S. M. "Resource leveling in construction by optimization." *J. Constr. Engrg. and Mgmt*. ASCE, 115(2), 302-316.

[6] Bandelloni, M., Tucci, M., and Rinaldi, R. (1994). "Optimal resource leveling using non-serial dynamic programming." *Eur: J. Operation Res*., 78(2), 1989, pp. 162-177.

[7] Burgess, A. R., and Killebrew, J., B. "Variation in activity level on a cyclic arrow diagram." *J. Industrial Engrg*., 13(2), 1962, pp. 76-83.

[8] Levy, F. K., Thompson, G. L., and Wiest, J. D. (1962). "Multiship, multishop, workload smoothing program." *Naval Res. Logistics Quarterly*, 9(1), 1962, pp. 37-44.

[9] Hegazy, T. "Optimization of Resource Allocation and Leveling Using Genetic Algorithms". *J. Constr. Engrg. and Mgmt*. ASCE, 125(3), 1999, pp. 167-175.

[10] Chan, W., and Chua, D. "Construction Resource Scheduling with Genetic Algorithms". *J. constr. Engrg. and Mgmt*. ASCE, 122(2), 1996, pp.125-132.

[11] Senouci, A., and Eldin, N. "Use of Genetic Algorithms in Resource Scheduling of Construction Projects". *J. Constr. Engrg. and Mgmt*. ASCE, 130(6), 2004, pp. 869-877.

[12] Son, J., and Matilla, K. G. "Binary resource leveling model: activity splitting allowed". *J. Constr. Engrg. and Mgmt*. ASCE, 2004, pp. 887-994.

[13] Primavera Systems, Inc. *Primavera project planner 3.0 reference manual*, Primavera Systems, Inc., Bala Cyynwyd, Pa, 1999a.

[14] Primavera Systems, Inc. *SureTrak 3.0 project manager user's guide*, Primavera Systems, Inc., Bala Cyynwyd, Pa, 1999b.

[15] Popescu, C. M., and Charoenngam, C. *Project planning, scheduling, and control in construction*. Wiley, New York, 1995.