Journal of The Korea Society of Computer and Information
Vol. 20 No. 9, pp. 81-86, September 2015
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.9.081

Complete Time Algorithm for Stadium Construction Scheduling Problem

Sang-Un Lee*

Abstract

This paper suggests heuristic algorithm with linear time complexity to decide the normal and optimal point at minimum loss/maximum profit maximum shortest scheduling problem with additional loss cost and bonus profit cost. This algorithm computes only the earliest ending time for each node. Therefore, this algorithm can be get the critical path and project duration within $O(n)$ time complexity and reduces the five steps of critical path method to one step. The proposed algorithm can be show the result more visually than linear programming and critical path method. For real experimental data, the proposed algorithm obtains the same solution as linear programming more quickly.

▸ Keyword : Critical path, Earliest ending time, Loss cost, Profit cost, Optimal point

I. Introduction

건축이나 소프트웨어 개발과 같은 프로젝트를 수행할 경우, 프로젝트 관리자는 m개의 작업 노드들 상호간에 n개의 순서(호)와 작업시간이 결정된 프로젝트를 최단기간 내로 수행하고자 한다. 이 경우, 하나의 작업은 반드시 이전 작업들 중 가장 마지막으로 종료된 작업 이후에만 시작할 수 있다. 이 문제를 임계경로(critical path, CP)를 찾는 일정계획 문제(scheduling problem, SP)라 한다[1-3]. 즉, 임계경로 상의 마지막 작업을 종료하는 시점이 프로젝트의 최소 소요 기간으로, 임계경로 상의 임의의 작업에 지연이 발생한 만큼 프로젝트 소요기간에 영향을 미치기 때문에, 임계경로 상의 작업들에 대해서는 정시에 작업을 수행하고, 정시에 종료해야만 프로젝트 일정 지연에 따른 위약금을 배상하지 않을 수 있다.

정상적인 작업수행 기간이 주어진 경우를 정상 일정 계획(normal SP, NSP)라 하자. 만약, 추가 인력과 장비를 투입하여 특정 작업의 수행기간을 단축시킬 수 있고, 이 경우 추가 입력과 장비 투입에 따른 추가 소요 비용(손실비용)이 발생하며, 프로젝트 일정 단축에 따른 보너스(이득비용)를 받을 수 있다면, 손실비용과 이득비용의 최적 점을 찾는 최대로 단축 가능한 프로젝트 일정을 찾는 문제가 제기될 수 있다[4]. 이를 최적 일정계획 문제(optimal SP, OSP)라 하며, 경제학에서의 최대 이익/최소 손실 점을 찾는 "Economics MR= MC" 문제로 볼 수 있다[5].

NSP에 대한 임계경로를 구하는 일반적인 방법으로는 임계경로법(critical path method, CPM)이 있다[6]. 이 방법은 5단계를 수행하여 임계경로를 찾는 복잡한 계산과정을 수행하지만 호 수 n개에 대해 수행 복잡도는 $O(n)$으로 알려져 있다. Guéret 등[4]은 선형계획법(linear programming, LP)을 적용하여 NSP와 OSP의 해를 구하였다. 노드 수 m에 대해 LP를 수행하기 위해서는 $O(m^4)$의 수행 복잡도로 알려져 있다.

본 논문에서는 NSP와 OSP에 대해 CPM의 5단계를 1단계로 축소하여 수행하는 수행 복잡도 $O(n)$의 알고리즘을 제안한다. 2장에서는 CPM 개념과 NSP와 OSP에 대한 Guéret 등[4]의 적용 결과를 고찰해 본다. 3장에서는 작업 종료시간 계산법으로 NSP와 OSP의 해를 $O(n)$ 수행 복잡도로 구할 수 있는 휴리스틱 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 실제 데이터에 적용하여 알고리즘 적합성을 평가해 본다.

II. CPM Concepts

프로젝트 관리자는 프로젝트 수행에 요구되는 활동들을 결정하고, 활동들의 수행 기간과 활동들 간의 의존관계가 결정되면 프로젝트 수행 일정을 계획하기 위해 망 다이어그램(network diagram)을 작성한다[7]. 다음으로 CPM을 망 다이어그램에 적용하여 임계경로를 계산하여 프로젝트 수행에 소요되는 최

• First Author: Sang-Un Lee, Corresponding Author: Sang-Un Lee
*Sang-Un Lee (sulee@gwnu.ac.kr), Dept. of Multimedia Engineering, Gangneung-Wonju National University
• Received: 2015. 07. 14, Revised: 2015. 07. 31, Accepted: 2015. 08. 14.

소 기간을 산출한다. 따라서 임계경로는 프로젝트 수행 기간을 결정하는 가장 중요한 요소이다[3].

임계경로를 계산하는 CPM은 네트워크 다이어그램에서 순방향으로 T_{ES} (earliest starting time,)를, 역방향으로 T_{LS} (latest starting time), 순방향으로 S_N (node slack time)과 S_A (total arc slack time)을 계산하고, $S_A = 0$ 인 호 들을 임계경로로 설정하는 5단계를 거친다[1-3].

CPM으로 얻은 네트워크 다이어그램을 프로젝트 일정 계획을 시각적으로 명확히 표현하는 PERT/GANTT 차트로 변환하기 위해 활동들의 수행 순서를 시각적으로 명확히 알 수 있도록 재배치하는 과정이 필요하다. 이를 위해 Lucko[8]는 CPM 계산 후 활동들의 수행 순서를 재배치하는 9 단계 과정을 제안하였다.

프로젝트 망은 노드(node 또는 event)와 호(arc 또는 activity)로 구성되어 있으며, 활동에 소요되는 노력(또는 기간)을 표현하는 방식에 따라 AoA(activity on arc)와 AoN (activity on node)이 있다[7-9]. AoA 망에서 노드는 시간과 노력을 소비하지 않는 이정표이며, 호는 시간과 노력을 소비하는 활동으로 표현된다. 반면에 AoN은 활동을 노드에 표현한다. AoA와 AoN 표현 중 어떤 방법이 더 좋은지는 의견이 분분한 상태이며, 대체로 AoN 표현 방법을 더 선호하는 경향이 있다[9].

CPM을 적용하여 망 다이어그램의 임계경로를 결정한 경우, 이를 PERT/GANTT 차트로 활용하기 위해 망 다이어그램의 노드들을 재배열시켜야만 한다. 이에 대한 Lucko[8] 방법은 CPM의 5 단계를 적용하여 임계경로를 찾고, 노드들의 T_{ES} 를 오름차순으로 정렬시켜 활동들을 재배치시키는 9 단계의 복잡한 과정을 거쳐 그래프의 표현을 명확히 하는 방법을 제시하였다. 그러나 Lucko[8] 방법으로 T_{ES} 를 오름차순으로 정렬시켜 활동들을 재배치하는데도 어려움이 있다.

Shankar와 Sireesha[10]은 CPM을 구하는 과정에서 T_{ES}, T_{LS} 와 S_A 의 값을 구하는 어려움을 조금이라도 쉽게 하고자 Dijkstra의 최단경로 (shortest path)를 구하는 알고리즘을 접목시켰다. 이와 같이 CPM은 $O(m)$ 복잡도로 단순해 보이지만 5단계를 수행하는 복잡성 또한 내포하고 있음을 알 수 있다.

NSP와 OSP에 대한 Guéret 등[4]이 제시한 [표 1]의 문제를 고찰해 보자.

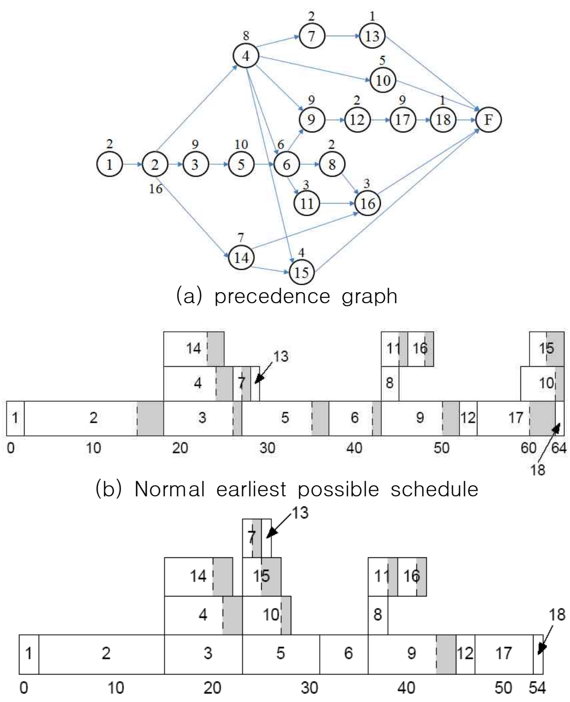
[표 1]은 한 도시의 의회에서 지역 주민들에게 공공서비스를 증진시키기 위해 소규모 경기장을 건축하고자 하는 계획이다. 입찰 공고가 난 이후에 한 지역 건축회사는 건축을 승인받아 가능한 최단기간 내에 완공하고자 한다. [표 1]은 건축에 필요한 주요 작업내용으로 18개 작업으로 구성되어 있으며, 각 작업의 수행 기간(단위 : 주간)과 이전 작업(predecessors)이 명시되어 있다. 즉, 특정 작업은 이전 작업이 완료된 이후에만 작업에 착수할 수 있는 조건을 만족해야 한다.

여기서, 첫 번째로 제기되는 문제(Q1)는 건축을 완공하기 위한 가장 빠른 가능한 기간을 구하고자 한다. 이는 임계경로를 구하는 문제이다. 두 번째로 제기되는 문제(Q2)로, 첫 번째 문제에서 건축회사에서 제시한 기간에 대해 시의회는 보다 일정을 단축시키기를 원하며, 일정을 단축시키는 매 주간 단위로 €30,000(€30K)의 보너스(bonus, performance-based benefits)를 제시하였다. 건축가는 추가 작업자를 고용하고, 보다 많은 장비들을 사전에 준비하여 일정을 단축시킬 수 있으며, 각 작업별로 최대로 단축시킬 수 있는 일정은 Max reduction duration에 제시하였다. 단, 이 때 추가로 소요되는 비용은 Additional cost per week에 제시하였다. 여기서, 건축회사는 이익(profit)을 최대로 할 수 있는 최소 건축기간을 도출하고자 한다.

Table 1. Scheduling data for stadium construction

Task	Description	Duration (week)	Predecessors	Max Reduction duration (week)	Additional Cost per week (in 1000 €)
1	Installing the construction site	2	-	0	-
2	Terracing	16	1	3	30
3	Constructing the foundations	9	2	1	26
4	Access roads and other networks	8	2	2	12
5	Erecting the basement	10	3	2	17
6	Main floor	6	4,5	1	15
7	Dividing up the changing rooms	2	4	1	8
8	Electrifying the terraces	2	6	0	-
9	Constructing the roof	9	4,6	2	42
10	Lighting of the stadium	5	4	1	21
11	Installing the terraces	3	6	1	18
12	Sealing the roof	2	9	0	-
13	Finishing the changing rooms	1	7	0	-
14	Constructing the ticket office	7	2	2	22
15	Secondary access roads	4	4,14	2	12
16	Means of signalling	3	8,11,14	1	6
17	Lawn and sport accessories	9	12	3	16
18	Handing over the building	1	17	0	-

Guéret 등[4]은 [표 1]의 경기장 건축 일정 데이터에 대해, 선형계획법을 활용한 Mosel 패키지 프로그램을 작성하여 [그림 1]의 (a) 그래프를 활용하여, Q1에 대해서는 (b)와 같이 64주가 소요되는 건축 일정을 얻었다. Q2에 대해서는 (c)와 같이 54주로 10주를 단축시켜 €87K의 이득을 얻었다.



(a) precedence graph

(b) Normal earliest possible schedule

(c) Optimal possible shortest schedule with maximum profit

Fig. 1. Construction scheduling plan of linear programming

3장에서는, [표 1]의 문제에 대해, 먼저, 임계경로를 $O(n)$ 의 선형 복잡도로 빠르게 찾을 수 있는 알고리즘을 제안하고, 다음으로, 이익을 최대로 할 수 있는 가능한 최대 단축기간도 $O(n)$ 선형 복잡도로 찾을 수 있는 알고리즘을 제안한다.

III. Complete Time Algorithm

CPM은 작업 시작 시작에 기반하여 임계경로를 구하는 시작시간 법(start time method, STM)이다. 본 장에서는 종료시간에 기반하여 CPM에 비해 보다 빠르게 임계경로를 구하는 종료시간 법(finish time method, FTM)을 제안한다.

주어진 문제에 대해, 첫 번째로 시작하는 작업 노드를 근 노드(root node)로, 자 노드(children node)가 없는 노드들에 가상의 자 노드인 단 노드(terminal node)를 추가하는 트리를 형성한다. 형성된 트리에 대해, 근 노드를 레벨 1, 근 노드의 자 노드들을 레벨 2, 레벨 2의 자 노드들을 레벨 3,⋯ 등으로 하여 단 노드까지 레벨을 부여한다. 이 트리에 대해 레벨단위의 너비우선 탐색(breadth first search, BFS)으로 식 (1)과 같이 가장 빠른 종료시간(earliest finish time, T_{EF})을 계산한다. 즉, 현재 레벨에 있는 자 노드 n_j의 작업종료시간 $T_{EF}(n_j)$는 부 노드들 중에서 가장 마지막에 종료되는 작업 종료시간인 $_{\max}T_{EF}(n_i)$ 이후에 시작하여 n_j의 작업 수행시간 $t_d(n_j)$이 경과한 시점이 된다.

$$T_{EF}(n_j) = {}_{\max}T_{EF}(n_i) + t_d(n_j) \tag{1}$$

such that n_i: 부 노드
n_j: 자 노드
T_{EF} : 가장 빠른 작업 종료시간
t_d : 해당 작업 수행시간

이와 같이 근 노드로부터 단 노드까지 가장 빠른 작업종료시간 호들이 결정되면 단 노드의 가장 빠른 작업 종료시간이 가능한 최단기간 일정 값이 된다. 단순히 단 노드에 연결된 노드들은 여유시간이 전혀 없는 $S_N = 0$ 로 이 노드들로 구성된 경로가 임계경로가 된다. 임계경로 상의 노드들의 일정 지연만큼 총 일정에 동일한 영향을 미친다. 따라서 임계경로상의 노드들을 정시에 종료할 수 있도록 관리가 요구된다. 임계경로 상에 존조하지 않는 노드를 n_i, 임계경로 상의 자 노드를 n_j라 하면, 임계경로 상에 존재하지 않는 노드들의 여유시간은 식 (2)로 계산된다.

$$\text{Latest finish time: } T_{LF}(n_i) = {}_{\min}T_{EF}(n_j) - T_{EF}(n_i)$$
$$\text{Slack time: } S_N(n_i) = T_{LF}(n_i) - T_{EF}(n_i) \tag{2}$$

이득을 최대로 할 수 있는 가능한 최단 일정을 구하는 문제에 대해서는, 기간 단축으로 인해 추가 발생하는 비용을 c_e, 보너스 비용을 c_b라 할 때, 식 (3)을 만족하는 노드들에 대해 가능한 단축기간으로 재설정한다.

$$c_e \le c_b \tag{3}$$

식 (3)을 만족하는 노드들은 거의 대부분 임계경로 상의 노드들이다. 따라서, 임계경로 상의 노드들만을 대상으로 쉽게 해를 구할 수 도 있지만, 이따금, 임계경로 상에 존재하지 않는 노드들의 작업시간 단축에 따라 임계경로에 영향을 미칠 가능성도 있기 때문에, 일반적인 방법으로 식 (3)을 만족하는 모든 노드들을 대상으로 결정한다.

재설정된 작업시간을 가진 트리에 대해 레벨 오름차순으로 각 노드의 가장 빠른 작업 종료시간 T_{EF}를 계산하고, 임계경로를 구한다.

다음으로, 임계경로 상에 위치하지 않은 노드들 중에서 작업시간이 단축된 경우, 원래의 작업시간으로 복구시킨다. 이는 작업시간 단축에 따른 이득이 없이 단지 초과 소요비용만이 추가로 소요되기 때문이다.

마지막으로, 단축기간에 대한 보너스 비용에서 임계경로상의 기간 단축에 따른 추가 비용 노드들의 추가비용 합을 뺀 값으로 기간 단축에 따른 이득을 계산한다.

제안된 알고리즘은 엄밀히 말해 해당 작업의 가장 빠른 종료시점(earliest ending time)을 얻는 방법으로, 이를 종료시간법 알고리즘(complete time method algorithm, CTMA)이라 하며, [그림 2]와 같이 수행된다.

[작업종료시간 기준 임계경로 결정법]

Step 1. 첫 번째 시작 노드를 근 노드로, 자 노드가 없는 노드들을 하나의 가상 노드인 단 노드로 연결, 근 노드를 레벨 1, 근 노드로부터 유입되는 노드들(근 노드의 인접 노드들, 근 노드의 자 노드들) 을 레벨 2, 레벨 i 노드들의 자 노드들을 레벨 $i+1$ 로 단 노드까지 레벨 설정.

Step 2. 레벨 오름차순 BFS로 레벨단위로 각 노드의 작업 종료시간 T_{EF} 계산. /* 수행 복잡도 : $O(n)$ */
부 노드를 n_i, 자 노드를 n_j라 할 때,
$T_{EF}(n_j) = {}_{\max}T_{EF}(n_i) + t_d(n_j), (n_i, n_j)$ 호 선택

Step 3. 단 노드부터 근 노드까지 연결된 호들 (n_i, n_j) 선택, 임계경로로 결정. /* 수행 복잡도 : $O(m)$ */

[최대이익 최단기간 결정법]

Step 1. 기간 단축으로 인해 추가 발생하는 비용 c_e 와 보너스 비용 c_b에 대해, $c_e \le c_b$ 인 노드들을 대상으로 기간 단축

Step 2. "작업 종료시간 기준 임계경로 결정법"으로 임계경로 결정

Step 3. 임계경로 상에 존재하지 않는 노드들은 원래의 정상 작업 수행시간으로 복구.

Step 4. 단축기간에 대한 보너스 비용에서 임계경로상의 기간 단축에 따른 추가 비용 노드들의 추가비용 합을 뺀 값으로 기간 단축에 따른 이득 계산.

Fig. 2. Complete time method algorithm

CTMA는 임계경로를 결정함에 있어 실제로 Step 2만을 수행하여 CPM의 5단계를 1단계로 축소시킬 수 있었다. Step 2는 BFS의 작업 종료시간 계산에 $O(n)$, 임계경로 결정에 $O(m)$이 소요되어, $O(n)$의 선형 복잡도 알고리즘이다. 왜냐하면, 노드 m과 호 n에 대해 일반적으로 $m \le n$이기 때문이다.

IV. Experiments and Result Analysis

본 장에서는 Guéret 등[4]에서 인용된 [표 1]의 실험 데이터에 대해 CTMA를 적용하여 본다. CTMA로 작업 종료시간을 구하여 임계경로를 결정한 결과는 [그림 3]에 제시하였다. 제안된 알고리즘은 레벨 순서로 n개 노드에 대해 {1}, {2}, {4,3,14}, {7,10,5,15}, {6,13}, {8,9,11}, [12,16}, {17}, {18}, {F} 순서로 수행 복잡도 $O(n)$으로 작업이 종료되는 시간을 계산하여 괄호 안에 표기하였다. 이 결과 [그림 3]에서, 총 소요 일정은 LP와 동일하게 64주를 얻었다. 따라서 제안된 알고리즘은 LP의 수행 복잡도 $O(m^4)$를 $O(n)$으로 단순화시켰으며, LP의 [그림 2] (b)에 비해 임계경로가 1→2→3→5→6→9→12→17→18임을 시각적으로도 보다 쉽게 파악할 수 있는 장점이 있다.

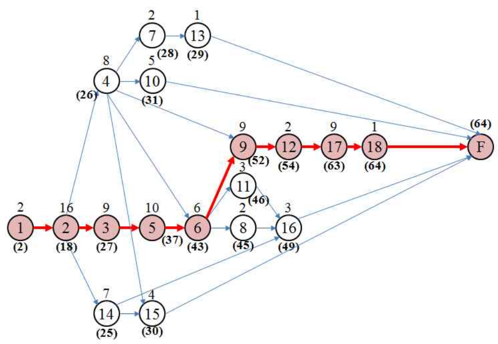
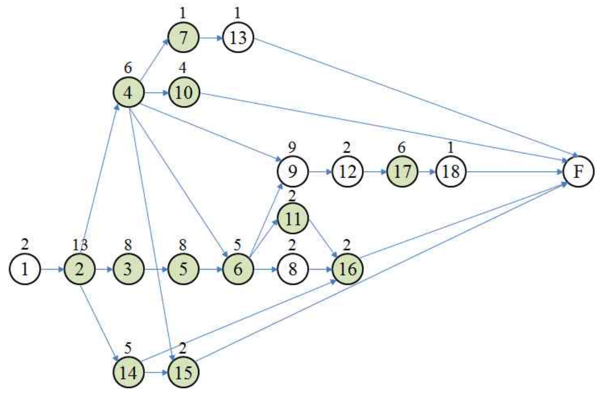
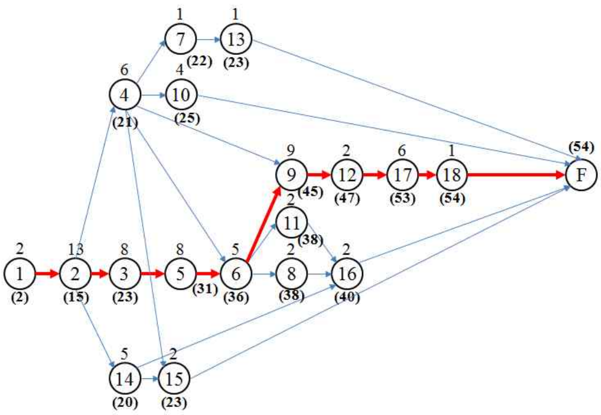


Fig. 3. Complete time and critical path using CTMA

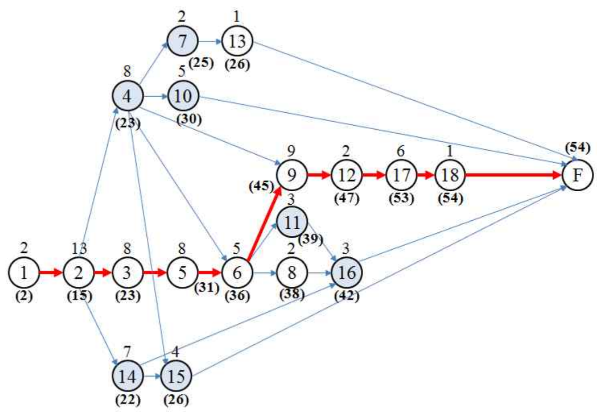
다음으로, 보너스를 최대로 받기 위해 가능한 최대 단축 일정을 구하는 문제에 대해, CTMA를 적용한 결과는 [그림 4]에 제시하였다. (a)에서는 작업시간을 단축가능하면서 $c_e \le c_b$인 노드들 2,3,4,5,6,7,10,11,14,15,16,17에 대해 최대로 단축 가능한 작업시간으로 치환한 결과이다. (b)는 최소 작업시간 망에 대해 각 작업의 가장 빠른 종료시간 $T_{EF}(n_i)$를 계산하고, 임계경로를 구한 결과이다. 이 결과는 정상적인(normal) 작업일정에 대한 그림 3의 64주에 대한 임계경로와 동일함을 알 수 있으며, 단지 64주를 54주로 단축시킬 수 있었다. (b)에서는, 임계경로 상에 존재하지 않는 작업시간을 단축한 노드들에 대해서는 원 상태의 정상적인 작업시간으로 복귀하여 추가 소요 비용을 "0"으로 치환하여 (c)의 결과를 얻었다. 4,7,10,11,14,15,16 노드들이 이에 해당된다. (d)는 10주 절감에 따른 보너스 €300K로 부터 임계경로 상의 작업시간을 단축한 2,3,5,6,18 노드에 대해 추가 소요비용 합 €213K를 제외한 순이익 €87K를 얻은 결과이다.



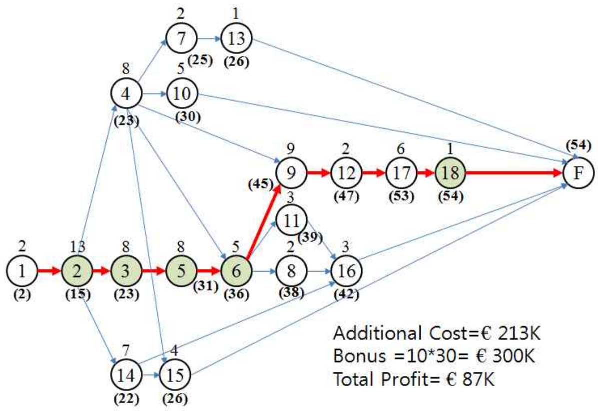
(a) Reduces duration for $c_e \le c_b$ nodes



(b) Complete time and critical path for minimum duration



(c) Recovery to original duration on non-critical path nodes



(d) Maximum profit nodes

Fig. 4. Solution of maximum profit using CTMA

제안된 CTMA와 LP, CPM의 성능을 비교한 결과는 [표 2]에 제시하였다.

Table 2. Compare with algorithm performance

알고리즘	수행 복잡도	수행 단계	정상 수행일정	일정 단축	
				단축기간	이득
LP	$O(m^4)$	-	**64**	**10주**	**€87K**
CTMA	$O(n)$	1	**64**	**10주**	**€87K**
CPM	$O(m)$	5	**64**	**10주**	**€87K**

표에서 CTMA는 LP와 동일한 해를 얻었음을 알 수 있다. 그러나 정상적인 경우 뿐 아니라 추가 소요비용을 상쇄하고 보너스를 최대로 얻을 수 있는 최적의 단축 일정을 찾는 문제에 대해서도 LP는 $O(m^4)$의 복잡한 최적화 과정으로 해를 찾으면서도 어떻게 해를 찾아가는지에 대한 답을 제시하지 못하고 있다. 반면에, 제안된 CTMA는 $O(n)$의 선형 복잡도로 레벨 순서 단위로 각 레벨에 속한 노드들의 작업 종료시간을 계산하는 규칙을 제시함으로서, 시각적으로도 쉽게 임계경로를 파악할 수 있을 뿐만 아니라 CPM 보다도 빨라 임계경로를 구하는 방법들 중에서는 가장 빨리 해를 얻을 수 있다.

V. Conclusions

본 논문은 정상적인 작업시간이 소요되는 경우와 작업 시간을 단축할 경우 추가 소요 비용인 손실비용과 보너스에 따른 이득비용의 최적 점(최소 손실/최대 이득)을 찾는 최대 단축 일정 문제에 대해, 선형 복잡도의 휴리스틱 알고리즘을 제안하였다.

본 문제에 대해, Guéret 등[4]은 선형계획법을 적용하였으며, 일반적으로는 5단계를 수행하는 임계경로법을 적용할 수도 있다. 그러나 선형계획법은 $O(m^4)$의 수행 복잡도가 요구되며, 임계경로법은 $O(n)$ 복잡도가 요구되지만 5단계를 수행해야 한다. 본 논문에서는 단지 각 노드의 가장 빠른 종료시간만을 계산하는 방법으로 임계경로법의 5단계를 1단계로 축소시켜 $O(n)$의 선형 복잡도로 임계경로를 구하고, 프로젝트의 소요 기간을 구할 수 있었다. 제안된 알고리즘은 선형계획법이나 임계경로법에 비해 작업의 순서를 보다 시각적으로 보여줄 수 있으며, 필요시 임계경로법에서 계산되는 각종 시간을 쉽게 계산할 수 있도록 하였다.

제안된 알고리즘을 실제 사례에 적용한 결과 LP와 동일한 해를 빠르게 얻었다.

따라서 제안된 알고리즘은 간단하고 쉽게 해를 구할 수 있는 관계로, 알고리즘에 대한 초보 지식을 갖고 있지 못한 프로젝트 관리자에게도 실제로 큰 도움을 줄 수 있을 것이다.

REFERENCES

[1] C. Z. D. Santos, “Critical Path in Software Management," Interactive Systems & Consulting Inc, 2005.

[2] J. H. Kang, "COSC 621: Advanced Construction Project Scheduling and Management," Department of Construction Science, College of Architecture, Texas A&M University, 2005.

[3] C. Hendrickson, "Project Management for Construction: Fundamental Concepts for Owners, Engineers, Architects and Builders," http://www.ce.cmu.edu/pmbook/10_Fundamental_Scheduling_Procedures.html, 2003.

[4] C. Guéret, X. Prins, and M. Sevaux, "Applications of Optimization with Xpress-MP: 7.1 Construction of a Stadium," Dash Optimization Ltd., pp. 81-86, Feb. 2005.

[5] M. Rieberman and J. F. Hall, "Introduction to Economics; Chapter 6. How Firms make Decisions: Profit Maximization," 2ed, South-Western/Thomson Learning, 2005.

[6] K. James, "Critical Path Planning and Scheduling: Mathematical Basis," Operations Research, Vol. 9, No. 3, pp. 296-320, May 1961.

[7] M. Sniedovich, "Towards an AoA-Free Courseware for the Critical Path Method," INFORMS Transactions on Education, Vol. 5, No. 2, Jan. 2005.

[8] G. Lucko, "An Activity and Arrow Arranging Algorithm for Clarity in Schedule Network Diagrams," Joint International Conference on Computing and Decision Making in Civil and Building Engineering, pp. 752-761, Jun. 2006.

[9] Y. Cohen and A. Sadeh, "A New Approach for Constructing and Generating AOA Networks," Journal of Engineering, Computing and Architecture, Vol. 1, No. 1, pp. 1-13, Jan. 2007.

[10] N. R. Shankar and V. Sireesha, “Using Modified Dijkstra's Algorithm for Critical Path Method in a Project Network,” International Journal of Computational and Applied Mathematics, Vol. 5, No. 2, pp. 217-225, Feb. 2010.

Authors


Sang Un Lee received the B. Sc. degree in avionics from the Korea Aerospace University in 1997. He received the M. Sc. and Ph. D. degrees in Computer Science from Gyeongsang National University, Korea, in 1997 and 2001, respectively.
He is currently Professor with the Department of Multimedia Science, Gangneung-Wonju National University, Korea. He is interested in software quality assurance and reliability modeling, software engineering, software project management, neural networks, and algorithm.