MODIFIED SIMULATED ANNEALING ALGORITHM FOR OPTIMIZING LINEAR SCHEDULING PROJECTS WITH MULTIPLE RESOURCE CONSTRAINTS

Po-Han Chen[1] and Seyed Mohsen Shahandashti[1]
School of Civil and Environmental Engineering,
Nanyang Technological University,
50 Nanyang Ave, Singapore 639798

Abstract

This paper presents a modified simulated annealing algorithm to optimize linear scheduling projects with multiple resource constraints and its effectiveness is verified with a proposed problem. A two-stage solution-finding procedure is used to model the problem. Then the simulated annealing and the modified simulated annealing are compared in the same condition. The comparison results and the reasons of improvement by the modified simulated annealing are presented at the end.

Keywords: Optimization, Linear Scheduling, Modified Simulated Annealing, two-stage solution-finding procedure.

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 Line of Balance method (LOB), Vertical Production Method (VPM) and Linear Scheduling Method (LSM) have been developed. Among these methods, LSM is chosen in this model. 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.

LSM was introduced by Johnston in 1981 [1] in a highway construction project. Figure 1 depicts a Linear Scheduling Method diagram. It is a time versus location or distance diagram. Activities are presented as line segments, blocks, or bars in the diagram. The slop of the line segments represents the production rate of the corresponding activities. "Controlling activity path" in linear scheduling is a concept similar to "critical path" in

CPM. Controlling activity path is a set of activities that constitute a path and dictate project duration.

Simulated annealing is introduced in 1983 in the science magazine [2] in 1983.This concept has been used in several engineering applications since 1983 but it is introduced to construction management by Chung-I Yen (2005) [3]. LSM was introduced by Johnston in 1981 [1] in a highway construction project. Harmelink (1995) [4] established a heuristic algorithm to determine the controlling activities path but without consideration of resource limitations. Mattila (1997) [5] considered resource leveling in a model of highway construction project. He solved the model by using mixed integer programming. Liu (1999) [6] 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) [7] proposed a precast production project and solved it with a genetic algorithm-based model. Finally Chung-I Yen (2005) [3] proposed simulated annealing for optimizing linear scheduling projects with multiple resource constraints. He considered resource allocation and resource leveling simultaneously.

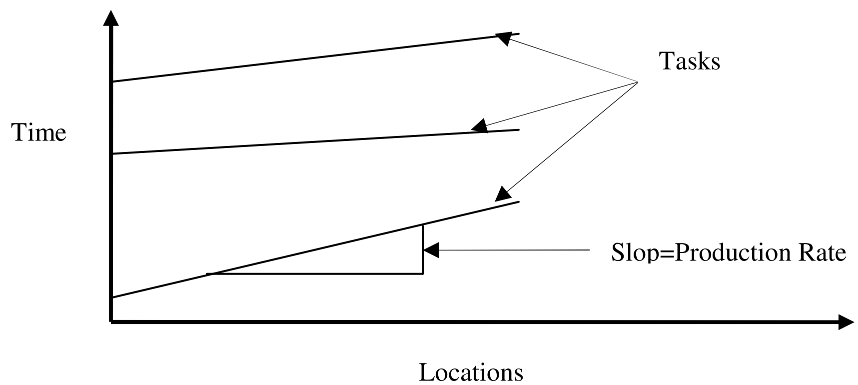

Figure1: Typical Linear Scheduling Diagram in Highway Construction [3]

2. Problem Formulation

Assume there is a linear project with N activities, M locations, and I critical resources.
The problem is formulated as follows [3]:

Objectives:

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

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

Constraints:

1. Activities precedence relationships

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

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

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

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

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

2. Resource availability

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

3. Activities completion

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

4. Resource usage deviation

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

$$\forall t = 1,T-1; i = 1,I$$

Notations:

$s(n,m)$: Start time of activity n at location m

$f(n,m)$: Finish time of activity n at location m

$dp_{t,i}$: 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

$r_i(n,m,t)$: Resource i assigned to activity n at location m at time t

$RA_i(t)$: Resource i availability at time t

$TR_i(n,m)$: Total amount of resource i required to complete activity n at location m

3. Simulated Annealing

Simulated annealing is based on the similarity between solid annealing process and combinatorial optimization. The algorithm consists of several decreasing temperatures. Suppose that finding the minimum of the cost function is favorable. Each temperature includes a sequence of iterations. First, the beginning temperature is chosen and the initial solution is selected and the cost function will be calculated. Then a new solution will be created in the neighborhood of the previous solution. New cost function will be calculated. If new cost function is less than the previous one, it will be accepted. If new cost function is less than the previous one, it will be accepted according to Metropolis's criterion [Metropolis et al., 1953] based on Boltzman's probability. According to Metropolis's criterion, if the difference between the cost function values of the current and the newly produced solutions (ΔE) is equal to or larger than zero, a random number δ in [0,1] is generated from a uniform distribution and if

$$\delta \leq e^{(-\Delta E/T)} \qquad \text{Eqn.(11)}$$

then the newly produced solution is accepted as the current solution. The number of new solutions which are created in each temperature is as many as the iteration number (termination condition). Iteration number can be a certain number of moves [3]. Then temperature will reduce upon temperature update rule and every above said step will be iterated until the temperature goes down the minimum temperature (halting criteria) [2]. The result will be affected with the number of iterations and the speed of reducing temperature. The halting criterion in this research is Eqn. 11.

$$Temperature = Te^{(-rt)} \qquad \text{Eqn.(12)}$$

where T is the initial temperature, r is a cooling ratio, and t is the number of times that temperature has been used. The chosen r is 0.1 in this research.

4. Modified Simulated Annealing

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 [3].

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 [8]. Geometric, arithmetic and logarithmic improvements 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, so some computations are required which are provided in three categories:

1. <u>Geometric improvement:</u>

    - Simulated Annealing Algorithm

$$t_{\max} = 70 \qquad \text{Eqn.(13)}$$
$$N(t) = 1000 \qquad \text{Eqn.(14)}$$
$$N = t_{\max} * N(t) = 70 * 1000 = 70000 \qquad \text{Eqn.(15)}$$

    - Geometrically improved simulated annealing algorithm

$$N(t) = t * 10 * x \quad \text{Eqn.(16)}$$

$$N = 10 * {(x^{t_{max}+1} - x)}/{(x-1)} \quad \text{Eqn.(17)}$$

where $N(t)$ is the number of iterations in each specified temperature, T is temperature, N is the Number of feasible solutions which are tested and t is the number of times that temperature has been used.

Eqn.(15) and Eqn.(18) should be equal to have the same number of feasible solutions which are tested in both algorithms so

$$10 * (\frac{x^{t_{max}+1} - x}{x-1}) = 70000 \quad \rightarrow \quad x=1.096 \quad \text{Eqn.(18)}$$

2. Logarithmic improvement

$$N(t) = {x}/{\log(T(t))} \quad \text{Eqn.(19)}$$

$$N = \sum_{t=1}^{70} ({x}/{\log(T_{max} * e^{(-t*0.1)})}) \quad \text{Eqn.(20)}$$

Eqn.(15) and Eqn.(20) should be equal to have the same number of feasible solutions which are tested in both algorithms so

$x = 184.305$

3. Arithmetic improvement

$$N(t) = N(t-1) + x \quad \text{Eqn.(21)}$$

$$N(1) = 10 \quad \text{Eqn.(22)}$$

$$N = N(1) * t_{max} + (\frac{t_{max} * (t_{max} + 1)}{2}) \quad \text{Eqn.(23)}$$

Eqn.(15) and Eqn.(23) should be equal to have the same number of feasible solutions which are tested in both algorithms so

$x = 27.8873$

4.1 Example

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 [3]:

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, i.e., the resource amount is a positive integer.
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. Table 2 presents information about resources. It is assumed that resources are labors. The maximum number of Labor 1 and 2 are 4 and 3 respectively.

Table1: Required information about the proposed project

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)

Table2: Required information about resources

Activity	Labour1 (men/day)	Labour2 (men/day)	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

4.2 Solution Methodology

A two-stage solution-finding procedure is introduced in this section. These two stages are explained briefly.

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 algorithm. It is assumed that resources are fixed for an activity in all locations. NTF concept is utilized in solving by considering multiple resource constraint. Table 2 represents selected parts of the solution with multiple resource constraint. It will be initial solution for stage 2.
Simulated annealing is used during stage 2 to find the optimized solution. 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 to have the minimum duration and fluctuation of resources assigned for the project. So search neighborhood is all possible resource assignments to activities. The maximum and minimum temperatures are 1000 and 1 respectively.

Table3: Initial solution for stage 2

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

CT: Current Time
NT: Next Time
NTF: Next Time Frame
As: Activities that can start at CT
Ap: Activities already in progress at CT
Af: Activities can start in short future

4.3 Results

The problem is written in Java programming language. Results of 100 implementations of each one are listed in Table4. Averages and the most optimized answers which are achieved from 100 implementations of each algorithm are represented in table5. The following conclusions can be achieved easily from the results.

1. 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.
2. It is possible to improve simulated annealing algorithm without spending more time and testing more feasible solutions.
3. All improved algorithms (geometrically, arithmetically, and logarithmically improved algorithms) are more effective than the simple one.
4. Logarithmically improved algorithm is the most effective algorithm. The average of results and the most optimized answer are 44.77 and 41 respectively for this algorithm.
5. The less effective improved algorithm is the arithmetically improved one.

Table4: Results of 100 implementations of each algorithm

Implementation no.	Geometrically improved	Ordinary	Logarithmically improved	Arithmetically improved	Implementation no.	Geometrically improved	Ordinary	Logarithmically improved	Arithmetically improved
1	45	48	42	47	51	46	44	43	47
2	44	47	47	44	52	43	47	48	46
3	45	46	47	47	53	45	44	44	46
4	46	48	48	47	54	44	48	45	45
5	43	46	42	46	55	46	47	43	44
6	45	43	46	47	56	45	46	48	47
7	46	48	43	48	57	44	46	47	43
8	45	45	46	45	58	43	46	47	45
9	44	46	44	47	59	46	45	44	45
10	46	46	43	46	60	47	48	45	47
11	47	48	42	43	61	46	46	44	46
12	45	47	46	47	62	46	46	43	45
13	44	47	45	49	63	45	47	43	47
14	45	44	47	46	64	43	44	43	43
15	43	45	43	44	65	47	49	44	44
16	46	45	45	47	66	44	45	43	45
17	43	47	43	45	67	47	44	41	47
18	46	48	45	46	68	46	46	46	46
19	44	46	47	46	69	45	45	48	48
20	45	47	44	47	70	44	48	47	48
21	45	48	46	46	71	45	49	46	43
22	45	49	45	44	72	45	45	46	43
23	42	46	46	45	73	46	47	45	46
24	46	46	46	45	74	45	44	43	47
25	43	46	48	45	75	45	46	44	46
26	47	48	42	44	76	45	48	43	46
27	46	46	46	48	77	47	46	44	47
28	45	44	46	45	78	45	46	43	47
29	46	45	41	46	79	44	45	45	46
30	46	46	45	45	80	45	46	46	46
31	44	47	48	45	81	45	44	45	45
32	44	47	44	48	82	44	46	47	48
33	46	48	44	44	83	47	46	45	45
34	46	46	47	46	84	48	48	45	47
35	45	45	46	45	85	45	46	42	44
36	45	45	45	47	86	44	49	43	45
37	45	45	43	47	87	43	47	45	46
38	48	48	43	47	88	45	47	42	46
39	45	46	45	47	89	47	46	48	45
40	44	46	44	46	90	45	45	44	44
41	44	47	43	47	91	46	47	44	45
42	45	46	45	45	92	44	47	45	47
43	45	46	45	45	93	45	47	43	47
44	46	48	43	44	94	46	47	46	47
45	44	45	43	45	95	43	47	48	48
46	46	46	43	46	96	44	48	49	45
47	44	46	48	45	97	43	47	46	47
48	45	47	47	48	98	46	49	42	43
49	43	47	45	44	99	43	49	43	44
50	47	49	43	47	100	45	46	44	46

Table5: Average and most optimized answer from 100 implementations

	Simulated annealing	Geometrically improved	Arithmetically improved	Logarithmically improved
Average	46.42	45.01	45.8	44.77
Most Optimized answer	43	42	43	41
Number of iteration of the most optimized answer	1	1	6	2

5. Conclusion

In this paper, a modified simulated annealing algorithm was proposed and compared to the conventional simulated annealing approach. The comparison between the proposed method and the conventional approach was carried out using a multiple-resource-constraint linear scheduling problem. A two-stage solution-finding procedure was adopted to model the problem. The comparison result, after 100 iterations, indicated that the proposed method has an advantage over the conventional approach by reaching a better optimized output.

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

[4] 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.

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

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

[7] 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.

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