*24th International Symposium on Automation & Robotics in Construction (ISARC 2007)*
*Construction Automation Group, I.I.T. Madras*

# MODIFIED SIMULATED ANNEALING ALGORITHM AND MODIFIED TWO-STAGE SOLUTION FINDING PROCEDURE FOR OPTIMIZING LINEAR SCHEDULING PROJECTS WITH MULTIPLE RESOURCE CONSTRAINTS

**Chen Po-Han**
Assistant Professor
N1-01a-14-NTU -Singapore639798
Cphchen@ntu.edu.sg

**Shahandashti Seyed Mohsen**
Research Student
N1-B4-05-NTU-Singapore639798
Seye0001@ntu.edu.sg

## ABSTRACT

This paper presents a modified two-stage solution finding procedure and some modified simulated annealing algorithms to optimize linear scheduling projects with multiple resource constraints and their effectiveness is verified with a proposed problem. Simulated annealing and improved simulated annealing are compared in the same condition. The Reasons of obtaining improvement by modified simulated annealing is explained briefly too.

## KEYWORDS

Modified Simulated Annealing, Modified Two-Stage Solution Finding Procedure, Linear Scheduling, Multiple Resource Constraints

## 1. INTRODUCTION

Combinatorial optimization is widely used these days for engineering purposes and one of its applications is optimization of linear projects. Linear construction projects such as tunnels contain repeated activities at different locations. CPM is not an effective method for scheduling linear projects, so other methods such as Linear Scheduling Method (LSM) have been developed. Practically, resources are limited in construction projects so multiple resource constraints are incorporated in the model by"resource leveling" and "resource allocation". Optimizing linear scheduling projects with multiple resource constraints is a combinatorial optimization problem, so it should be optimized by one of the combinatorial optimization tools. Modified simulated annealing is chosen for this goal in this paper and it is compared in the same circumstances with simulated annealing to show its effectiveness.

## 2. LITERATURE REVIEW

LSM was introduced by Johnston in 1981 [1] in a highway construction project and simulated annealing is introduced in 1983 in the science magazine [2] in 1983. Harmelink (1995) [3] established a heuristic algorithm to determine the controlling activities path but without consideration of resource limitations. Mattila (1997) [4] considered resource leveling in a model of highway construction project. He solved the model by using mixed integer programming. Liu (1999) [5] considered single resource allocation and proposed a heuristic solution procedure using the Tabu Search algorithm in his model. The model included two stages. Lue & Hwang (2001) [6] proposed a precast production project and solved it with a genetic algorithm-based model. Finally Chung-I Yen (2005) [7] proposed simulated annealing for optimizing linear scheduling projects with multiple resource constraints. He considered resource allocation and resource leveling simultaneously.

Although the proposed models are useful, there are some modifications which can be done to make the models more realistic and easier to use. This paper discusses three modifications and the effectiveness of them is examined too.

## 3. PROBLEM OBJECTIVES

Assume there is a linear project with N activities, M locations, and I critical resources. The objectives are as following [7]:

$$Minimize\,Max\{f(n,m) \mid n=1..N; m=1..M\} \quad (1)$$

$$Minimize\sum_{t=1}^{T-1}\sum_{i=1}^{I} w_i\,(dp_{t,i}+dm_{t,i}) \quad (2)$$

There are some other equations which should be satisfied but they are not the main objective. Some of them are introduced here

Constraints:

- Activities precedence relationships
- Resource availability
- Activities completion
- Resource usage deviation

Notations:

$dp_{t,i}$ : Absolute difference plus value of resource I assignment between day t+1 and day t

$dm_{t,i}$ : Absolute difference minus value of resource I assignment between day t+1 and day t

$w_i$ : Weighting factor for resource i

## 4. MODIFIED SOLUTION METHODOLOGY

A two-stage solution-finding procedure is introduced in this section and two possible modifications are discussed too.

Stage one begins with solving the problem without resource constraint. A linear scheduling diagram will be drawn too. Then the problem will be solved by multiple resource allocation algorithms. It is assumed that resources are fixed for an activity in all locations. NTF concept is utilized in solving by considering multiple resource constraints. Stage one demand a time-consuming work; so a simple feasible solution for it is proposed in this paper. In this method we imagine that activities don't have any overlapping so the first activity of the project in all locations should be finished before the second activity with the second priority starts and it continues till the project is finished. In the other words, it is the maximum duration for the project but it guarantees that we have enough resources for activities to be done. It is assumed that resources are fixed for an activity in all locations in this stage. The effectiveness of this approach is examined later in this paper.

Simulated annealing is used during the second stage to find the optimized solution. The number of one of the resources of one of the activities is changed during each iteration of the algorithm to find the best allocation. The problem will be solved by multiple-resource allocation algorithm. NTF concept is utilized in solving by considering multiple resource constraints. The resources assigned to a repetitive activity can be varied at different locations within a specified range. The goal is to find best assignment of resources to activities and the best sequence of activities to have the minimum duration and fluctuation of resources assigned for a project. In the proposed models [7], having the minimum duration is the first objective and when only two projects have the same duration, the second criterion is considered. So we can conclude that fluctuation is a passive criterion. It is the location of the second modification. Most projects have a deadline. In the modified algorithm, the priority of objectives is not fixed and it depends on the deadline of the project. When the durations of comparing projects are more than the deadline, the duration objective has the priority and when the durations are less than the deadline, the fluctuation has the first priority. Search neighborhood is all possible resource assignments to activities. The maximum and minimum temperatures are 1000 and 1 respectively.

## 5. MODIFIED SIMULATED ANNEALING

Simulated annealing is based on the similarity between solid annealing process and combinatorial optimization. The algorithm consists of several decreasing temperatures.

Simulated annealing algorithm utilizes acceptance probability which helps it to escape from being trapped in local solution. The chance of acceptance of new solution is high at high temperatures and this chance reduces when temperature decreases. It is because the fact that the chance of being trapped in a local solution is high during first temperature steps. It works more random at the beginning and gradually turns into a more traditional local search algorithm [7].

Although the chance of being trapped in a local solution is high at the beginning, there is no need to have a certain iteration number. It can be less at the beginning and higher at the higher temperatures. It is because the fact that it is seen that the solution in higher temperatures is among last iterations while the solution in lower temperatures doesn't follow a fixed rule. It can be reduced with a lot of schemes but gradual reduction has been used in this paper. Geometric, arithmetic and logarithmic improvements [8] are presented in this paper.

Simulated annealing and modified simulated annealing algorithms test some feasible solutions to find the most optimized solution. It is needed to have the same number of testing to be able to compare them and show which one is more powerful. Number of testing of feasible solutions can be calculated as following.

$$N = \sum_{t=1}^{t_{\max}} N(t) \tag{3}$$

where N is the number of testing of feasible solutions, t is the number of times that temperature has been decreased which can be a number between 1 and $t_{max}$, and $t_{max}$ is the maximum number of times that the temperature has been changed.

Modifications are provided in three categories which are discussed later and corresponding computations are presented for each one but first it is needed to compute the number of testing for the non-modified simulated algorithm with a fixed number of iterations in each temperature. The following formula calculates the number of testing of feasible solutions for a simulated annealing with a fixed number of iterations in each temperature.

$$N = N_f \times t_{\max} \tag{4}$$

where $N_f$ is the number of iterations in each temperature which is fixed. For example, suppose that the initial temperature is 1000, the minimum temperature is 1, the number of iterations in each temperature is fixed to 1000, and the cooling ratio is 0.1. In this algorithm, the maximum number of times that temperature has been changed is 70, so the number of feasible solutions which are tested is 70000. The number of feasible solutions which are tested in the other algorithms must be 70000 to be able to show the probable advantages of them over the standard simulated annealing. The following subsections elaborate three methods to improve simulated annealing.

### 5.1 Geometric Improvement

In this method, the number of iterations in each temperature geometrically increases as t rises. Equation (5) represents the gradual reduction of iterations.

$$N(t) = t \times x \quad \Big| \quad t = 1 \ldots t_{\max} \tag{5}$$

The number of testing of feasible solutions can be easily computed with (6).

$$N = \sum_{t=1}^{t_{\max}} N(t) = x \times \left( {(t_{\max} + 1) \times t}\big/{2} \right) \tag{6}$$

where x is the parameter of geometric improvement which should be calculated for each specific example if the goal is comparison of this modification with the other methods. For instance, the following equation calculates x in order to have the same number of testing for the above-said example in which $t_{max}$ was 70 and the number of testing of feasible solutions for a non-modified simulated annealing was 70000.

$$x \times {(71 \times 70)}\big/{2} = 70000 \quad \rightarrow \quad x=28.17$$

### 5.2 Logarithmic Improvement

The number of iterations in each temperature logarithmically increases as t rises. (7) and (8) represent this method.

$$N(t) = x \big/ \log(T(t)) \quad \Big| \; t = 1 \ldots t_{\max} \tag{7}$$

$$N = \sum_{t=1}^{t_{\max}} N(t) = \sum_{t=1}^{t_{\max}} \left( x \big/ \log(T_{\max} \times e^{(-t*0.1)}) \right) \tag{8}$$

where T(t) is temperature as it has been changed t times and $T_{max}$ is the maximum temperature. x can be calculated for above-said example.

$$\sum_{t=1}^{70} \left( x \big/ \log(1000 * e^{(-t \times 0.1)}) \right) = 70000 \rightarrow x=18$$

4.3

### 5.3 Arithmetic Improvement

The number of iterations in each temperature arithmetically increases as t rises. (3.10) and (3.11) and (3.12) represent this method.

$$N(t) = N(t-1) + x \quad \Big| \quad t = 1 \ldots t_{\max} \tag{9}$$

$$N(1) = 10 \tag{10}$$

$$N = N(1) \times t_{\max} + x \times \left( \frac{t_{\max} \times (t_{\max} - 1)}{2} \right) \tag{11}$$

For example, x is computed for the above-said example as following.

$$10 \times 70 + x \times \left( \frac{70 \times (70-1)}{2} \right) = 70000 \rightarrow$$

$$x = 28.69$$

## 6. TEST OF THE MODIFICATIONS

The proposed problem is a part of a linear schedule. There are 6 identical semi-detached houses, so there are six locations. There are five repetitive activities which are flooring, utilities services, air-conditioning, painting and cleaning. Activities fight for two critical resources in the problem. The goal of this problem is to find the best resource assignment combination and the best sequence of activities to minimize the total project duration and the fluctuation of resource usage. The main objective is to find the minimum project duration. The second objective (minimum fluctuation of resource usage) is used when there are two schedules with the same duration. The following assumptions have been observed [7]:

1. A task can not be split. Once an activity is started, it will continue without interruption until it is finished.
2. Resources are limited. The limitation is assumed to be a constant across the entire project life span.
3. A resource con not be split.
4. Resources are assumed to maintain a constant productivity level within a certain range of assignment.
5. A resource can not be split.

Table 1 lists required information about the proposed artificial project and its activities. Table 2 presents information about two critical resources. The maximum number of resources 1 and 2 are 4 and 3 respectively.

**Table 1 Required Information About the Proposed Artificial Project and its Activities**

| Activity (Description) | Duration (Days) | Predecessor |
|---|---|---|
| A (Flooring) | 4 | |
| B (Utilities services) | 2 | A (FS0) |
| C (Air conditioning) | 2 | B (FS0) |
| D (Painting) | 2 | C (FS1) |
| E (Cleaning) | 1 | D (FS0) |

**Table 2 Information About Two Critical Resources**

| Act. | L1 | L2 | Total L1 Required | Total L2 Required | Activity priority | L1 Range | L2 Range |
|---|---|---|---|---|---|---|---|
| A | 3 | 2 | 4*3=12 | 4*2=8 | 1 | 2-4 | 1-3 |
| B | 2 | 1 | 2*2=4 | 2*1=2 | 2 | 1-3 | 1-2 |
| C | 2 | 1 | 2*2=4 | 2*1=2 | 3 | 1-3 | 1-3 |
| D | 1 | 1 | 2*1=2 | 2*1=2 | 4 | 1-2 | 1-2 |
| E | 1 | 1 | 1*1=1 | 1*1=1 | 5 | 1-2 | 1-2 |

**Table 3 Average and Most Optimized Answer Which is Achieved from 100 Implementations**

| | Average | Most Optimized answer |
|---|---|---|
| Simulated annealing | 46.42 | 43 |
| Geometrically improved | 45.01 | 42 |
| Arithmetically improved | 45.8 | 43 |
| Logarithmically improved | 44.77 | 41 |

**Table 4 Results of 100 Implementations of the Program with Two Types of Initial Solution**

| | Carefully produced initial solution | Easily produced initial solution |
|---|---|---|
| Average | 46.42 | 46.52 |
| Most optimized answer | 43 | 43 |
| Variance | 2.333 | 1.828 |

## 7. SUMMARY AND CONCLUSION

The problem is written in Java programming language. The written program is run 100 times for each of the following cases: no modification, modified simulated annealing, modified first stage and modified objective function. The following conclusions can be achieved easily from the results:

- Modified simulated annealing algorithms are significantly effective. The effectiveness is more highlighted when there is a real project with more activities, locations, and critical resources. Table 3 summarizes the results of implementations.
- There is not a significant change in the results of implementations when either the initial solution is simply produced based on the fact that activities should not have any overlapping or is intricately produced by multiple resource allocation algorithms. Table-4 depicts the results of 100 implementations of the program with two types of initial solution. It is due to the fact that simulated annealing and the model work somewhat randomly. Although the difference between results may be larger in multi-project or mega-project examples, the time which can be saved will justify using this approach. Table 5 shows a part of the conventional part to produce initial answer which is so time-consuming.
- The improved objectives of modified solution-finding procedure make the model more realistic

**Table 5 Producing the Initial Solution for the First Stage**

| step | CT | NT | As | Ap | L1 available | L2 available | L1 assigned | L2 assigned | assigned | completed |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 4 | A1 | - | 4 | 3 | 3 | 2 | A1 | A1 |
| 2 | 4 | 8 | A2 | - | 4 | 3 | 3 | 2 | A2 | A2 |
| 3 | 8 | 12 | A3 | - | 4 | 3 | 3 | 2 | A3 | A3 |
| 4 | 12 | 14 | A4 | - | 4 | 3 | 3 | 2 | A4 | - |
| 5 | 14 | 16 | B1 | A4 | 1 | 1 | 3 | 2 | A4 | A4 |
| 6 | 16 | 18 | A5,B1 | - | 4 | 3 | 3 | 2 | A5 | - |
| 7 | 18 | 20 | B1 | A5 | 1 | 1 | 3 | 2 | A5 | A5 |
| 8 | 20 | 22 | A6,B1 | - | 4 | 3 | 3 | 2 | A6 | - |
| 9 | 22 | 24 | B1 | A6 | 1 | 1 | 3 | 2 | A6 | A6 |
| 10 | 24 | 26 | B1 | - | 4 | 3 | 2 | 1 | B1 | B1 |
| 25 | 47 | 48 | E5 | D6 | 3 | 2 | 2 | 2 | E5,D6 | E5,D6 |
| 26 | 48 | 49 | E6 | - | 4 | 3 | 1 | 1 | E6 | E6 |

## 8. REFERENCES

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

[2] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. Science, 220:671-680, 1983.

[3] D.J. Harmelink. Linear scheduling model: the development of a linear scheduling model with micro computer applications of highway construction control. PhD thesis, Iowa State University, Ames, Iowa, 1995.

[4] Kris Gordon Mattila. Resource Leveling of Linear Schedules: A Mathematical Approach Using Integer Linear Programming. PhD thesis, Purdue University, 1997.

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

[6] Sou-Sen Lue and Shao-Ting Hwang. Optimal repetitive scheduling model with Shareable resource constraint. Journal of Construction Engineering and Management, ASCE, 127(4):270-280, 2001.

[7] Chung-I Yen. Simulated annealing for optimizing linear scheduling projects with multiple resource constraints. PhD thesis, Purdue University, 2005.

[8] Mark E. Johnson. Simulated Annealing. American Journal of Mathematical and Management Sciences, 1988.