SIMULATED ANNEALING FOR LINEAR SCHEDULING PROJECTS WITH MULTIPLE RESOURCE CONSTRAINTS

C.I. Yen[1]

[1] Graduate Institute of Construction Engineering and Management, National Central University, No. 300, Jhongda Rd., Jhongli, Taiwan 32001, R.O.C.

Abstract

Many construction projects such as highways, pipelines, tunnels, and high-rise buildings typically contain repetitive activities. Research has shown that the Critical Path Method (CPM) is not efficient in scheduling linear construction projects that involve repetitive tasks. Linear Scheduling Method (LSM) is one of the techniques that have been developed since 1960s to handle projects with repetitive characteristics. Although LSM has been regarded as a technique that provides significant advantages over CPM in linear construction projects, it has been mainly viewed as a graphical complement to the CPM. Studies of scheduling linear construction projects with resource consideration are rare, especially with multiple resource constraints.

The objective of this proposed research is to explore a resource assignment mechanism, which assigns multiple critical resources to all activities to minimize the project duration while satisfying the activities precedence relationship and resource limitations. Resources assigned to an activity are allowed to vary within a range at different stations, which is a combinatorial optimization problem in nature. A heuristic multiple resource allocation algorithm is explored to obtain a feasible initial solution. The Simulated Annealing search algorithm is then utilized to improve the initial solution for obtaining near-optimum solutions. A housing example is studied to demonstrate the resource assignment mechanism.

Keywords: Linear Scheduling, Simulated Annealing, Resource Allocation, Resource Leveling, Heuristic Optimization.

Symbols

$s(n, m)$	start time of activity n at location m
$f(n, m)$	finish time of activity n at location m
$dp_{t,i}$	t,i absolute difference plus value of resource i between day t + 1 and day t
$dm_{t,i}$	t,i absolute difference minus value of resource i between day t + 1 and day t
w_i	i weighting factor for resource i
$L(n,p)$	lead time between activity n and its predecessor p
$r_i(n, m, t)$	i resource i assigned to activity n at location m at time t
$RA_i(t)$	i resource i availability at time t
$TR_i(n, m)$	i total amount of resource i required to complete activity n at location m

1. Introduction

Many construction projects, such as highway construction, pipelines, tunnels, and high-rise buildings, typically contain activities repeated continuously at different locations. Research has shown that CPM is not efficient enough in scheduling linear construction projects with repetitive activities. Many studies of scheduling for construction projects with repetitive tasks have been undertaken, such as the Line of Balance (LOB) and the Linear Scheduling Method (LSM). Project scheduling techniques such as network methods (CPM/PERT) have been thoroughly developed. Resource allocation and resource leveling techniques have also been explored and developed to incorporate with traditional network methods. However, only a few research considering multiple resources allocation and resource leveling for linear scheduling project are done. Resource leveling and allocation problems are combinatorial optimization problem by nature. Heuristic searching techniques are commonly utilized for solutions. A two-stage heuristic algorithm utilizing the Simulated Annealing is developed in this research to solve linear scheduling problems with multiple resources constrained. The two main techniques are discussed below.

1.1 Linear Scheduling Method

A typical Linear Schedule diagram is a chart with time on one axis and location on the other axis. Activities are represented by line segments, while the slope of the lines represents the production rate. Johnston (1981) [1] introduced the LSM to a highway construction project. Johnston also compared the LOB and LSM and stated that, in several ways, the LSM diagram resembles the objective chart of the LOB technique. Harmelink (1998) [2] developed the Linear Scheduling Model (LSM), which implants the concepts of CPM into linear scheduling. The model defined controlling activities and path, which are similar to critical activities and path for the CPM method. The controlling path determines the duration of a project and should not be delayed. Harmelink[2] also developed a heuristic algorithm to determine the controlling activities path in a linear project and also compared CPM and LSM by scheduling two projects using both methods[3].

1.2 Simulated Annealing

Simulated Annealing (SA), also known as Monte Carlo annealing, statistical cooling, probabilistic hill-climbing, and stochastic relaxation, is based on an analogy taken from he physical annealing process of obtaining a crystalline structure. In 1983, three IBM researchers published a paper called ”Optimization by Simulated Annealing” in the Science magazine [4]. It was the first paper to introduce Simulated Annealing to the optimization world. SA has been widely used in many research areas, such as chemical engineering and electrical engineering, as a search technique for solving optimization problems.

Compared with the traditional Local Search, SA randomly chooses one point from the search neighborhood and evaluates, while Local Search evaluates the entire search neighborhood and chooses the best one to which to move. In addition, SA will move to a worse solution based on an acceptance probability, while Local Search will never move to a worse solution. Because of the possible acceptance of a worse solution, it allows

Simulated Annealing an opportunity to escape from being trapped in a local optimum (as Local Search always does) and a better chance to find a better solution. Figure 1 depicts the basic concepts behind the Simulated Annealing algorithm. The following steps explain how the algorithm searches within a solution space:

1. Given an initial solution, which is a point in the solution space, and a high initial temperature T.
2. Identify the search neighborhood, which is a set of points that can be reached in one move from the current point.
3. Randomly choose one point from the search neighborhood and evaluate the objective value. If the new point has a better objective value than the current point, the algorithm accepts this point and move on. If the new point is worse than the current point, the algorithm only accepts it based on an acceptance probability $e^{eval(V\,c)-eval(V\,n)/T}$, which will be explained in next subsection.
4. Continue repeating the previous step under the same temperature until a termination condition is met.
5. Lower the temperature and repeat the previous 3 steps.
6. Repeat lowering the temperature again and again until a halting criterion is met.

Procedure Simulated Annealing (Minimization problem)

```
Begin
   t := 0
   Initialize T
   Select a current point Vc at random
   Evaluate Vc
  Repeat
     Repeat
        Select a new point Vn in the neighborhood of Vc
        If eval(Vn) < eval (Vc)
           Then Vc ← Vn
           Else  If random [0,1) < e^(eval(Vc)-eval(Vn)/T)
                    Then Vc ← Vn
     Until (termination-condition)
     T ← g(T,t)
     t ← t + 1
   Until (halting-criterion)
End
```

Figure 1: Simulated Annealing Algorithm (Michalewicz & Fogel[5])

2. Multiple Resource-Constrained Linear Scheduling Problems

Assume a construction project contains N repetitive activities and repeats at M different locations and that I different critical resources will affect the project schedule. The problem is to determine the resource assignments of all resources for all activities at all locations. With the resource assignments determined, the duration of each activity can be calculated and thereby determine the entire project duration. The problem can be formulated as follows:

Objectives :

$$Min\ Max\{f(n,m)|n = 1..n; m = 1..M\} \qquad \text{Eqn. (1)}$$

$$Min \sum_{t=1}^{T-1} \sum_{i=1}^{I} w_i (dp_{t,i} + dm_{t,i}) \qquad \text{Eqn. (2)}$$

Constraints :

$$s(n,m+1) \geq f(n,m) \quad \forall n = 1..N; \forall m = 1..M \qquad \text{Eqn. (3)}$$

$$s(n,m) \geq f(p,m) + L(n,p) \quad \forall m = 1..M; \forall p \in P(FS) \qquad \text{Eqn. (4)}$$

$$s(n,m) \geq s(p,m) + L(n,p) \quad \forall m = 1..M; \forall p \in P(SS) \qquad \text{Eqn. (5)}$$

$$f(n,m) \geq s(p,m) + L(n,p) \quad \forall m = 1..M; \forall p \in P(SF) \qquad \text{Eqn. (6)}$$

$$f(n,m) \geq f(p,m) + L(n,p) \quad \forall m = 1..M; \forall p \in P(FF) \qquad \text{Eqn. (7)}$$

$$\sum_{m=1}^{M} \sum_{n=1}^{N} r_i(n,m,t) \leq RA_i(t) \quad \forall t = 1..T; \forall m = 1..M; \forall i = 1..I \qquad \text{Eqn. (8)}$$

$$\sum_{t=1}^{T} r_i(n,m,t) \geq TR_i(n,m) \quad \forall n = 1..N; \forall m = 1..M; \forall i = 1..I \qquad \text{Eqn. (9)}$$

$$\sum_{n=1}^{N} \sum_{m=1}^{M} [r_i(n,m,t+1) - r_i(n,m,t)] - dp_{t,i} + dn_{t,i} = 0 \quad \forall t = 1..T-1; i = 1..I \qquad \text{Eqn. (10)}$$

There are two objective functions for this problem. The first one Eqn.(1) is to minimize the project duration. The second objective function Eqn.(2) is to minimize the daily resource usage fluctuation, which can be achieved by minimizing the total sum of the absolute resource usage change between any two consecutive days for the entire project time span. The model needs to satisfy four sets of constraints. Eqn.(3) ~ Eqn.(7) constrain the activities precedence relationships. Eqn.(8) ensures that at any given day t within the project duration, the summation of resource usage assigned to all activities does not exceed the resource availability at time t. Eqn.(9) guarantees that any activity n at location m, the total resource allocated to this activity across time needs to be greater than or equal to the total resource amount (e.g., man-hour) required to complete this activity. Eqn.(10) ensures the balance of resource usage deviation for any given day t.

3. Solution Procedure

The proposed problem is focused on assigning resources to activities at different locations, which naturally is a combinatorial optimization problem. Combinatorial optimization problems can guarantee a global optimum solution by evaluating all possible solutions. However, the total number of evaluations can grow exponentially and sometimes are not feasible in real world practices. Therefore, it is very common using a heuristic algorithm to find a close-to-optimum solution for a combinatorial optimization problem.

Liu (1999) [6] developed a two-stage structure for solving a linear scheduling problem with single resource constraint. The same structure will be utilized in the research for a multi-resource model. Stage one utilizes a heuristic multiple resources allocation algorithm to obtains a initial feasible solution of the proposed problem. Stage two then uses the SA search technique to improve the initial feasible solution to find a close-to-optimum one.

3.1 Multiple Resources Allocation Algorithm

Before explaining the details of the multiple resources allocation algorithm, an important concept, called ”The Next Time Frame (NTF),” needs to be discussed. Resource assignments may change regularly over the entire project life span. The idea of the NTF is to identify a time frame, within which the resource assignments can remain unchanged. Activities that are allowed to work within this time frame are also identified and assigned with resources. The NTF was introduced by Moselhi & Lorterapong (1993) [7].

The multiple resource allocation algorithm flowchart is depicted in Figure 2. *As* represents a list of activities that can be started at time T, while *Ap* is the list of activities that are already in progress at time T. The algorithm starts by reading the activity information. The algorithm then examines the completion of all activities. If not all activities are completed, the algorithm will identify a potential activity list *As* and evaluate whether the current available resources are capable of starting all activities in the *As*. All activities in the *As* will start and include to *As* if there is enough resources, otherwise, the resources will be assigned to activities based on the activity priority. After resource assignments are completed, the finish times of activities in *As* and *Ap* can be determined. Next timeframe (NT) can be identified. CT can then switch to the NT and go back to start the next iteration by examining the completion of all activities. If all activities are completed, the scheduling process is completed. The project duration can be identified by finding the minimum finish time of all activities.

3.2 Simulated Annealing for Linear Scheduling

Figure 3 demonstrates the flowchart of the stage-two SA process incorporated with the multiple resource allocation algorithm. The algorithm begins with inputting an initial schedule from the result of stage one and the cooling parameter T. The algorithm then defines the searching neighborhood of the current solution and randomly picks one neighbor from it to evaluate the objective functions, which are to minimize the project duration and resource usage fluctuation. If the chosen neighbor results in a better project duration or less usage fluctuation, the search moves to the chosen point and replaces it as the current point. If the chosen neighbor point generates a worse solution, the search will move to it as a new point based on an acceptance probability.

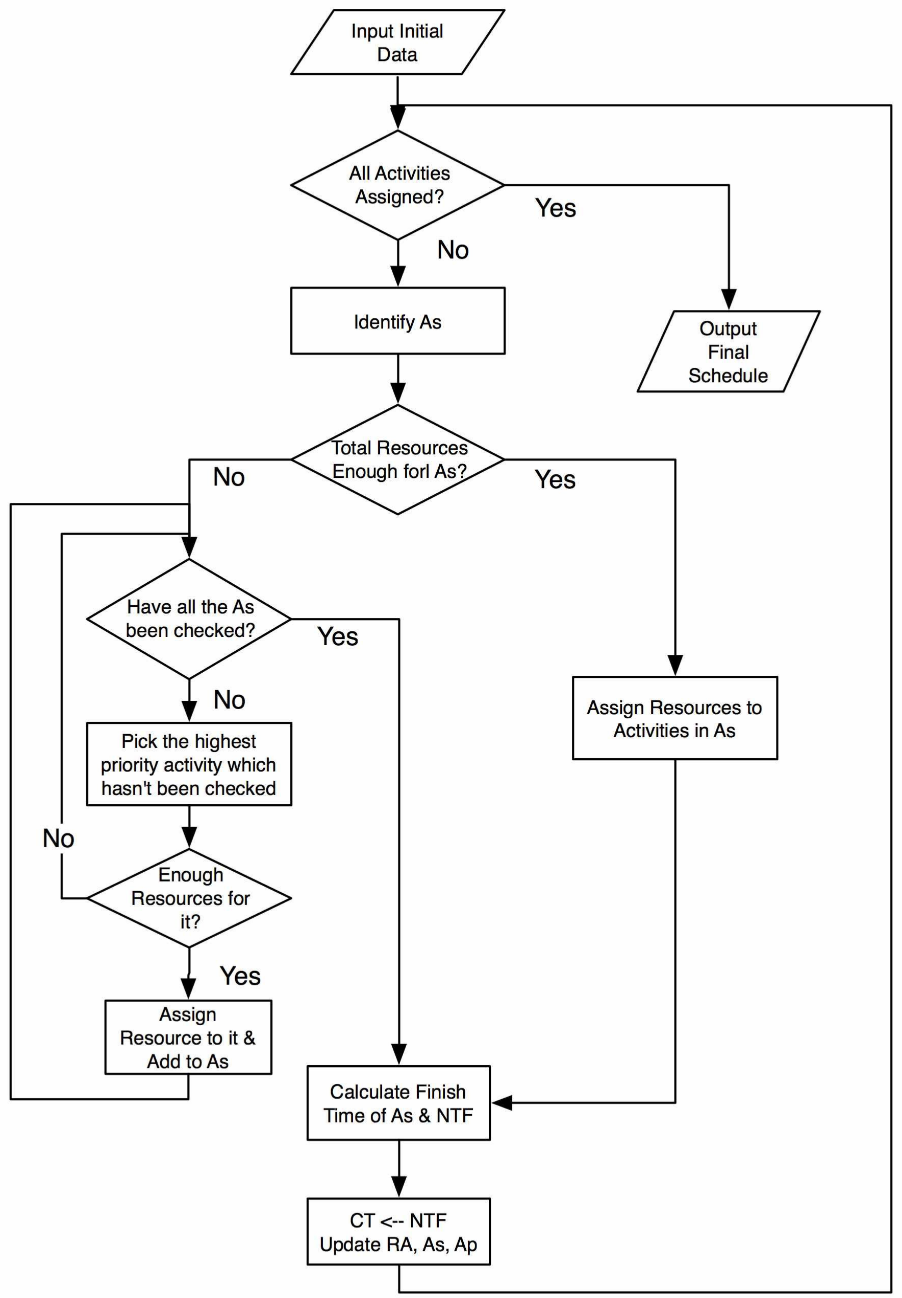

Figure 2: Multiple Resource Allocation Algorithm

The algorithm then generates a random number between 0 and 1. If the randomly generated number is greater than the acceptance probability, the chosen neighbor point will be accepted as the new current point. Otherwise, the algorithm discards this point and goes back to choose another neighbor point for evaluation. Under the same temperature T, the algorithm will repeatedly evaluate and move to new neighbor points until a specific termination condition is met, which can be as simple as a fixed number of iterations. After the termination condition is met, the algorithm decreases the temperature T and starts a new round of evaluation as described above. The algorithm will continue the same process and keep lowering temperature T until the halting criterion is met.

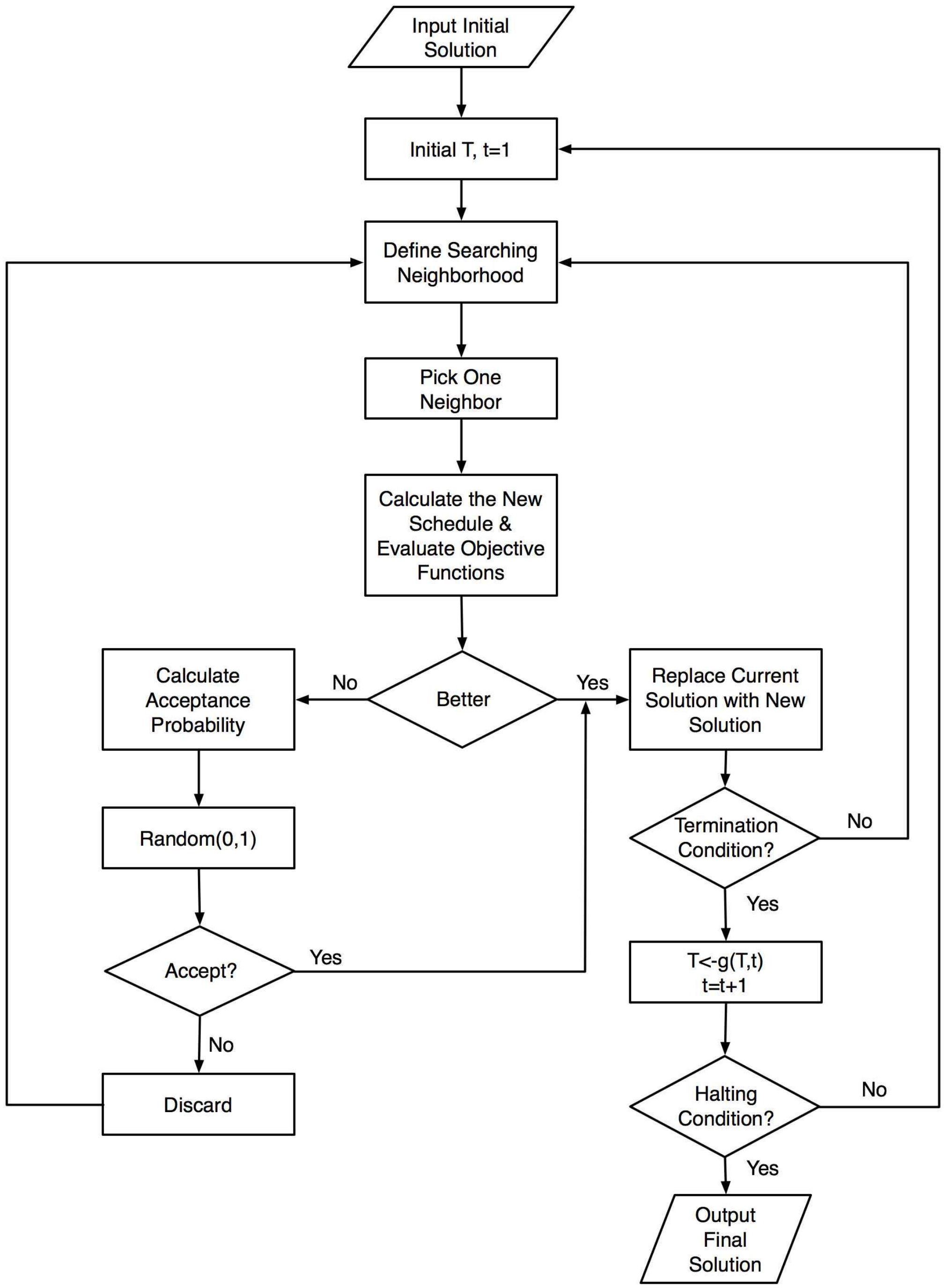

Figure 3: Simulated Annealing Incorporates with Multiple Resource Allocation Algorithm

4. Example: Housing Project

Assume a housing project consists of four activities: foundation (Activity A), ground-floor walls (Activity B), floor slab (Activity C), and first-floor walls (Activity D). A total of five housing units will be constructed in this project and all activities are identical for each unit.

Table 1 lists the activity ID, the duration, and the precedent relationship with other activities. For example, Activity B can start only after Activity A is finished for one day. Two different type of labors are assumed to be required in the housing project. The daily availability of Labor 1 is 8unit/day Table 2 lists the resource-related data for each activity. The ”Initial Labor 1 Assignment” and ”Initial Labor 2 Assignment” columns represent the daily labor units that are assigned to each activity. The number insides the parenthesis is the total labor units required to complete the activities. The ”Priority” column shows the priority of the activity, which will be used in the resource allocation procedure. The smaller the number, the higher the priority. The ”L1 Range” and ”L2 Range” columns represent the labor unit range that can maintain the same productivity as the originally assigned unit number.

Table 1: Activity Information

Activity ID (Description)	Duration (Days)	Predecessor
A (Foundation)	1	-
B (Ground-Floor Walls)	3	A(FS,1)
C (Floor Slab)	5	B(FS,0)
D (First-Floor Walls)	3	C(FS,1)

Table 2: Activity and Resource Assignment Information

Activity ID	Initial Labor 1 Assignment (Total)	Initial Labor 2 Assignment (Total)	Priority	L1 Range	L2 Range
A	2(2)	3(3)	1	1-2	1-3
B	4(12)	2(6)	2	3-6	1-3
C	5(25)	2(10)	3	4-6	1-4
D	3(9)	4(12)	4	2-3	2-4

Based on the solution procedure describes in the previous section, an initial solution can be found by using the multiple resource allocation algorithm. Figure 4 depicts the initial solution with 46 days project duration. The x-axis represents the location (5 repeated units of house), the y-axis is time (days). The four lines represents the four activities, while each segment means the start time and finish time of an activity at a certain location.

The SA algorithm is then applied in the initial solution to find a better solution. Under the assumptions of initial temperature T=1000, cooling ration r=0.01, the termination condition for a temperature setting to 300 moves, and the halting condition T<1, the SA algorithm finds an improved solution with 37 days project duration. Figure 5 shows the final solutions. The final solution adjusts the initial solution by assigning less labor units to Activities A, B, and D. Although the duration for the individual processes increase, Activity C can start together with B, moving its start time ahead and thus shortening the entire project duration by 9 days.

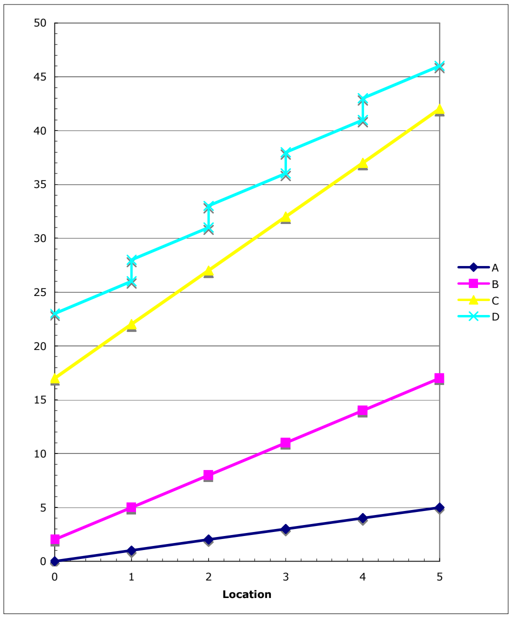

Figure 4: Initial Solution with 46-days project duration

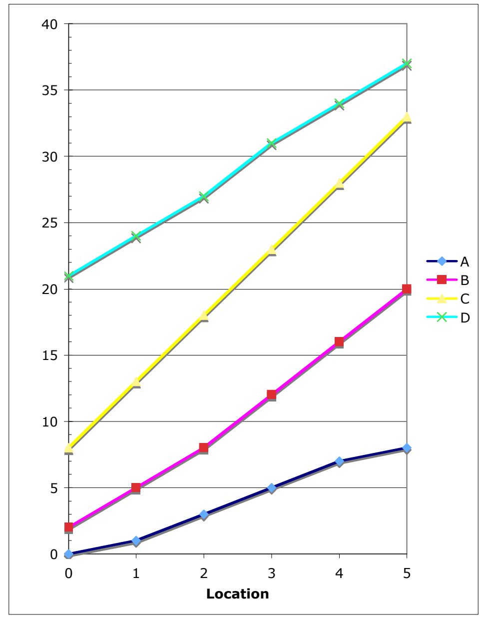

Figure 5: Final Solution with 37-days project duration

5. Conclusion

In summary, this research explored a system framework to allocate multiple resources and schedule activities for linear scheduling projects. It provides a quantitative analysis mechanism to the LSM. The two objective functions of the model, minimizing project duration and resource usage fluctuation, achieved the goal of considering resource allocation and leveling simultaneously. The two-stage solution procedure successfully incorporated the Simulated Annealing searching technique with the multiple resource allocation algorithm and proved to be effective in the study case.

References

[1] **Johnston, David W. (1981).** "Linear scheduling method for highway construction". *Journal of Construction Div. ASCE, 107(Co2):247–261.*

[2] **Harmelink, David J. and Rowings, James E.,** " Linear scheduling model: Development of controlling activity path". *Journal of Construction Engineering and Management, ASCE, 124(4):2263–268, 1998.*

[3] **Harmelink, David J.,** " Linear scheduling model: Float characteristics ". *Journal of Construction Engineering and Management, ASCE, 127:2263–268, 2001.*

[4] **Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P,** " Optimizaiton by Simulated Annealing ". *Science, 220:671-680, 1983.*

[5] **Zbigniew Michalewicz & David B. Fogel. (2000).** “How to Solve It: Modern Heuristics” *Springer:115–125.*

[6] **Liu, Shu-Shun (1999).** "Resource-Constrained Scheduling of Linear Projects: A Heuristic Approach Using Tabu Search Heuristics". *PhD thesis, Purdue University.*

[7] **Moselhi, Osama and Lorterapong, Pasit. (1993).** "Least impact algorithm for resource allocation". *Can. Journal. of Civil Engineering, 20:180–188.*