

다수 자원제약 하에서의 프로젝트 일정계획에 관한 연구

이정훈¹ · 김관술² · 문일경^{3*}

¹한국전자통신연구원 우정물류기술연구부 / ²한국수력원자력(주) 고리원자력본부 / ³부산대학교 산업공학과

A Study on Project Scheduling under Multiple Resource Constraints

Jeong-Hun Lee¹ · Pan-Sool Kim² · Il-Kyeong Moon³

¹Postal & Logistics Technology Research Department, ETRI, Daejeon 305-700

²Kori Nuclear Power Site, Korea Hydro & Nuclear Power Co. LTD., Busan 619-177

³Department of Industrial Engineering, Pusan National University, Busan 609-735

The PERT/CPM are considered as the base procedures for the most successful project scheduling programs. Unfortunately, it is not easy to apply these procedures to real-life projects. This is due to the fact that PERT/CPM assume an infinite number of resources for each activity in project networks. Obviously, the completion time under no constraints is less than when constraints are imposed. One way of approaching this problem is to use heuristic solution techniques. In this paper, we present three heuristics; MRU (Maximum Resource Use) rule, STU (Shortest Time Use) rule, MRUP (Max Resource Use and Period) rule for allocating resources to activities of projects under multiple resource constraints. Comparisons of the project durations show that these heuristic rules are superior to AG3 rule that has been widely used in practice (Elsayed and Boucher, 1994).

Keywords: Project Scheduling, Resource Constraint, Heuristic

1. 서론

프로젝트의 일정계획 및 통제를 위한 관리기법으로 개발된 PERT(Program Evaluation and Review Technique)와 CPM(Critical Path Method)은 일반적으로 산업체에서 널리 쓰이고 있는 기법이다. PERT/CPM 기법은 프로젝트 일정계획의 어느 순간에 가용할 수 있는 자원이 무한하다고 가정한다. 그러나 현실은 자원의 제약이 항상 따르기 마련이며, 프로젝트 관리자는 자원제약 하에서 프로젝트의 완료시간을 단축시키려고 노력하고 있다.

프로젝트의 활동에서 어떤 종류의 자원제약은 프로젝트의 완료시간을 증가시키게 된다. 기존의 PERT/CPM 기법은 전 활동의 완료시간이 지나면 바로 후 활동을 진행시키면 되었지만

자원의 제약이 있으면 아무리 전 활동의 완료시간이 끝났다 하더라도 자원의 양이 충분하지 못하면 후 활동으로 넘어갈 수 없기 때문이다. 즉, 각 활동(activity)에서 자원의 제약이 있으면 자원의 제약이 없을 때 보다 최소한 완료시간이 같거나 늦어지는 것은 명백하다. 자원의 제약이 있는 일정계획 문제에서 최적의 해법을 얻는 것은 어렵기 때문에 발견적 기법(heuristic technique)을 사용하여 문제를 해결할 필요가 있다. 자원의 제약이 있는 프로젝트의 일정 계획에서 대부분의 발견적 기법절차의 기본전제는 각 활동들의 순서를 정하기 위해 우선 규칙(priority rule)을 사용한다는 것이다. 그리고 활동들의 순서를 정한 일정계획은 자원제약을 위배하지 않고 일정계획 기간이 가능한 짧게 되어야 한다.

단일 자원제약 하에서 활동들의 순서를 정하기 위하여 ROT

이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

* 연락저자 : 문일경 교수, 609-735 부산광역시 금정구 장전2동 산30번지 부산대학교 산업공학과, Tel : 051-510-2451, Fax : 051-512-7603,

E-mail : ikmoon@pusan.ac.kr

2010년 7월 14일 접수; 2010년 9월 7일 수정본 접수; 2010년 9월 10일 게재 확정.

(Resources Over Time)(Elsayed and Boucher, 1994), ACTIM(Activity Time)(Bedworth and Bailey, 1994) 등의 우선규칙이 사용되고 있다. 다수 자원제약 하에서는 단일 자원제약 기법들을 확장하여 개발한 Bedworth and Bailey(1994)와 Elsayed and Boucher(1994)의 알고리즘과 breadth-first 기법을 활용한 Nazareth *et al.*(1999)의 알고리즘이 있다.

다수 프로젝트의 일정계획에 대한 연구에서는 동일한 자원을 2개 이상의 프로젝트에서 사용하는 경우로 단일 프로젝트에서 활용되는 알고리즘들을 확장하여 다양한 해법 개발에 대한 연구가 존재한다(Herroelen, 2005; Krüger and Scholl, 2010; Kumanan *et al.*, 2006). 그러나 본 연구에서는 원자력발전소의 단위공정을 대상으로 다수 자원제약이 있는 단일 프로젝트의 일정계획 문제를 다루므로 다수 프로젝트에 대한 부분은 제외하였다.

산업체, 특히 전력을 생산하는 발전소는 일일 24시간 연속 운전하기 때문에 일정기간동안 발전소를 일시정지하고 각 기기의 점검 및 정비를 수행하고 있다. 원자력발전소의 경우에는 이에 더하여 연료 교체작업과, 원전 비상상태 발생에 대비한 안전성 설비의 성능시험 및 점검을 수행하며, 이를 계획 예방정비(Planned Plant Overhaul)라 부른다. 계획예방정비는 많은 수의 단위공정과 이를 종합한 주공정으로 구분되며, 일정 계획은 주로 PERT/CPM 기법을 이용하여 수행하고 있으나 자재, 인력 등의 여러 제약요인이 있게 된다. 특히, 원자력발전소의 경우에는 외국산 자재를 많이 사용하고 있으므로 외국산 자재를 사용할 경우 적기에 자재가 입고되지 않는 문제 등이 그렇다. 또한 몇몇 공정의 경우에는 PERT/CPM 기법의 적용이 어려운 경우가 있다. 예를 들면 터빈과 발전기의 정비를 위한 일정에서 기기 분해 및 조립을 위한 천정 크레인의 사용 시 자원제약이 한 개이므로 기존의 단일자원제약의 발견적 기법을 사용하면 되지만, 비상상태 발생 시 안전주입(Safety Injection)을 하는 밸브의 성능시험의 경우처럼 시험을 위한 장비가 2종류이며, 장비가 한정되어 있고 밸브특성에 따라 장비를 설치할 인력의 수가 한정되어 있는 경우 단일 자원제약의 발견적 기법을 사용하기 어렵기 때문에 다수 자원제약의 알고리즘이 필요하게 된다. 실제로 원자력발전소에서는 이러한 단위공정에 어떤 특별한 프로젝트 기법을 사용하지는 않고 단지 주공정에 단위 공정이 따라 올 수 있도록 하는 방법을 사용하고 있다.

본 논문에서는 이러한 단위공정 관리의 문제점을 개선하기 위한 다수 자원제약 하에서의 일정계획 수립에 적용할 수 있는 새로운 발견적 기법을 제시하고자 한다. 본 논문의 구성은 다음과 같다. 제 2장에서는 단일 자원제약과 다수 자원제약 하에서 기존의 발견적 기법을 고찰한다. 제 3장에서는 다수 자원제약 하에서의 새로운 발견적 기법을 제안한다. 제 4장에서는 개발한 다수 자원제약 알고리즘과 기존 알고리즘을 수치실험을 통해 비교한다. 마지막으로 제 5장에서는 본 연구의 결론을 맺는다.

2. 기존 자원제약 하에서의 발견적 기법 고찰

자원제약 하에서의 발견적 기법들의 기본 가정과 사용된 기호들은 아래와 같다.

- 기본 가정
 - 1) 네트워크상의 각 활동의 소요기간과 자원소요는 알려져 있다.
 - 2) 프로젝트에서 자원 유형의 요구자원과 가능자원은 알려져 있다.
 - 3) 각 활동은 자원 할당이 끝난 후, 활동 중에 없어지는 자원량은 없다. 즉, 자원의 퇴화나 소멸은 없다.
 - 4) 미리 자원을 선점해 두지 못한다.
- 기호
 - 1) TNOW : 자원 할당을 고려하는 현재 시점
 - 2) Duration : 각 활동의 기간
 - 3) TSTART : 시간과 자원이 모두 충분하여 현 자원을 할당하는 실제 시점
 - 4) TFIN : 각 활동의 완료 시간 즉, TSTART와 Duration의 합
 - 5) Resource available : 할당 가능한 자원보유량
 - 6) Resource required : 해당 활동에 필요한 자원 요구량
 - 7) Activities allowed : 할당 가능한 활동의 종류

2.1 단일 자원제약

단일 자원제약 하에서 활동들의 우선순위를 정하기 위한 대표적인 발견적 기법으로는 ROT(Resources Over Time), ACTIM(Activity Time), ACTRES(Activity Resources), TIMRES(Time and Resources Combination), GENRES, TMROS의 6가지가 있다.

첫째, Elsayed(1982)는 단일 자원제약 하에서 프로젝트의 기간을 최소로 하는 활동들을 할당하기 위한 ROT(Resources Over Time)를 개발하였다. 네트워크상에서 단일 종류의 자원을 할당하기 위해 ROT 기준을 갖는데, 이는 네트워크상에서 현재 활동이 가장 큰 ROT 값을 갖는 주경로(critical path)를 찾아서, 필요한 총 ROT 값을 계산한다. 이때, ROT 값이란 각 활동에서 자원을 시간으로 나눈 것을 합한 것이다. 이 값이 큰 순서대로 활동을 정렬하여 이를 할당 우선순위로 사용한다.

둘째, Brooks(1963)는 단일 자원을 할당하는 방법으로 ACTIM(Activity Time)을 제안하였다. 이는 현재 활동의 시간을 최대한으로 하는 주경로를 찾고, 필요한 총소요시간을 계산한다. 이 값이 큰 순서대로 정렬하여 이를 할당 우선순위로 사용한다.

셋째, Bedworth(1973)는 ACTRES(Activity Resources) 기준을 기초로 하는 알고리즘을 개발하였는데, 이는 각 활동의 시간과 자원을 곱하고 현재 활동에서 가장 큰 ACTRES 값을 갖는 주경로를 찾아서, 필요한 총 ACTRES 값을 계산한다. 이 값이 큰 순서대로 정렬하여 이를 할당 우선순위로 사용한다.

넷째, Mason(1970)은 각 활동의 ACTIM과 ACTRES 값을 합하여 TIMRES(Time and Resources combination) 값을 구하여 이 값이 큰 순서대로 정렬하여 이를 할당 우선순위로 사용하는 알고리즘을 개발하였다.

다섯째, Whitehouse and Brown(1979)은 TIMRES 기법이 ACTIM과 ACTRES의 가중치를 같게 두는 단점을 보완하기 위해 서로 효과가 상충(trade-off) 되도록 서로 다른 가중치를 부여하는 알고리즘을 개발하였다. 즉, 한쪽에 가중치(W)를 주면, 다른 쪽은 $(1-W)$ 의 가중치 값을 부여하는 GENRES 기법을 제안하였다.

여섯째, Nasr(1983)과 Elsayed and Nasr(1986)는 ACROS와 ACTIM을 동시에 사용하기 위해 TMROS 기준을 개발하였다. 여기서 ACROS는 현재 활동에서 가장 큰 자원 값을 갖는 주경로를 찾아서 필요한 총 자원 값을 계산한다. 이 값이 큰 순서대로 정렬하여 이를 할당 우선순위로 사용한다. 이는 ACTIM 계산과 비슷하지만, 시간대신에 자원을 계산한다는 것이 다르다. TMROS는 ACTIM과 ACROS의 효과가 상충되도록 서로 다른 가중치를 부여하는 기법이다.

위에서 제시한 6가지 단일 자원제약에 대한 우선순위를 정하기 위한 발견적 기법들을 Khattab and Choobineh(1991)이 정리한 내용을 본 연구에 맞게 간단히 요약하면 <Table 1>과 같다.

Table 1. Summary of priority technique for single resource constraint

Priority rule	Reference
$ROT(m, n) = \max_k \sum_{i,j \in P_{mk}} \left\{ \frac{R_{ij}}{T_{ij}} \right\}$	Elsayed(1982)
$ACTIM(m, n) = \max_k \sum_{i,j \in P_{mk}} T_{ij}$	Brooks(1963)
$ACTRES(m, n) = \max_k \sum_{i,j \in P_{mk}} T_{ij} R_{ij}$	Bedworth(1973)
$TIMRES = (ACTRES) + (ACTIM)$	Mason(1970)
$GENRES = W \cdot (ACTRES) + (1 - W) \cdot (ACTIM)$ (단, $0 \leq W \leq 1$)	Whitehouse and Brown(1979)
$TMROS = W \cdot (ACROS) + (1 - W) \cdot (ACTIM)$ (단, $0 \leq W \leq 1$)	Nasr(1983) Elsayed and Nasr(1986)
$ACROS(m, n) = \max_k \sum_{i,j \in P_{mk}} R_{ij}$	

이때, R_{ij} 는 활동 ij 를 완성하는데 필요한 자원 요구량, T_{ij} 는 활동 ij 를 완성하는데 필요한 시간 요구량, P_{mk} 는 활동 mn 을 포함한 노드 m 에서 마지막 노드까지의 k 번째 직접 경로에 대한 활동의 집합을 나타낸다.

2.2 다수 자원제약

다수 자원제약 문제는 단일 자원제약의 문제보다 복잡하다. 기존에 연구되었던 다수 자원제약의 문제를 풀기 위한 기법들은 대부분 제 2.1절의 단일 자원제약 문제를 풀기 위한 기법을

응용하여 개발되었다. 대표적인 알고리즘으로 Bedworth and Bailey(1994)와 Elsayed and Boucher(1994)가 개발한 알고리즘을 들 수 있다.

Bedworth and Bailey(1994)는 ACTIM을 이용한 다수 자원제약이 있는 경우의 프로젝트 일정계획문제를 풀기 위한 알고리즘을 개발하였다. 이때 ACTIM을 이용하는 이유는 다수 자원제약이 있는 경우에도 시간 제약은 하나이기 때문에 한 개의 자원제약이 있는 경우와 같은 방법으로 사용할 수 있기 때문이다. Elsayed and Boucher(1994)는 복수 프로젝트와 복수 자원제약이 있는 문제를 단일 자원제약에서의 방법 중 하나인 TMROS의 확장 형태로 eq. ACROS와 ACTIM 사이에 가중치 (W)를 부여하여 할당 순서를 정하는 방법을 다음과 같은 식 (1)을 이용하여 제시하였다.

$$AG3 = W(eq. ACROS) + (1 - W) (ACTIM) \quad (1)$$

(단, $0 \leq W \leq 1$)

여기에서 eq. ACROS는 현재 활동에서 자원 유형별로 가능 자원과 모든 경로에 대한 총 요구자원을 나눈 값 중 가장 큰 값을 계산하여 할당하는 것으로 현재 활동에서 가장 큰 자원 값을 계산하여 할당하는 ACROS와 차이가 있다. 좀 더 구체적인 eq. ACROS의 계산 방법은 제 4장에서 예를 들어 설명하였다.

Bedworth and Bailey(1994)가 제시한 알고리즘은 시간 제약 중심의 알고리즘으로 자원을 충분히 고려하지 못한다. 또한 대부분의 경우 ACTIM보다 TMROS가 더 좋은 결과를 도출한다(Elsayed and Boucher, 1994). Nazareth *et al.*(1999)이 제안한 breadth-first 기법은 AG3 기법과 유사한 가중치(θ)를 사용하고 계산 과정이 더 복잡하기 때문에 본 논문에서는 Elsayed and Boucher(1994)가 제시한 알고리즘과 다음 장에서 제시하는 3가지 발견적 기법을 비교실험 한다.

3. 다수 자원제약 알고리즘의 개발

Elsayed and Boucher(1994)의 AG3 규칙은 가중치들마다 결과 값이 달라지기 때문에, 최적의 가중치를 정하기 어렵고, 시간이 많이 걸린다. 또한, 제 2.2절에서 언급한 2가지의 방법들은 미리 전체 순서를 정하고, 그에 따라 활동들을 할당하고 있다. 그러나 활동의 수가 많거나 자원 제약의 수가 많을 경우엔 이러한 방법들은 계산이 복잡해지고, 최적 종료일을 찾기가 매우 힘들다. 따라서 간단하면서도 즉시 최적에 가까운 종료일을 구할 수 있는 발견적 기법(heuristic technique)이 개발되어야 할 필요가 있다. 따라서 다수의 자원제약이 있는 경우에 사용할 수 있는 3가지 발견적 기법을 제시한다.

3.1 최대 자원 활용(MRU; Max Resource Use) 기법

이 기법은 활동의 우선순위를 정할 때 현재시점(TNOW)에

서 활동의 총 요구자원(Resource required)에 비해 가능자원(Resource available)이 얼마나 필요한지를 다음의 식 $\sum Resource\ required/Resource\ available$ 을 이용하여 각 자원 유형별로 비율을 구해 가장 큰 값이 나오는 자원을 가장 많이 필요로 하는 자원으로 정해서 어떤 자원 종류가 가장 많이 필요한지를 구해, 이를 최대로 할당하고자 하는 자원 중심적 발견적 기법이다. MRU 기법에 대한 상세한 단계는 다음과 같다.

단계 1 : 각 활동에 필요한 각 자원 유형 (resource type)의 요구 자원이 각 자원 유형의 가능자원보다 작은지 확인한다. 만약 작다면 단계 2로 간다.

단계 2 : 초기 가능자원으로 할당 가능한 활동들을 정해서 할당가능 활동 (activity allowed)을 결정한다.

단계 3 : 할당가능 활동에 1개의 활동이 존재한다면, 이를 할당하고, 2개 이상의 활동들이 할당가능 활동에 있을 때는 각 자원 유형에 대해 요구자원의 총합을 구한다.

단계 4 : 각 활동의 $\frac{\sum Resource\ required}{Resource\ available}$ 값을 구한 뒤, 큰 값의 자원 유형에 우선순위를 부여한다.

$\frac{\sum Resource\ required}{Resource\ available}$ 값이 같아서, 우선순위를 부여할 수 없을 경우엔, eq. ACROS에 따라 활동을 할당

하고, 단계 6으로 간다.

단계 5 : 만약, 할당될 자원의 수가 같고 다른 자원 유형들도 할당 가능하다면 가능한 한 많은 활동이 선택되도록 정한다(예를 들어 가능자원이 4라고 할 때, 2개의 활동의 요구자원이 합해서 4이고 1개의 활동의 요구자원이 4라고 가정하면, 한 개의 활동이라도 더 수행되는 앞의 2개의 활동을 먼저 선택한다).

단계 6 : 할당가능한 자원이 없으면, 현재시점 (TNOW)에서 가장 가까운 활동 완료시간 (TFIN)을 현재시점으로 넣고, 잡고 있던 자원들을 다시 회수하고, 단계 3으로 간다.

단계 7 : 활동들의 할당이 끝나면, 가장 긴 활동 완료시간을 최적의 작업 완료시간으로 선택한다.

3.2 최단 시간 활용(STU; Shortest Time Use) 기법

이 기법은 다수 자원제약 하에서 시간이 최소인 것을 우선순위로 두는 것으로 Bedworth and Bailey(1994)의 ACTIM 규칙을 이용한 다수 자원제약 알고리즘을 응용한 것이다. 활동의 할당 순서를 ACTIM 값이 아닌 시간이 짧게 걸리는 것부터 우선 할당하도록 하고 만약 동일한 시간이 걸리면, ACTIM 규칙을 사용하도록 하고자 하는 시간 중심적 발견적 기법이다. 이는 자원을 할당하고 난 후, 자원이 최소기간동안 머무르는 활동을 먼저 할당하여 자원은 많이 필요로 하진 않지만 기간이 많이 걸리는 활동으로 인해 전체 프로젝트 완료시간이 길어질

을 방지할 수 있다. 기존 연구에서 시간 중심적 알고리즘으로 ACTIM 규칙을 많이 이용하는데 ACTIM 규칙은 후속 활동이 가장 길게 소요되는 경로를 따라 가는데 비해 이 방법은 현재 활동만 고려하여 가장 짧게 소요되는 경로를 선택한다는 점이 특징이다. STU 기법에 대한 상세한 단계는 다음과 같다.

단계 1 : 각 활동에 필요한 각 자원 유형의 요구자원이 각 자원 유형의 가능자원보다 작은지 확인한다. 만약 작다면 단계 2로 간다.

단계 2 : 초기 가능자원으로 할당 가능한 활동들을 정해서 할당가능 활동을 결정한다.

단계 3 : 할당가능 활동에 1개의 활동이 존재한다면, 이를 할당하고, 2개 이상의 활동들이 할당가능 활동에 있을 때는 할당가능 활동 중 최단 시간이 걸리는 것을 선택한다.

단계 4 : 만약 할당 가능한 활동의 최단 시간이 같다면, ACTIM 값에 따라 활동을 할당한다.

단계 5 : 할당가능한 자원이 없으면, 현재시점(TNOW)에서 가장 가까운 활동 완료시간(TFIN)을 현재시점으로 넣고, 잡고 있던 자원들을 다시 회수하고, 단계 3으로 간다.

단계 6 : 활동들의 할당이 끝나면, 가장 긴 활동 완료시간을 최적의 작업 완료시간으로 선택한다.

3.3 최대 자원 활용 및 자원/기간(MRUP; Max Resource Use and Period) 기법

MRUP 기법은 MRU 기법에 각 활동의 기간을 고려하는 방법으로 MRU 기법에서 우선순위가 높은 자원 유형에 각 활동에 대한 자원/기간(resource/duration)의 비율을 구하여 이 값이 큰 순서대로 할당한다. 이는 MRU 기법이 완전 자원 중심적 기법이라면, 이를 시간요소를 고려하여 시간이 적게 걸리면서도 자원이 많이 할당되는 활동을 찾도록 해준다.

기존 연구에서도 Elsayed and Boucher(1994)가 AG3 규칙을 이용하여 시간을 이용한 ACTIM 값과 자원을 이용한 eq. ACROS 값을 가중치(W)를 이용하여 서로 상충(trade-off) 되도록 했지만 앞에서 언급한 것과 같이 최적의 가중치를 얻는 것이 어렵다. 이런 점을 보완하기 위해 본 연구에서는 자원 중심적 MRU 기법과 각 활동의 기간에 소요되는 요구자원을 조사하여 각 단위 시간당 소요되는 자원이 최대가 되도록 활동을 할당해 준다. MRUP 기법에 대한 상세한 단계는 다음과 같다.

단계 1 : 각 활동에 필요한 각 자원 유형의 요구자원이 각 자원 유형의 가능자원보다 작은지 확인한다. 만약 작다면 단계 2로 간다.

단계 2 : 초기 가능자원으로 할당 가능한 활동들을 정해서 할당가능 활동을 결정한다.

단계 3 : 할당가능 활동에 1개의 활동이 존재한다면, 이를 할당하고, 2개 이상의 활동들이 할당가능 활동에 있을 때

는 각 자원 유형에 대해 요구자원의 총합을 구한다.

단계 4 : 각 활동의 $\frac{\sum Resource\ required}{Resource\ available}$ 을 구한 뒤, 큰 값의 자원 유형을 선택한다.

단계 5 : 이 자원 유형에 대한 각 활동들의 자원/기간을 구하고, 큰 순서대로 활동을 할당한다.

단계 6 : 만약, 할당될 자원의 수가 같고 다른 자원 유형들도 할당 가능하다면 가능한 한 많은 활동이 선택되도록 정한다(예를 들어 가능자원이 4라고 할 때, 2개의 활동의 요구자원이 합해서 4이고 1개의 활동의 요구자원이 4라고 가정하면, 한 개의 활동이라도 더 수행되는 앞의 2개의 활동을 먼저 선택한다).

단계 7 : 할당가능한 자원이 없으면, 현재시점(TNOW)에서 가장 가까운 활동 완료시간(TFIN)을 현재시점으로 넣고, 잡고 있던 자원들을 다시 회수하고, 단계 3으로 간다.

단계 8 : 활동들의 할당이 끝나면, 가장 긴 활동 완료시간을 최적의 작업 완료시간으로 선택한다.

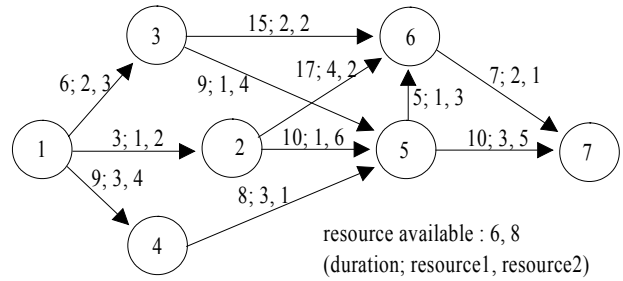


Figure 1. Example of multiple resource network

먼저 Elsayed and Boucher(1994)의 AG3 규칙을 이용한 활동 완료시간을 구하기 위해 다음과 같은 방법으로 eq. ACROS와 ACTIM 값을 구하여 활동의 우선순위를 정한다.

각 자원에 대해 A_r/S_r 를 계산하면, resource 1 : $A_1/S_1 = 6/23 = 0.26$, resource 2 : $A_2/S_2 = 8/33 = 0.24$ 와 같이 각 활동의 자원 값을 계산하면 아래와 같다(단, A_r 은 자원 유형 r 의 가능자원, S_r 은 네트워크상의 모든 경로에 대한 자원 유형 r 의 총 요구자원).

$$\text{activity 1-2} = \frac{1}{0.26} + \frac{2}{0.24} = 12.18$$

$$\text{activity 1-3} = \frac{2}{0.26} + \frac{3}{0.24} = 20.19$$

$$\text{activity 1-4} = \frac{3}{0.26} + \frac{4}{0.24} = 28.21$$

$$\text{activity 2-5} = \frac{1}{0.26} + \frac{6}{0.24} = 28.85$$

$$\text{activity 2-6} = \frac{4}{0.26} + \frac{2}{0.24} = 23.72$$

$$\text{activity 3-5} = \frac{1}{0.26} + \frac{4}{0.24} = 20.51$$

4. 수치실험 및 비교실험

4.1 수치실험

Elsayed and Boucher(1994)에서 발췌한 예제인 <Figure 1>의 network상에서 AG3 규칙을 이용하여 활동 완료시간을 구해보고 새로 개발한 기법인 MRU, STU, MRUP 기법의 활동 완료시간을 구해보면 <Table 2>와 같으며, 구하는 방법은 다음과 같다.

Table 2. Comparison results of completion time among the heuristics

Heuristic Time	MRU	STU	MRUP	AG3				
				$W = 0.0$	$W = 0.3$	$W = 0.5$	$W = 0.7$	$W = 1.0$
Completion time(day)	48	54	54	52	56	54	54	51

Table 3. eq. ACROS and ACTIM values for example

Activity	eq. ACROS		Activity	ACTIM	
	Value	0~100 Scale		Value	0~100 Scale
1-2, 2-5, 5-7	73.40	96.21	1-2, 2-6, 6-7	27	93.10
1-3, 3-5, 5-7	73.07	95.78	1-3, 3-6, 6-7	28	96.55
1-4, 4-5, 5-7	76.29	100.00	1-4, 4-5, 5-6, 6-7	29	100.00
2-5, 5-7	61.22	80.25	2-5, 5-6, 6-7	22	75.86
2-6, 6-7	35.58	46.64	2-6, 6-7	24	82.76
3-5, 5-7	52.88	69.31	3-5, 5-6, 5-7	21	72.41
3-6, 6-7	27.89	36.56	3-6, 6-7	22	75.86
4-5, 5-7	48.08	63.02	4-5, 5-6, 5-7	20	68.97
5-6, 6-7	28.21	36.98	5-6, 6-7	12	41.38
5-7	32.37	42.43	5-7	10	34.48
6-7	11.86	15.55	6-6	7	24.14

$$\text{activity 3-6} = \frac{2}{0.26} + \frac{2}{0.24} = 16.03$$

$$\text{activity 4-5} = \frac{3}{0.26} + \frac{1}{0.24} = 15.71$$

$$\text{activity 5-6} = \frac{1}{0.26} + \frac{3}{0.24} = 16.35$$

$$\text{activity 5-7} = \frac{3}{0.26} + \frac{5}{0.24} = 32.37$$

$$\text{activity 6-7} = \frac{2}{0.26} + \frac{1}{0.24} = 11.86$$

현재 활동의 후행 활동에 해당되는 A_r/S_r 의 값을 모두 합산

하여 eq. ACROS와 ACTIM 값을 계산하면 <Table 3>과 같으며 <Table 3>에 서로 다른 가중치를 사용하여 활동들의 값을 구하면 <Table 4>와 같다.

새로 개발한 알고리즘인 MRU 기법을 사용하여 활동 완료 시간을 <Table 5>에 따라 계산하면 활동 완료시간이 48일이 됨을 알 수 있다.

<Table 5>를 보면 TNOW = 0에서 가능자원의 첫 번째 자원 유형은 6이고, 두 번째 자원 유형은 8에서 할당 가능 활동은 1-2, 1-3, 1-4가 되는데, 할당순서를 정하기 위해 MRU 기법에 따라 $\sum \text{Resource required} / \text{Resource available}$ 값은 첫 번째 자원 유형은 6/6(= 1), 두 번째 자원 유형은 9/8(1.13)이다. 두 번째 자

Table 4. AG3 value according to the weight

Activity	W = 0.0	W = 0.3	W = 0.5	W = 0.7	W = 1.0
1-2	93.10	94.03	94.66	95.28	96.21
1-3	96.55	96.32	96.17	96.01	95.78
1-4	100.00	100.00	100.00	100.00	100.00
2-5	75.86	77.18	78.06	78.93	80.25
2-6	82.76	71.92	64.70	58.38	46.64
3-5	72.41	71.48	70.86	70.24	69.31
3-6	75.86	64.07	56.21	48.35	36.56
4-5	68.97	67.19	66.00	64.81	63.02
5-6	41.38	40.06	39.18	38.30	36.98
5-7	34.48	36.87	38.46	40.05	42.43
6-7	24.14	21.56	19.85	21.05	15.55

Table 5. Computational result of MRU heuristic for example

TNOW	Activity	Duration	TSTART	TFIN	Resource Available		Activities allowed	Remarks
0	-	-	-	-	6	8	1-2, 1-3, 1-4	(6, 9)6/6, 9/8
0	1-4	9	0	9	3	4		
0	1-3	6	0	6	1	1		
6	-	-	-	-	3	4	1-2, 3-5, 3-6	(4, 8)4/3, 8/4
6	1-2	3	6	9	2	2		
6	3-6	15	6	21	0	0		
9	-	-	-	-	4	6	2-5, 2-6, 3-5, 4-5	(9, 13)9/4, 13/6
9	4-5	8	9	17	1	5		
9	3-5	9	9	18	0	1		
17	-	-	-	-	3	2		
18	-	-	-	-	4	6	2-5, 2-6	(5, 8)5/4, 8/6
18	2-5	10	18	28	3	0		
21	-	-	-	-	5	2	2-6	
21	2-6	17	21	38	1	0		
28	-	-	-	-	2	6	5-6	
28	5-6	5	28	33	1	3		
33	-	-	-	-	2	6		
38	-	-	-	-	6	8	5-7, 6-7	(5, 6)5/6, 6/8
38	5-7	10	38	48	3	3		
38	6-7	7	38	45	1	2		

원 유형이 첫 번째 자원 유형보다 크므로 두 번째 자원 유형에 우선순위를 부여하면 1-4, 1-3, 1-2의 순이므로 먼저 활동 1-4를 할당하면 활동 1-4의 요구자원은 첫 번째 자원 유형이 3이고, 두 번째 자원 유형이 4이므로 남은 자원은 첫 번째 자원 유형은 3이고, 두 번째 자원 유형은 4가 되는데 활동 1-3의 요구자원이 첫 번째 자원 유형은 1이고, 두 번째 자원 유형이 2이므로 활동 1-3을 할당한다. 그러면 남은 자원이 첫 번째 자원 유형이 1이고, 두 번째 자원 유형도 1이므로 할당할 활동들이 없어 활동 1-3이 종료될 때까지 기다린다. 다음에 TNOW = 6에서 가능자원은 활동 1-3에서 할당했던 자원을 회수하면 첫 번째 자원 유형은 3이고, 두 번째 자원 유형은 4이다. 할당 가능활동은 1-2, 3-5, 3-6이 되고 할당순서를 정하기 위해 1-2와 3-6의 요구자원을 합하면 첫 번째 자원 유형이 3이고, 두 번째 자원 유형이 4이므로 MRU 기법 단계 5에 따르면 할당될 자원의 수가 같고 다른 자원 유형들도 할당가능 하다면 가능한 한 많은 활동들이 선택되도록 정하므로 활동 1-2와 3-6을 할당한다. 위와 같은 방법으로 진행하여 모든 활동들의 할당이 끝나면, 가장 긴 TFIN인 48일을 최적의 작업 완료시간으로 선택한다.

다음으로 두 번째 STU 기법을 예제에 적용하여 <Table 6>에 따라 계산하면 활동 완료시간이 54일이 됨을 알 수 있다.

<Table 6>을 보면 TNOW = 0에서 가능자원은 첫 번째 자원 유형은 6이고, 두 번째 자원 유형은 8에서 할당 가능 활동은 1-2, 1-3, 1-4가 되는데, 할당순서를 정하기 위해 STU 기법에 따

라 활동 1-2, 1-3, 1-4 중 최단시간인 1-2를 할당하면 남은 자원은 첫 번째 자원 유형은 5이고, 두 번째 자원 유형은 6이므로 그 다음 최단 시간인 활동 1-3을 할당하면 남은 자원은 첫 번째 자원 유형은 3이고, 두 번째 자원 유형도 3이다. 활동 1-4의 요구자원은 첫 번째 자원 유형이 3이고, 두 번째 자원 유형은 4이므로 할당 할 수가 없어 활동 1-2가 종료될 때까지 기다린다. 다음에 TNOW = 3에서 가능자원은 활동 1-2에서 할당했던 자원을 회수하면 첫 번째 자원 유형은 4이고, 두 번째 자원 유형은 5이다. 이때 할당가능 활동은 1-4, 2-6이 되는데 2-5가 할당 가능하지 못한 이유는 활동 2-5의 요구자원이 첫 번째 자원 유형은 1이고, 두 번째 자원 유형이 6이므로 두 번째 자원 유형이 가능자원 5를 초과하기 때문이다. 활동 1-4, 2-6에서 최단시간인 1-4를 할당한다. 위와 같은 방법으로 진행하여 모든 활동들의 할당이 끝나면 가장 긴 TFIN인 54일을 최적의 작업완료 시간으로 선택한다.

마지막으로 MRUP 기법을 사용하여 활동 완료시간을 <Table 7>에 따라 계산하면 활동 완료시간이 54일이 됨을 알 수 있다.

<Table 7>을 보면 TNOW = 0에서 가능자원은 첫 번째 자원 유형은 6이고, 두 번째 자원 유형은 8에서 할당 가능 활동은 1-2, 1-3, 1-4가 되는데, 할당순서를 정하기 위해 최대자원 활용 및 자원/기간 알고리즘에 따라 $\sum Resource\ required / Resource\ available$ 은 첫 번째 자원 유형은 $6/6(= 1)$, 두 번째 자원 유형은 $9/8(1.13)$ 이다. 두 번째 자원 유형이 첫 번째 자원 유형보다 크므로

Table 6. Computational result of STU heuristic for example

TNOW	Activity	Duration	TSTART	TFIN	Resource Available	Activities allowed	
0	-	-	-	-	6	8	1-2, 1-3, 1-4
0	1-2	3	0	3	5	6	
0	1-3	6	0	6	3	3	
3	-	-	-	-	4	5	1-4, 2-6
3	1-4	9	3	12	1	1	
6	-	-	-	-	3	4	2-5, 3-6
6	3-5	9	6	15	2	0	
12	-	-	-	-	5	4	2-6, 3-6, 4-5
12	4-5	8	12	20	2	3	
12	3-6	15	12	27	0	1	
15	-	-	-	-	1	5	
20	-	-	-	-	4	6	2-5, 2-6
20	2-5	10	20	30	3	0	
27	-	-	-	-	5	2	2-6
27	2-6	17	27	44	1	0	
30	-	-	-	-	2	6	5-6, 5-7
30	5-6	5	30	35	1	3	
35	-	-	-	-	2	6	
44	-	-	-	-	6	8	5-7, 6-7
44	6-7	7	44	51	4	7	
44	5-7	10	44	54	1	2	

Table 7. Computational result of MRUP heuristic for example

TNOW	Activity	Duration	TSTART	TFIN	Resource	Available	Activities allowed	Remarks
0	-	-	-	-	6	8	1-2(2/3), 1-3(3/6), 1-4(4/9)	(6, 9)6/6, 9/8
0	1-2	3	0	3	5	6		
0	1-3	6	0	6	3	3		
3	-	-	-	-	4	5	1-4(3/9), 2-6(4/17)	(7, 6)7/4, 6/5
3	1-4	9	3	12	1	1		
6	-	-	-	-	3	4	3-5(4/9), 3-6(2/15)	(3, 6)3/3, 6/4
6	3-5	9	6	15	2	0		
12	-	-	-	-	5	4	2-6(4/17), 3-6(2/15), 4-5(3/8)	(9, 5)9/5, 5/4
12	4-5	8	12	20	2	3		
12	3-6	15	12	27	0	1		
15	-	-	-	-	1	5		
20	-	-	-	-	4	6	2-5(6/10), 2-6(2/17)	(5, 8)5/4, 8/6
20	2-5	10	20	30	3	0		
27	-	-	-	-	5	2	2-6	
27	2-6	17	27	44	1	0		
30	-	-	-	-	2	6	5-6	
30	5-6	5	30	35	1	3		
35	-	-	-	-	2	6		
44	-	-	-	-	6	8	5-7(3/10), 6-7(2/7)	(5, 6)5/6, 6/8
44	5-7	10	44	54	3	3		
44	6-7	7	44	51	1	2		

두 번째 자원 유형의 자원/기간을 구하면 활동 1-2는 2/3(=0.67), 활동 1-3은 3/6(=0.5), 활동 1-4는 4/9(=0.44)이다. 이 중에서 활동 1-2의 2/3(=0.67)이 제일 크므로 먼저 활동 1-2를 할당하면 활동 1-2의 요구자원은 첫 번째 자원 유형이 1이고, 두 번째 자원 유형이 2이므로 남은 자원은 첫 번째 자원 유형은 5이고, 두 번째 자원 유형이 6이 되어 활동 1-3을 할당한다. 그러면 남은 자원은 첫 번째 자원 유형은 3이고, 두 번째 자원 유형도 3이므로 할당 할 활동들이 없으므로 활동 1-2가 종료될 때 까지 기다린다. 다음에 TNOW = 3에서 위와 같은 방법으로 진행하여 모

든 활동들의 할당이 끝나면 가장 긴 TFIN인 54일을 최적의 작업완료시간으로 선택한다.

4.2 비교실험

본 연구에서 제시한 MRU, STU, MRUP 기법을 Elsayed and Boucher(1994)의 AG3 규칙과 임의로 추출한 10 문제에 대해 활동 완료시간을 구한 것이 <Table 8>에 나타나 있으며 실험결과 는 다음과 같이 나타났다.

Table 8. Comparison results of examples

Examples	MRU	STU	MRUP	AG3				
				W = 0.0	W = 0.3	W = 0.5	W = 0.7	W = 1.0
Example 1	48	54	48	55	59	54	54	58
Example 2	50	53	50	55	55	54	55	55
Example 3	56	58	56	58	58	59	58	59
Example 4	82	87	82	87	82	88	88	88
Example 5	33	35	39	36	35	35	35	35
Example 6	30	31	30	30	30	32	29	30
Example 7	50	53	54	55	55	55	56	55
Example 8	55	52	52	56	54	56	56	56
Example 9	46	48	48	44	44	47	48	48
Example 10	37	37	37	34	37	34	37	37

4.2.1 MRU 기법과 결과 비교

MRU 기법과 AG3 규칙의 값을 서로 비교해보면 AG3 규칙의 값을 1로 두었을 때 각 가중치에 따른 MRU 기법의 결과 값을 AG3 규칙의 결과 값으로 나누면 <Table 9>와 같다.

Table 9. Comparison results between MRU and AG3

Examples	AG3				
	$W = 0.0$	$W = 0.3$	$W = 0.5$	$W = 0.7$	$W = 1.0$
Example 1	0.873	0.814	0.889	0.889	0.827
Example 2	0.909	0.909	0.926	0.909	0.909
Example 3	0.966	0.966	0.949	0.966	0.949
Example 4	0.943	1.000	0.932	0.932	0.932
Example 5	0.917	0.943	0.943	0.943	0.943
Example 6	1.000	1.000	0.938	1.034	1.000
Example 7	0.909	0.909	0.909	0.893	0.909
Example 8	0.982	1.019	0.982	0.982	0.982
Example 9	1.045	1.045	0.979	0.958	0.958
Example 10	1.088	1.000	1.088	1.000	1.000

MRU 기법의 효율을 보면 MRU 기법의 결과 값을 AG3 규칙의 결과 값으로 나눈 값의 평균값은 0.954로 평균비율이 1보다 적은 것으로 나타났으며 최대값은 1.088, 최소값은 0.814로 MRU 기법이 AG3 규칙보다 더 우수하다는 것을 알 수 있다. 또한 AG3 규칙의 가중치 중에서 example 1에서 10까지의 활동 완료시간이 가장 짧은 것 한 개씩을 추출해내어 합해 나눈 것과 MRU 기법의 결과 값을 비교하면, $(0.889+0.926+0.966+1.000+0.943+1.000+0.909+1.019+1.045+1.088)/10 = 0.978$ 로 평균비율이 1보다 적으므로 MRU 기법이 AG3 규칙보다 더 우수하다는 것을 알 수 있다.

4.2.2 STU 기법과 결과 비교

STU 기법과 AG3 규칙의 값을 서로 비교해보면 AG3 규칙의 값을 1로 두었을 때 각 가중치에 따른 STU 기법의 결과 값을 AG3 규칙의 결과 값으로 나누면 <Table 10>과 같다.

Table 10. Comparison results between STU and AG3

Examples	AG3				
	$W = 0.0$	$W = 0.3$	$W = 0.5$	$W = 0.7$	$W = 1.0$
Example 1	0.982	0.915	1.000	1.000	0.931
Example 2	0.964	0.964	0.981	0.964	0.964
Example 3	1.000	1.000	0.983	1.000	0.983
Example 4	1.000	1.061	0.989	0.989	0.989
Example 5	0.972	1.000	1.000	1.000	1.000
Example 6	1.033	1.033	0.969	1.069	1.033
Example 7	0.964	0.964	0.964	0.946	0.964
Example 8	0.929	0.963	0.929	0.929	0.929
Example 9	1.091	1.091	1.021	1.000	1.000
Example 10	1.088	1.000	1.088	1.000	1.000

STU 기법의 효율을 보면 STU 기법의 결과 값을 AG3 규칙의 결과 값으로 나눈 값의 평균값은 0.991로 평균비율이 1보다 적은 것으로 나타났으며 최대값은 1.091, 최소값은 0.915로 STU 기법이 AG3 규칙보다 더 우수하다는 것을 알 수 있다. 그리고 AG3 규칙의 가중치 중에서 example 1에서 10까지의 활동 완료시간이 가장 짧은 것 한 개씩을 추출해내어 합해 나눈 것과 STU 기법과 결과 값을 비교하면, $(1.000+0.981+1.000+1.061+1.000+1.069+0.964+0.963+1.091+1.088)/10 = 1.022$ 로 평균비율이 1에 근접하여 최단시간 활용규칙이 AG3규칙과 비슷하다는 것을 알 수 있다.

4.2.3 MRUP 기법과 결과 비교

MRUP 기법과 AG3 규칙의 값을 서로 비교해보면 AG3 규칙의 값을 1로 두었을 때 각 가중치에 따른 MRUP 기법의 결과 값을 AG3 규칙의 결과 값으로 나누면 <Table 11>과 같다.

Table 11. Comparison results between MRUP and AG3

Examples	AG3				
	$W = 0.0$	$W = 0.3$	$W = 0.5$	$W = 0.7$	$W = 1.0$
Example 1	0.873	0.814	0.889	0.889	0.827
Example 2	0.909	0.909	0.926	0.909	0.909
Example 3	0.966	0.966	0.949	0.966	0.949
Example 4	0.943	1.000	0.932	0.932	0.932
Example 5	1.083	1.114	1.114	1.114	1.114
Example 6	1.000	1.000	0.938	1.034	1.000
Example 7	0.982	0.982	0.982	0.964	0.982
Example 8	0.929	0.963	0.929	0.929	0.929
Example 9	1.091	1.091	1.021	1.000	1.000
Example 10	1.088	1.000	1.088	1.000	1.000

MRUP 기법의 효율을 보면 MRUP 기법의 결과 값을 AG3 규칙의 결과 값으로 나눈 값의 평균값은 0.977로 평균비율이 1보다 적은 것으로 나타났으며 최대값은 1.114, 최소값은 0.814로 MRUP 기법이 AG3 규칙보다 더 우수하다는 것을 알 수 있다. 그리고 AG3 규칙의 가중치 중에서 example 1에서 10까지의 활동 완료시간이 가장 짧은 것 한 개씩을 추출해내어 합해 나눈 것과 MRUP 기법의 결과 값을 비교하면, $(0.889+0.926+0.966+1.000+1.114+1.000+0.982+0.963+1.091+1.088)/10 = 1.002$ 로 평균비율이 1에 매우 근접한 것으로 보아 MRUP 기법이 AG3 규칙과 거의 비슷하다는 것을 알 수 있다.

4.2.4 현장 적용 예

원자력 발전소에서 원자로 냉각재배관 등의 파단으로 인한 냉각재 상실사고와 같은 사고가 발생했을 때 원자로 노심에 부반응도를 주기 위해 봉산수를 주입한다. 봉산수를 주입하는 밸브들은 높은 신뢰성을 가지고 있어야 하므로 정적인 상태와 동적인 상태에서 밸브들의 안전성을 평가하여야 한다. <Figure

2>는 주요한 6개 밸브에 대한 network이다. 활동 1→2는 HV 020 정적시험, 1→3은 HV023 정적시험, 1→4는 HV024 정적 시험, 2→5는 HV020 동적시험, 3→5는 HV023 동적시험, 3→6은 HV018 정적시험, 4→7은 HV024 동적시험, 5→9는 HV223 동적시험, 6→8은 HV018 동적시험, 7→8은 HV036 정적시험, 8→9는 HV036 동적시험을 나타낸다.

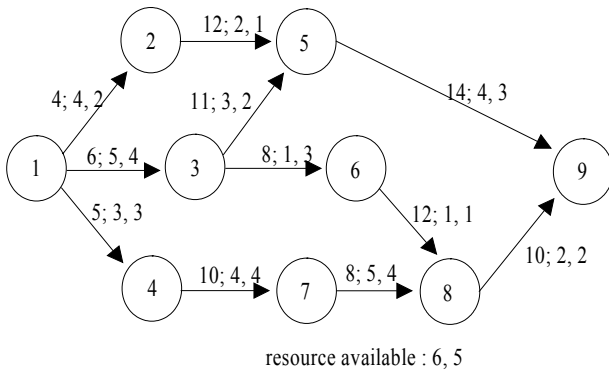


Figure 2. Network for the valve stability test

<Table 12>는 기존 AG3 규칙과 제안한 3가지 기법의 결과를 비교한 것으로 STU 기법의 완료시간은 59시간, AG3 규칙의 완료시간은 63시간임을 알 수 있다. 따라서 본 연구에서 제안한 기법이 기존 AG3 규칙보다 짧은 시간에 우수한 결과 값을 찾는다라는 것을 알 수 있다. 계산 시간 측면에서 AG3 규칙은 가중치(W) 값을 변경시켜가면서 최소 완료시간을 찾아야 하는 반면 본 연구에서 제시한 방법들은 바로 최소 완료시간을 찾을 수 있기 때문에 AG3 규칙보다 짧은 시간이 소요된다는 것을 직관적으로 알 수 있다.

만약 가능자원 1을 6에서 7로 증가시켰을 때의 작업완료 시간을 비교하면 <Table 13>과 같이 제안한 기법의 완료시간은 59시간, AG3 규칙도 59시간임을 알 수 있다. 또한 가능자원 2를 5에서 6으로 증가시켰을 때의 작업완료 시간을 비교하면

제안한 기법의 완료시간은 58시간, AG3 규칙은 59시간임을 알 수 있다.

5. 결론

과거 미국 캘리포니아주 전력부족 사태에서 보았듯이 전력의 공급부족은 소비자에게 직접적으로 전가되어 많은 불편을 초래할 뿐만 아니라 산업시설 전반에도 악영향을 미친다는 것을 알 수 있다. 특히, 우리나라의 경우에는 지정학적인 여건상 외국으로부터의 전기 수입이 불가능하므로 만일 이러한 사태가 우리에게 발생한다면 치명적인 약점으로 작용할 수 있다. 이러한 사태에 대비하기 위하여 정부와 산업계에서는 이 사태가 발생한다면 어떻게 될 것인지, 이를 대비하기 위한 대응책은 무엇인지에 대하여 여러 각도로 분석하고 있다. 그러나 이러한 분석은 주로 원론적, 총론적인 시각에서만 분석될 뿐 구체적인 분석은 전무하다고 할 수 있다. 따라서 본 논문에서는 자원의 효율적 이용이라는 측면과, 부족한 전기의 발생 가능성을 일정부분 예방한다는 측면에서 다수 자원제약 하에서의 프로젝트 일정계획을 위한 새로운 알고리즘을 개발, 원자력발전소 단위공정에 적용하여 계획예방정비기간을 단축하고 발전소 효율을 향상시키는 방법을 제시하고 있다. 이를 위하여 본 논문에서는 다수 자원제약 하에서의 알고리즘인 Elsayed and Boucher(1994)의 AG3 규칙보다 짧은 시간에 대부분 문제에서 좋은 결과 값을 찾을 수 있는 3가지 발전적 기법을 개발하였다. 또한 수치실험과 기존 규칙과의 비교연구를 통해서 제안한 기법이 기존의 기법보다 대부분 더 우수하다는 것을 입증하였다. 그동안 원자력발전소 계획 예방정비 시 많은 단위공정의 프로젝트가 개략적으로 수행되어 온 것을 본 논문을 통하여 다수 자원제약 하에서의 프로젝트 일정계획을 위한 새로운 알고리즘을 개발하여 적용함으로써 원자력발전소의 효율 향상과 자원의 효율적 이용에 효과를 얻을 수 있을 것이다.

Table 12. Comparison results of completion time among the heuristics

Time	Heuristic	MRU	STU	MRUP	AG3		
					W = 0.0	W = 0.5	W = 1.0
Completion time(day)		65	59	65	65	73	63

Table 13. Comparison results according to the increase one unit

Time	Heuristic	MRU	STU	MRUP	AG3		
					W = 0.0	W = 0.5	W = 1.0
Completion time(day)	Resource 1 (6 → 7)	61	59	63	62	64	59
	Resource 2 (5 → 6)	59	59	58	65	59	59

참고문헌

- Bedworth, D. D. (1973), *Industrial Systems : Planning, Analysis, Control*, The Ronald Press Company, New York.
- Bedworth, D. D. and Bailey, J. E. (1994), *Integrated Production Control Systems : Management, Analysis, Design*, 2nd edition, Prentice Hall, New York.
- Brooks, J. R. (1963), Algorithm for Activity Scheduling to Satisfy Constraints on Resource Availability, *unpublished paper*, Arizona State University.
- Elsayed, E. A. (1982), Algorithm for Project Scheduling with Resource Constraints, *International Journal of Production Research*, **20**(1), 95-103.
- Elsayed, E. A. and Boucher, T. O. (1994), *Analysis and Control of Production Systems*, 2nd edition, Prentice Hall, New York.
- Elsayed, E. A. and Nasr, N. Z. (1986), Heuristic for Resource-Constrained Scheduling, *International Journal of Production Research*, **24**(2), 299-310.
- Herroelen, W. S. (2005), Project scheduling-Theory and practice, *Production and Operations Management*, **14**, 413-432.
- Khattab, M. M. and Choobineh, F. (1991), A New Approach for Project Scheduling with Limited Resource, *International Journal of Production Research*, **29**(1), 185-198.
- Krüger, D. and Scholl, A. (2010), Managing and Modelling General Resource Transfers in (multi-)Project Scheduling, *OR Spectrum*, **32**, 369-394.
- Kumanan, S., Jose, G. J., and Raja, K. (2006), Multi-project Scheduling using an Heuristic and a Genetic Algorithm, *The International Journal of Advanced Manufacturing Technology*, **31**, 360-366.
- Mason, R. (1970), An Adaptation of the Brook's Algorithm for Scheduling Projects under Multiple Source Constraints, *M. S. thesis*, Arizona State University.
- Nasr, N. Z. (1983), New Algorithms for Project Scheduling under Resource Constraints, *M. S. thesis*, Rutgers University.
- Nazareth, T., Verma, S., Bhattacharya, S., and Bagchi, A. (1999), The Multiple Resource Constrained Project Scheduling Problem : A Breadth-first Approach, *European Journal of Operational Research*, **112**, 347-366.
- Whitehouse, G. E. and Brown, J. R. (1979), GENRES : An Extension of Brooks Algorithm for Project Scheduling with Resource Constraints, *Computers and Industrial Engineering*, **3**, 262-268.