

자원제약하의 복수 프로젝트 일정계획을 위한 휴리스틱 알고리즘

공명달¹ · 김정자²

¹영산대학교 경영정보학과/²동아대학교 산업시스템공학과

A Heuristic Algorithm for Resource-Constrained Multi-Project Scheduling

Myung-Dal Kong¹ · Jung-Ja Kim²

Resource-constrained project scheduling is to allocate limited resources to activities to optimize certain objective functions and to determine a start time for each activity in the project such that precedence constraints and resource requirements are satisfied.

This study suggests a multi-project scheduling model which can level work loads, make the most of production capacity and restrain the delay of delivery by developing a heuristic algorithm which minimizes the project completion time and maximizes the load rate under resource constraints.

1. 서론

1.1 연구배경 및 목적

일정계획 문제는 순서계획(sequencing)에서 작업순서와 방법이 결정되고, 공수계획에서 생산량이 소요공수(부하량)로 환산되면 다음에는 이들의 작업시기를 결정하는 일정계획(scheduling)이 마련되어야 한다.

이것은 생산현장마다 각기 다른 모형을 가지고 있는데, 이들 다양한 모델의 문제를 해결하기 위하여 오랜 연구가 수행되어 왔지만 간단한 일정계획 모형에서의 문제만 최적해를 구해주는 알고리즘이 존재할 뿐, 복잡한 시스템의 일정계획문제를 위한 최적화 알고리즘은 거의 개발되지 못하고 있다. 개별 공정(Job s-hop)모형과 흐름공정(flow shop)모형의 경우 가능한 작업순서가 천문학적인 수가 되기 때문이다. 프로젝트 일정계획의 경우에도 대규모 프로젝트의 수리적 정식화의 어려움 및 실행시의 복잡성으로 인하여 현실적이지 못하다.

이러한 이유로 해서 휴리스틱 접근방법들이 연구되어 왔다.

프로젝트 일정계획(project scheduling)은 개별 공정형 및 흐름공정형과는 달리 작업순서가 대부분 미리 정해져 있기 때문에 순서계획은 비교적 간단하며 반면에 공수계획에 의한 일정

계획이 주안점이 된다. 프로젝트 일정계획은 PERT/CPM 기법에 의하여 사업관리, 건설공사관리 등 1회성의 비반복적 업무에 이용되어 왔으나 제조업체의 생산관리 분야의 반복적 업무에는 적용 예가 아주 저조한 실정이다. 또한 기존의 연구에서는 프로젝트 소요기간의 최소화 등 일정의 길이를 단축시키는 연구는 많으나 복수 프로젝트 일정계획을 위한 부하평준화시 자원의 이용률인 평균 부하율(부하/능력×100)을 실증적으로 제시한 연구는 많지 않은 실정이다. Davis가 각 시점에서 자원유형별로 자원의 가용량에 대한 총소요량의 비율로서 이용률인 자(UR)를 사용하여 복수 프로젝트 일정계획 규칙의 수행도를 판별하기 위한 기준으로 제시한 바 있다.

본 연구에서는 복수 프로젝트인 경우 자원제약하에서 프로젝트 완료시점을 최소화시키며 부하율을 최대화시키는 휴리스틱 알고리즘을 개발하여 제한된 능력으로 작업부하의 평준화를 이루어 생산능력의 최대한 활용과 납기지연을 최대한 방지할 수 있는 일정계획 모형을 제시하고자 한다.

대부분의 휴리스틱 절차는 우선순위 규칙(priority rule)에 기반을 둔 단일 단계 휴리스틱(single pass heuristic)이다. 이것은 가능한 한 짧은 시간 내에 합리적인 좋은 해를 얻기 위하여 이용되어 왔다.

일정계획은 어떤 목적함수를 최적화시키기 위하여 제한된 자원을 활동에 할당하는 것이다. 단일 프로젝트 및 복수 프로

제트의 경우 가장 폭넓게 이용된 목적(objective)은 프로젝트 소요기간의 최소화이다. 그의 총프로젝트 비용의 최소화, 프로젝트 순현재가(net present value)의 최대화 등의 연구가 수행되어 왔으며, 복수 프로젝트의 경우 지연벌금(tardiness penalty) 또는 개개의 프로젝트 완료일자의 어떤 함수를 최적화시키는 것이다.

본 연구에서는 실제 문제에 적용가능한 프로젝트 일정계획 알고리즘을 개발하기 위하여 A업체에서 현재 운영중인 업무 모형에 따른 기존 시스템의 알고리즘(이하 기존 알고리즘이라 칭함)의 문제점을 분석, 고찰한다. 그리고 기존 알고리즘에 의한 일정계획의 프로젝트 완료시점(project completion time)을 최소화시키기 위하여 시간분석(일정전개)에 의하여 초기일정을 산출하고 여기에 가장 효과적인 우선순위 규칙을 적용하여 부하평준화를 통한 최적일정 생성, 부하평준화(load leveling) 및 부하율을 최소화시키는 알고리즘을 개발한다. 특히 활동별로 작업의 특성을 감안하여 작업중단 가능작업, 작업중단 불가능작업등을 고려함으로써 일정계획의 유연성과 실제 적용가능성을 높이며 총자원의 양이 제한된 조건하에서 유휴자원의 최소화 및 일정지연의 최소화를 위하여 작업장간 자원차입을 통한 일시적 능력조정이 가능한 알고리즘을 개발한다.

본 연구는 부하평준화를 통한 최적 일정산출에 중점을 두며, 기존 알고리즘에 의한 일정계획과 새로운 알고리즘에 의한 일정계획을 프로젝트 완료시점 및 부하율 측면에서 비교 분석함으로써 새로운 알고리즘에 의한 일정계획이 더 우수함을 증명한다. 다시 말해서 기존의 알고리즘에 비하여 제안 알고리즘이 일정을 더욱 단축시키며 자원의 이용률이 높다는 것을 제시한다. 또한 제안 알고리즘의 성능평가를 위하여 기존의 몇몇 휴리스틱 규칙에 의한 일정과 비교한다.

1.2 기존의 연구

지금까지 연구된 여러 휴리스틱 절차들 중 많은 연구자들이 가장 효과적이라고 제시한 우선순위 규칙은 최소작업 여유 또는 최소 총여유규칙(MINSLK, LTF)이며, 그의 SASP, FCFS 규칙 등이 있다. 이들 우선순위 규칙은 일정계획의 유효성에 큰 영향을 미치게 된다.

또한 부하평준화를 통한 최적 프로젝트 일정계획에 대한 휴리스틱 연구가 수행되어 왔다. 즉, Levy 등은 생산 인력 소요량을 평준화하기 위하여 가장 빠른 가능한 시작일자(EST)에 따라 각 작업장의 모든 작업들을 계획하고, 최대 부하시점(peak periods)에 계획된 몇몇 작업들을 이후의 (여유)기간으로 이동시킴으로써 최대 소요량(peak requirements)을 감소시키려고 하였다.

Burgess와 Killebrew는 각 시점의 부하제공을 전체기간에 걸쳐 합하여 이 값을 최소화시키도록 작업의 시작일을 조정하고자 하였으나, 가용능력을 고려하지 못하는 문제점이 있다.

Wiest는 가용자원(능력)에 초점을 두어 빠른 시작시점(EST)의 순서로 정렬된 작업에 대하여 자원을 기간별로 연속적으로 할당하였다.

그리고 Leachman은 연속적인 작업강도(work intensity)에 의하여 부하가 집중된 시점(peak region) 내의 작업강도를 변경시킴으로써 여러 자원들의 부하 최대점(load peak)을 최소화시키는 휴리스틱 기법을 제안하였다. 이들 연구는 작업활동 수, 작업장 수, 계획기간 등의 제한이 있거나 일정의 최소단위가 일(day) 단위로 되어 있어 실용성이 부족하다.

따라서 생산현장에서 실용성이 높고 효율적인 일정계획 방법이 요구된다.

2. 일정계획 절차 모형 설계

A업체는 현재 약 50여종(완성장비 30종, 구성품 20종)의 건설장비를 정비하고 있는 대단위 정비업체로서 연간 정비사업에 대한 계획 정비를 하고 있으며, 작업 방식은 Bay식 작업형태(프로젝트 작업방식과 유사)로 이루어지고 있다.

정비계획에 의거해서 실제로 작업을 실시하기 위해서는 언제 작업을 착수하고 작업을 완료할 것인가에 대한 실행계획으로서의 일정계획이 수립되어야 한다. 프로젝트 일정계획은 PERT 네트워크를 토대로 하여 전개된다.

즉 네트워크상에서 활동(공정)별 시작일과 완료일을 산정한다. 소일정계획 수립을 위한 절차모형을 설계하면 <그림 1>과 같다.

정비계획이 수립되면 이에 의하여 대일정, 중일정, 소일정계획을 수립하게 되며 이들 계획 중 소일정계획은 계획수립단계와 작업실행단계를 가진다.

<그림 1>에서 일정계획 절차 중 핵심적인 부분은 일정전개와 부하평준화이다.

일정전개는 계획공정도(Network Diagram)에 의거하여 초기일정을 산출하는 과정이다. 능력과 부하를 전혀 감안하지 않고 활동의 소요기간(duration)에 의해서만 일정을 전개한다.

부하평준화를 위해서는 대상기간, 작업장별 부하율, 공정 우선순위 기준 등을 입력 매개변수(parameter)로 부여할 수도 있으나 이들 값을 미리 정해둘 수 있다.

본 연구에서는 대상기간을 시작 시점만 설정하는 방식을 택하며 시작시점은 현재기준일을 부여할 수 있도록 한다.

작업장별 부하율도 현장의 사정을 감안하여 조정할 수 있으나 기본적으로 사용하는 100%를 기준으로 적용한다. 공정 우선순위(priority rule)는 기존 시스템에서 적용되는 ES, TE, DU, RN의 조합규칙과 기존의 연구에서 가장 효과적이라고 입증된 MINSLK법, MINSLK를 조합한 MINSLK/RSM/MINLFT법, SASP법, LALP법, 그리고 본 연구에서 처음 제안하는 MLPF 조합법 등의 규칙들을 적용한다.

<그림 1>에서 부하평준화를 위한 처리 절차는 다음과 같다.

- (가) 일정전개에 의하여 산출된 초기계획에 의거 대상기간, 작업장, 일자별로 보유공수를 산정한다. 이때 공장유효력

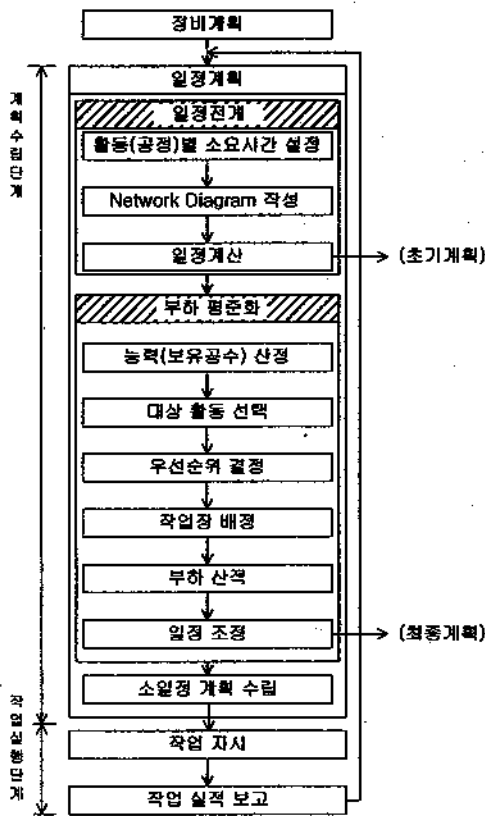


그림 1. 일정계획 절차 모형.

을 감안하며 작업장별 실근무시간, 실작업인원수, 부하율, 가동률을 곱하여 산출한다(보유공수 산정).

- (나) 일정계획 테이블에 등록된 활동(공정)들을 대상으로 대상기간에 해당되는 활동들을 선택한다. 이때 활동의 초기계획상의 시작일이나 완료일이 대상기간 내에 하나라도 존재하면 대상이 되며, 완료된 활동(실제 완료일이 있는 경우)은 대상에서 제외된다(대상활동 선택).
- (다) 선택된 활동들을 정해진 우선순위 기준에 따라 정렬(sort)한다(우선순위결정).
- (라) 활동별로 작업장이 이미 정해져 있는 경우에는 작업장 배정을 별도로 할 필요가 없으나, 활동과 작업장이 1:n의 관계인 경우 작업장의 부하율(부하공수/보유공수 × 100)이 최저인 작업장을 우선적으로 선정한다(작업장 배정).
- (마) 작업장이 정해진 활동의 경우 해당 작업장의 전개 시작일부터 부하를 능력 이하로 산적한다(부하산적).
- (바) 부하를 산적한 후 과부하분에 대하여 일정을 조정한다. 활동을 산적한 후 과부하가 아니면 초기계획상의 해당 활동의 일정을 그대로 설정한다. 대상기간 내의 모든 활동이 처리된 후 연동성을 확인(check)하여 일정을 조정하게 되며 이에 따라 소일정계획이 수립된다(일정 조정).

3. 휴리스틱 알고리즘의 개발

본 연구에서 사용하는 기존 알고리즘과 제안 알고리즘의 용어에 대하여 다음과 같이 정의한다.

기존 알고리즘이란 기존의 A업체에서 사용하고 있는 일정계획을 위한 세부 절차 및 방법들을 말하고, 제안 알고리즘은 기존 알고리즘을 개선하고 기능을 보강시켜 개발한 알고리즘을 말한다.

본 연구의 대상이 되는 소일정계획은 계획의 정확도를 높이기 위하여 일일단위로 수립되며, 공정의 소요시간(duration) 단위는 분(minute)단위이고, 보유공수(능력) 및 부하공수의 단위는 인분(man-minute)이다.

3.1 기존 알고리즘

기존 시스템의 알고리즘 단계별 절차는 다음과 같다.

단계 0. 변수들을 초기화한다.

(EST, EFT, LST, LFT, TF, FF)

단계 1. 활동 데이터를 읽는다.

단계 2. 초기 일정을 계산한다.

(EST, EFT, LST, LFT, TF, FF 등)

단계 3. 보유 공수를 산정한다

(작업장/일자별).

단계 4. 소요 공수(부하공수)를 산정한다

(작업장/일자별).

단계 5. 평준화 대상 활동의 우선 순위를 결정한다.

단계 6. 우선 순위별로 활동들을 정렬한다.

단계 7. 활동별로 작업장을 결정한다.

단계 8. 부하를 산적한다.

단계 9. 부하산적 후 보유 공수(능력) 대 소요 공수(부하)를 비교한다.

단계 10. 과부하 여부를 확인한다.

(a) 과부하가 아니면 단계 13으로 간다.

(b) 과부하이면 단계 11로 간다.

단계 11. 부하를 조정한다. 과부하분은 다음 일자에 산적(할당)한다.

단계 12. 과부하인 활동의 일정을 변경한다.

단계 13. 모든 활동에 자원이 할당되었는지 확인한다.

(a) 할당이 완료되었으면 종료한다.

(b) 할당이 완료되지 않았으면 단계 7로 간다.

단계 1에서는 공정 테이블에 있는 활동의 소요기간(duration), 선후관계, 연동관계 등의 활동 데이터를 읽어 가져온다.

단계 2에서 초기 일정을 계산한 것이 초기 계획이 되며, 자원, 능력, 부하 등을 전혀 고려하지 않은 소요 시간(duration)

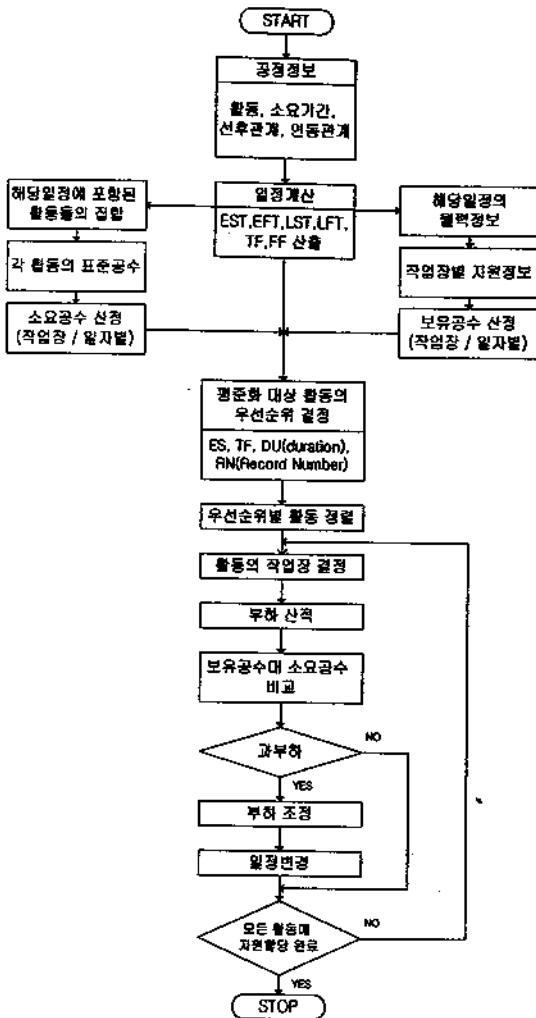


그림 2. 기존 알고리즘의 흐름도.

에 의하여 전개된 일정으로서 이때 현재기준일을 배개 변수로 입력받아 이 시점 이후로 일정을 전개하게 된다.

단계 3에서 보유 공수 산정시에는 작업장별 실작업 인원수와 유효 정보에 의한 실근무 시간, 가동률, 부하율 등을 감안하여 계산한다. 여기서 가동률과 부하율은 각각 기본적으로 적용하는 100%를 적용한다.

단계 4에서 소요 공수는 작업 물량을 부하량으로 계산한 것으로, 각 장비별 활동의 표준 공수(입력 데이터)에 의거 작업장/일자별로 산정한다.

단계 5에서 우선 순위 결정에 단일 단계 휴리스틱(single pass heuristic) 규칙인 우선 순위 규칙(작업 할당 규칙, 순서 규칙)이 적용된다. 기존 시스템의 우선 순위는 EST가 빠른 활동, TF(Total Float)가 작은 활동, 소요 시간(duration)이 짧은 활동, RN(Record Number)이 작은 활동의 순서이다. 현재 진행중인 활동은 기본적으로 가장 높은 우선 순위를 가진다.

단계 6에서는 주어진 우선 순위에 의하여 활동들이 오름차순으로 정렬(ascending sort)되면, 이 활동들의 순으로 계속적으로 읽어서 모든 활동에 자원이 할당될 때까지 처리한다.

단계 7에서 활동별 작업장 결정시에는 활동별로 이미 공정 정보에 등록되어 있는 작업장(주로 단일 작업장)을 이용한다.

단계 8에서 부하 산적은 초기일정을 기준으로 작업장의 가용 자원(능력) 범위 내에서 작업장별/일자별로 작업 물량(장비)의 활동별 표준 공수를 할당하는 것이다.

부하 산적 후 보유 공수와 부하 공수를 대비하여 과부하(보유 공수 < 부하공수)인 경우에는 부하를 조정하여 해당 활동의 완료일자를 지연시키고, 과부하가 아닌 경우에는 초기 계획의 해당 일자에 부하를 산적한다.

단계 9에서는 각각 단계 3과 단계 4에서 산정한 보유공수와 소요공수를 작업장별/시점별로 비교한다.

단계 10에서는 부하와 능력을 비교한 결과 과부하가 아니면 해당 활동의 초기 일정으로 계획이 설정되고 과부하가 발생하면 부하를 조정하는 루틴을 수행하게 된다.

단계 11에서 과부하가 발생한 활동의 일정은 지연이 발생하여 초기 일정보다 늦은 시점에 계획이 설정된다.

단계 13에서는 다음 활동들에 대해서도 부하산적을 하여 자원이 할당되었는지 확인하는 것이다. 즉 모든 활동에 대해서 부하를 산적하여 자원을 할당하여야 각 작업장에서 누락되는 활동(작업)이 없이 일정계획이 수립되는 것이다.

3.2 제안 알고리즘

제안 알고리즘은 활동의 분할을 허용한다.

활동의 분할 또는 선매는 활동의 소요시간(duration) 또는 자원의 분할을 의미하며, 따라서 부하 산적시 과부하가 발생하면 잔여 자원을 활동의 최소공수와 비교한 후 그 다음 시점에 분할하여 우선적으로 할당할 수 있게 한다. 이렇게 함으로써 일정 지연을 억제시킨다.

본 연구에서는 프로젝트 완료시점을 최소화시키고 자원의 활용도를 높이기 위하여 부하평균화시 과부하가 발생한 경우에 한하여 최소작업소요공수를 적용시키고, 그렇지 않으면 표준공수로 부하를 산적하여 계획한다.

각 활동별로 표준공수와 최소 공수를 가지고 있으며, 이것이 활동분할의 기준이 된다. 즉 표준공수와 최소공수가 같은 활동은 작업분할이 불가능하다. 제안 알고리즘의 흐름도를 나타내면 <그림 3>과 같다.

또한 유휴자원의 활용도를 높이고 작업지연을 억제시키기 위하여 가용자원의 차입을 허용한다.

단계 0. 각 변수들을 초기화한다.

(EST, EFT, LST, LFT, FF, TF 등)

단계 1. 활동 데이터를 읽는다.

단계 2. 초기 일정을 계산한다.

(EST, EFT, LST, LFT, FF, TF)

단계 3. 보유공수를 산정한다(작업장/ 일자별).

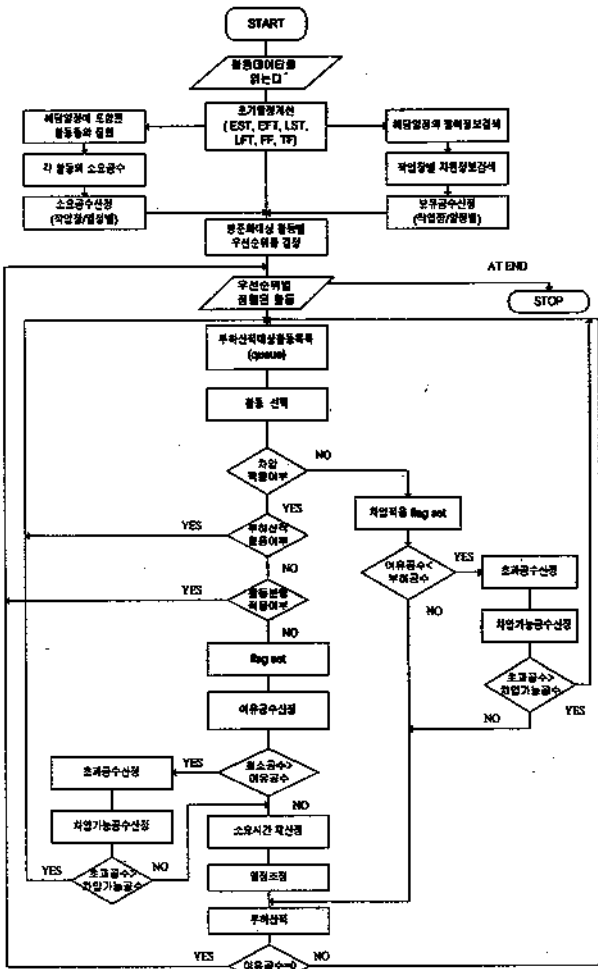


그림 3. 제안 알고리즘의 흐름도.

- 단계 4. 소요공수(부하공수)를 산정한다(작업장/일자별).
- 단계 5. 평균화 대상 활동들의 우선순위를 결정한다.
- 단계 6. 우선순위에 따라 정렬(sort)된 활동들을 읽는다. 활동을 다 읽었으면(AT END) 종료한다.
- 단계 7. 부하산적대상 활동목록(Queue)을 읽는다.
- 단계 8. 하나의 활동을 선택한다.
- 단계 9. 자원차입의 적용 여부를 검토한다.
 - (a) 차입을 적용하지 않았다면 단계 10으로 가고,
 - (b) 차입을 적용하였으면, 단계 15로 간다.
- 단계 10. 차입 적용 표시(flag)를 설정한다(flag='1').
- 단계 11. 여유공수 < 부하공수인지 확인한다.
 - (a) 여유공수 < 부하공수이면 단계 12로 가고,
 - (b) 여유공수 < 부하공수가 아니면 단계 24로 간다.
- 단계 12. 초과공수를 산정한다.
- 단계 13. 타 작업장으로부터 차입 가능한 공수를 산정한다.
- 단계 14. 초과공수 > 차입가능 공수인지 확인한다.
 - (a) 초과공수 > 차입가능 공수이면 단계 7로 가고,
 - (b) 초과공수 > 차입가능 공수가 아니면 단계 24로 간다.
- 단계 15. 부하를 산적한 활동인지 확인한다.

- (a) 부하를 산적한 활동이면 단계 7로 가고,
- (b) 부하를 산적한 활동이 아니면 단계 16으로 간다.
- 단계 16. 활동 분할적용여부(flag='2')를 확인한다.
 - (a) 활동분할이 적용되었다면 단계 6으로 가고,
 - (b) 활동분할이 적용되지 않았다면 flag='2'로 설정하고 단계 17로 간다.
- 단계 17. 여유공수를 산정한다.
- 단계 18. 활동의 최소공수 > 여유공수인지 확인한다.
 - (a) 최소공수 > 여유공수이면 단계 19로 가고,
 - (b) 최소공수 > 여유공수가 아니면 단계 22로 간다.
- 단계 19. 초과공수를 산정한다.
- 단계 20. 차입가능 공수를 산정한다.
- 단계 21. 초과 공수 > 차입가능 공수인지 확인한다.
 - (a) 초과공수 > 차입가능 공수이면 단계 7로 가고,
 - (b) 초과공수 > 차입가능 공수가 아니면 단계 22로 간다.
- 단계 22. 소요시간(duration)을 재산정한다.
- 단계 23. 일정을 조정한다.
- 단계 24. 부하를 산적한다.
- 단계 25. 여유공수 = 0인지 확인한다.
 - (a) 여유공수 = 0 이면 단계 6으로 가고,
 - (b) 여유공수 = 0 가 아니면 단계 7로 간다.

알고리즘의 처리순서를 보면, 먼저 일정계산에 필요한 변수들을 초기화하여 계산상의 오류를 방지하도록 한다.

- 단계 1에서는 기초자료 구축을 통하여 이미 공정(활동)테이블에 등록되어 있는 복수 프로젝트(네트워크)의 활동들에 대한 소요기간(duration), 선후관계, 연동관계 등의 일정계산에 필요한 데이터를 읽어 가져온다.
- 그 다음 단계 2에서는 이들 데이터에 따른 각 네트워크상에서의 활동간의 연결관계(선후관계, 연동성)를 이용하여 활동별 소요기간(duration)에 따라 전진계산법으로 EST, EFT를, 그리고 후진계산법으로 LST, LFT, TF 및 FF등을 산출한다.
- 이때 능력, 자원 등을 전혀 고려하지 않고 입력된 현재 기준일을 시작활동의 EST로 설정하여 현재기준일의 시점을 기준으로 시작활동에서부터 종료활동까지 소요기간에 의해서만 순차적으로 윌력을 감안하여 일정을 전개하게 된다.
- 단계 3에서는 현재 기준일 이후 시점의 윌력정보(근무일, 비근무일정보)에 따른 실제 근무시간, 작업장의 실제 작업인원수(설비대수), 가동률, 부하율을 곱하여 작업장별/일정별로 보유공수를 산출한다.
- 단계 4에서는 일정전개에 의해 산출된 초기일정상의 각 활동들의 표준공수(입력데이터)를 작업장별/초기일정별로 산정하여 소요공수로 설정한다.
- 단계 5에서는 일정계획 수립 대상 활동(activity)들에 대하여

단일 단계 휴리스틱(single pass heuristic)에서 사용하는 우선순위 규칙(priority rule) 또는 작업할당 규칙(dispatching rule)을 적용하여 대상 활동들을 우선순위별로 오름차순에 따라 정렬(sort)시킨다. 어떤 시점에서 시작가능한 활동들의 집합을 Queue에 배열시켜 정렬시킨다. 즉 일정한 시점에 일정계획이 가능한 활동의 집합에 우선순위를 부여하고 일정이 동시에 전개되며, 우선순위가 현재 활동의 구성에 따라 일정계획중에 결정되는 병렬법(parallel method)이다.

단계 6, 7, 8에서는 선정된 우선순위 규칙에 의하여 정렬된 활동들에 대한 부하 평균화 대상 활동들을 선정하고 활동 queue에 배열된 활동들을 하나씩 읽어 처리한다.

단계 9에서는 자원차입이 적용되지 않았을 경우(NO인 경우)에는 표준공수로 부하를 산적하게 되고 차입이 적용되었으면 다음 확인 사항인 부하산적 여부를 확인하게 된다.

차입이 적용되지 않았을 경우에는 부하산적 후 일정전개시의 초기계획의 소요시간을 적용한다. 차입이 적용되었을 경우는 과부하시의 부하산적 루틴이 된다. 이때에는 작업분할이 적용된다.

단계 11에서 여유공수 < 부하공수이면 과부하가 발생하여 초과(over)되는 공수를 산정하여 이 초과되는 공수만큼 타 작업장으로부터 자원을 차입하게 된다.

단계 12는 과부하가 발생한 경우 부하공수에서 여유공수를 뺀 차이분 만큼의 초과공수를 산정하게 되는데, 이것은 이 초과공수만큼 타 작업장으로부터 자원을 차입해오기 위한 것이다.

단계 14에서 초과공수와 차입가능공수를 산정하고 난 후 이 두 가지의 공수를 비교하게 되는데 초과공수 > 차입가능공수이면 차입을 하지 않고 일정을 지연시키게 되고 차입해서 초과공수를 충당할 수 있다면 타 작업장으로부터 같은 시점에 초기일정이 설정되어 있는 경우 차입을 하게 된다.

단계 15는 해당 활동이 부하산적을 완료한 활동인지의 여부를 확인하는 것이다. 부하산적을 한 활동이면 다음 활동을 읽어서 처리한다.

단계 16은 활동(작업)의 분할 적용여부를 확인하여 분할가능 활동인 경우 단계 18에서 활동 분할을 적용하게 된다.

단계 17에서는 여유공수를 산출하게 되는데, 여유공수는 보유공수(능력)에서 누적된 부하공수를 뺀 값이다.

단계 18에서는 활동 입력자료상의 최소공수와 여유공수를 비교하여 최소공수를 산적할 만한 여유를 가지고 있으면(최소공수 ≤ 여유공수) 최소공수로 부하를 산적하게 되고 반면에 소요시간(duration)은 활동 입력표상의 최대소요시간을 적용하여 단계 22에서 소요시간 계산정을 하게 된다.

만약 최소공수를 수용할 만한 여유공수가 없으면(최소공수 >

여유공수) 초과공수(단계19)와 차입가능 공수(단계 20)를 산정하여 초과공수와 차입가능 공수를 비교한다. 초과공수 > 차입가능 공수이면 다음 활동을 읽어서 처리하고 아니면 소요시간 계산정을 통하여 일정조정 및 부하산적 후 일정계획을 하게 된다.

단계 22에서 소요시간 계산정은 과부하 활동에 대한 소요시간으로서 일정계산에 의하여 산출된 초기일정상의 활동의 소요시간보다 초과공수만큼 더 일정을 지연시켜 계획한다. 즉 늘어나는 총 소요시간은(표준공수/잔여공수) × 표준소요시간으로 산출한다. 여기서 잔여공수는 잔여 가용자원, 즉 해당 작업장에서 현재 남아있는 잔여공수를 말한다.

단계 25에서는 부하산적 후 보유공수의 한도 내에서 부하를 산적할 여지가 있는지(여유공수=0) 해당 작업장/시점에 대하여 판단하여, 여유가 없으면 다음 시점에 부하를 산적하고 여유가 있으면 다음 활동을 읽어서 해당 시점에 부하를 산적/일정조정을 하게 된다.

제안 알고리즘은 부하평균화시 과부하가 아닌 경우에는 활동의 표준공수(소요공수)로 부하를 산적하며, 과부하인 경우에는 활동의 최소공수로 부하를 산적한다. 초기 일정을 산출하는 처리절차는 기존 알고리즘과 같다. 기존 알고리즘과의 큰 차이점은 활동의 분할이 가능한 것과 가용 자원의 차입을 허용하는 것의 두 가지이다. 여기서 사용된 몇 가지 용어에 대하여 정의를 한다. 보유공수란 해당 작업장에서 현재 시점에 보유하고 있는 가용자원(능력)을 공수로 나타낸 것이며, 부하공수 또는 소요공수는 특정 활동을 수행하는 데 소요되는 공수를 말한다.

보유공수에서 누적된 부하공수를 뺀 것이 여유공수이다. 즉 여유공수는 특정시점에 능력을 초과하지 않는 한도 내에서 부하를 산적할 수 있는 최대 부하산적 가능 공수이다. 초과 공수는 부하를 산적하려고 하는 활동의 부하공수에서 여유공수를 뺀 공수값이다. 부하공수 > 여유공수인 경우 과부하, 즉 초과공수가 발생하며 부하공수 = 여유공수인 경우에는 최대한도의 부하를 산적하여 더 이상 산적할 여유가 없는 경우이다. 부하공수 < 여유공수인 경우에는 아직 부하를 산적할 수 있는 여유를 가지고 있는 경우이다. 초과공수는 가용자원 측면에서 보면 활동을 수행하는 데 필요한 부족 공수가 된다. 차입가능공수는 부하산적시 특정작업장에서 가용자원(능력)이 부족한 경우 자원의 여유가 있는 타 작업장으로 부터 대체할 수 있는 자원을 일시적으로 차입하는 공수이다.

이것은 유휴자원을 최대한 활용하는 동시에 일정의 지연을 억제시키기 위한 것이며, 고도의 숙련작업인 경우는 적용하지 않는다.

본 알고리즘에서 활동분할의 기준이 되는 것은 활동별로 가지고 있는 입력데이터인 표준공수와 최소공수이며, 표준공수와 최소공수가 같으면 분할이 불가능한 활동이다.

3.3 제안한 우선순위 규칙

기존 연구에서 복수 프로젝트 문제에 가장 효과적인 휴리스틱 규칙이라고 발표되었던 규칙은 MINSLK (Fendley, Mize, Patterson), SASP (Kurtulus 와 Davis, Tsubakitani 와 Decro), MINSLK / RSM / MINLFT (Boctor) 규칙 등이 있으며, 기존 휴리스틱 규칙의 유효성(effectiveness)에 관한 가장 최근의 완전한 분석이 Davis 와 Patterson의 연구(1975)에서 이루어졌다. 이 연구결과에서는 MINSLK 규칙이 단일 프로젝트 및 복수 프로젝트에서 효과적인 것으로 나타났으며, 몇몇 휴리스틱의 성능과 개개의 문제 특성간의 명백한 관계는 밝히지 못하였다. Bell 과 Han은 기존의 휴리스틱 보다 더 우수하게 수행되며 기존 휴리스틱과는 다른 방식의 2단계 휴리스틱을 개발하였다.

이전의 연구결과에 따른 휴리스틱 규칙 외에 제안 알고리즘의 성능향상을 위하여 본 연구에서는 MLPF 규칙을 처음으로 제안한다.

MLPF(Mixed Largest-Project-First)법은 복수 프로젝트에 적용되는 규칙으로서 가장 긴 프로젝트, ES(Earliest Start Time), TF(Total Float), DU(Duration), RN(Record Number)의 순서로 적용된다. 즉 여러 프로젝트 중 가장 긴 프로젝트의 활동, ES가 빠른 활동, TF가 작은 활동, 소요기간(DU)이 짧은 활동, RN이 작은 순으로 우선순위가 높게 부여된다. 어떤 시점에서 일정계획이 가능한 활동들의 집합에서 대상활동의 프로젝트의 길이가 같은 경우 그 다음 적용기준인 ES를 비교하여 ES가 빠른 활동의 우선순위를 높게 부여하며 ES가 같은 경우 그 다음 적용기준인 TF를 비교하게 된다. TF가 같은 경우 DU(Duration)에 의하여 비교하며, DU가 같은 경우에는 RN(Record Number)을 비교하여 우선순위를 매기게 된다. 보통 수많은 활동 데이터를 대상으로 우선순위를 매기는 경우에도 긴 프로젝트, ES, TF, DU까지 적용하면 대부분 우선순위가 결정된다. 그러나 혹시 활동의 우선순위가 중복(duplicate)되는 경우가 발생할 수 있는데 이때 적용되어 우선순위를 결정짓는 것이 RN(Record Number)이며, 이를 tie-breaker 라고 한다. 여기서 RN은 FCFS 역할을 한다. 이 순서에 따라 오름차순으로 정렬(sort)하여 활동이 배열된다. 이러한 우선순위로 배열된 활동을 순차적으로 선택하여 자원을 할당하게 된다.

4. 알고리즘의 적용

4.1 실험 모형

본 연구에서 실험한 모형은 실제 모형을 축소한 모형으로서 4개의 프로젝트(네트워크)에 대하여 실험을 하였는데, 각 프로젝트별로 11~21개의 활동 등 총 65개의 활동(activities), 4개의 작업장 및 자원유형으로 구성되어 있으며, 동일한 모형으로 기존 시스템의 알고리즘과 신규의 제안 알고리즘을 각각 적용하여 실험하였다.

가. 입력자료

실험에 사용된 입력자료(input data)는 활동(activity)과 자원(resource)에 관련된 자료로서 다음의 <표 1> 및 <표 2>와 같다.

이들 자료는 프로젝트 일정계획을 수립하는 데 필요한 기본적인 자료이다.

<표 1>은 프로젝트 P1, P2, P3, P4에 대한 각각의 세부적인 활동(activity)의 입력자료를 나타낸다.

표 1. 활동 입력자료

프로젝트	활동 코드	활동 명	소요시간(분)		작업장	소요인원(명)	
			표준	최대		표준	최소
P1	0102	외부부품 해체	1,356	5,424	3D350	4	2
	0103	엔진분리 인계	1,214	3,035	3D350	5	2
	0204	최종구동장치 해체	2,094	8,114	3D350	9	9
	0304	변속기 분리 해체	2,196	6,588	3D350	3	1
	0408	자체정비 및 프레임조립	6,482	15,125	3D320	7	3
	0809	엔진부착 및 성능시험	6,642	19,926	3D320	4	3
	0905	블록 정비	1,954	5,862	3D320	3	1
	0506	실린더 헤드 정비	2,431	7,293	3D320	3	3
	0607	내부 부속 수리 조립	1,418	2,172	3D320	3	2
	0708	엔진 조립 및 기능검사	2,701	10,804	3D320	4	4
0910	도색 및 최종 검사	1,412	1,412	3D340	2	2	
P2	0102	장비 입고 및 외부부품해체	3,519	8,873	3D350	5	2
	0203	엔진 및 구동장치 해체	3,824	17,208	3D350	9	2
	0304	차체 기초정비	1,025	3,075	3D320	9	1
	0305	변속기 조립	995	2,985	3D320	3	3
	0406	도올러 및 아이들러 조립	1,567	3,918	3D320	5	2
	0506	최종구동장치 조립	1,532	3,830	3D320	5	2
	0608	유압실린더 조립	1,210	2,420	3D320	2	2
	0810	컨버터 조립	1,125	3,375	3D320	3	3
	1012	트랙프레임 조립	1,334	5,336	3D320	4	1
	1219	컨버터 및 엔진 부착	2,012	8,128	3D320	8	2
	1309	부품 입고 분류 저장	580	1,160	3D320	2	1
	0709	크랭크 샤프트 연마	1,428	4,284	3D520	3	3
	0911	실린더 헤드 정비	1,928	5,784	3D520	3	1
1112	엔진 조립 및 기능검사	2,230	5,575	3D520	5	2	
1314	최종검사 및 납품	305	610	3D340	2	1	
P3	0102	작업장치 및 예술분리해체	2,238	6,714	3D350	6	2
	0203	엔진 및 하부 해체	3,428	8,570	3D350	5	2
	0304	해체 부품 입고	656	1,968	3D320	3	3
	0406	스타링 작업	889	2,667	3D320	3	3
	0609	정비 기초 작업	1,132	4,528	3D320	4	1
	0407	클러치 조립	1,274	2,518	3D320	2	1
	0710	변속기 조립	1,358	2,716	3D320	2	2
	0912	사륜 및 탄압 조립	1,585	7,925	3D320	5	1
	1013	중간맞손 조립	921	1,382	3D320	3	2
	1216	작업 장치 정비	822	2,466	3D320	3	1
	1315	후드 및 배기장 정비	856	1,111	3D320	4	3
	1516	엔진 및 유압장치 조립	1,563	3,126	3D320	4	4
	0905	크랭크 샤프트 연마	1,214	3,612	3D520	3	1
	0508	블록 정비	1,025	2,050	3D520	2	2
	0811	실린더 헤드 정비	1,301	3,903	3D520	3	1
1114	내부부속 수리 조립	701	3,124	3D520	4	1	
1415	사동 및 기능 검사	856	1,712	3D520	2	1	
1617	최종검사 및 납품	456	912	3D340	2	1	
P4	0102	외부부품 해체	308	924	3D350	3	1
	0203	상부엔진 분리 해체	1,102	11,021	3D350	3	3
	0304	하부엔진 분리 해체	1,205	205	3D350	3	3
	0305	상부엔진 세척	874	1,718	3D350	2	1
	0405	하부엔진 세척	874	1,718	3D350	2	1
	0506	분류 및 검사	658	1,974	3D350	3	1
	0207	부품 세척	325	325	3D350	1	1
	0608	상부 프레임 정비	1,125	4,500	3D320	4	1
	0809	상부 맞손 정비	932	3,728	3D320	4	1
	0810	하부 클러치 정비	916	2,718	3D320	3	1
	0911	상부엔진 라디에터 정비	691	1,728	3D320	5	2
	1012	하부엔진 라디에터 정비	852	2,130	3D320	5	2
	1114	상부클러치 및 와이어 정비	1,468	5,138	3D320	7	2
	1214	하부유압장치 정비	1,267	3,801	3D320	3	1
	1416	상부엔진 도색	889	2,667	3D320	3	3
1617	하부엔진 도색	1,001	3,003	3D320	3	3	
0713	크랭크 샤프트 연마	1,647	4,911	3D520	3	1	
1315	실린더 헤드 정비	1,746	5,238	3D520	3	3	
1517	부속 수리 및 조립	1,324	5,296	3D520	4	1	
1718	엔진부착 및 기능검사	1,648	4,120	3D520	5	2	
1819	최종검사 및 납품	1,025	2,050	3D340	2	1	

표 2. 자원 입력자료

작업장(자원)	보유 인원수	보유 공수
3D350	21	1260
3D320	20	1200
3D520	10	600
3D540	5	300

예를들어 프로젝트 P1의 경우 11개의 활동에 대한 코드와 명칭이 나타나 있으며 이들 각각의 활동에 대한 소요시간(duration)의 표준치와 최대치, 작업장, 소요인원의 표준치와 최소치를 제시하였다.

각 활동에 대한 작업장은 단일 작업장으로 설정하였고 소요인원 중 표준인원과 최소인원이 같은 경우에는 활동의 분할이 불가능하며, 표준공수는 표준소요시간에 표준인원을 곱한 값이다.

<표 2>에서 작업장은 동일한 종류의 작업(예: 해체, 조립, 도장 등)을 수행하는 작업자들의 집단으로서 자원유형(resource type)에 해당된다. 보유인원수는 특정 작업장에서 가용자원(available resource)으로서 능력(capacity)이 된다.

나. 실험 환경

실험은 기존 시스템의 실제 운영환경에서 수행하였다. 즉 클라이언트-서버 구조로 구축된 환경에서 전산실의 서버인 HP9000/G40을 이용하여 기존 알고리즘과 신규의 제안 알고리즘의 성능을 실험하였다. 기존 시스템은 영국 Lucas Management사 제품의 공정관리 전용 4세대 언어인 ARTEMIS로 개발하여 실험하였으며, 제안 알고리즘(신규 알고리즘)은 Gupta사 제품의 4세대 언어인 SQLWindows로 개발하여 실험을 하였다. 기존 알고리즘은 ARTEMIS, 제안 알고리즘은 SQLWindows로 개발하여 실험하였으나 이들은 제품이 서로 다른 4세대 언어일 뿐 컴퓨터 언어상의 기능이나 실험과정에서의 제약이나 문제점은 없었다.

4.2 실험 조건

기존 알고리즘과 제안 알고리즘을 동일한 실험조건하에서 실시하였으며, 실험 파라미터는 다음과 같다.

- 목표 시작일 : 1998. 7/1. 08 : 00
- 작업 근무시간 : 매일 00 : 00분 ~ 24 : 00분
(7/1일은 08 : 00 ~ 24 : 00)
- 비근무시간 : 고려안함
- 일정의 기본단위 : 분단위
- 우선순위 규칙 : ES, TF, DU, RN 규칙, SASP 규칙, LALP 규칙, MINSLK 규칙, MLPF 규칙(Davis, E.W.(1973))

4.3 실험 결과

가. 기존 알고리즘에 의한 결과

<표 3>에서 초기일정에 비하여 부하조정으로 가장 많이

표 3. 기존 알고리즘에 의한 초기일정과 최종일정의 비교

우선순위규칙	프로젝트	초기일정의 완료시점	최종일정의 완료시점	지연시간
ES, TF, DU, RN	P1	7/13, 19:06	7/16, 12:21	2일 17시간 15분
	P2	7/12, 10:21	7/13, 22:38	3일 12시간 17분
	P3	7/10, 06:22	7/13, 13:23	3일 7시간 1분
	P4	7/10, 07:26	7/13, 03:13	2일 19시간 47분
MINSLK, RSM, MINLFT	P1	7/13, 19:06	7/15, 17:42	1일 22시간 36분
	P2	7/12, 10:21	7/12, 10:21	0
	P3	7/10, 06:22	7/12, 01:38	1일 19시간 16분
	P4	7/10, 07:26	7/17, 00:40	6일 17시간 14분
SASP	P1	7/13, 19:06	7/20, 01:02	6일 5시간 56분
	P2	7/12, 10:21	7/16, 20:40	4일 10시간 19분
	P3	7/10, 06:22	7/10, 06:22	0
	P4	7/10, 07:26	7/10, 07:26	0
LALP	P1	7/13, 19:06	7/13, 19:06	0
	P2	7/12, 10:21	7/12, 10:21	0
	P3	7/10, 06:22	7/15, 13:56	5일 7시간 34분
	P4	7/10, 07:26	7/16, 09:47	6일 2시간 21분
MINSLK	P1	7/13, 19:06	7/15, 17:42	1일 22시간 36분
	P2	7/12, 10:21	7/12, 10:21	0
	P3	7/10, 06:22	7/12, 01:38	1일 19시간 16분
	P4	7/10, 07:26	7/17, 00:40	6일 17시간 14분
MLPF	P1	7/13, 19:06	7/13, 19:06	0
	P2	7/12, 10:21	7/12, 10:21	0
	P3	7/10, 06:22	7/20, 01:30	9일 19시간 8분
	P4	7/10, 07:26	7/14, 07:11	3일 23시간 45분

지연된 프로젝트는 MLPF 조합을 적용한 경우의 P3 프로젝트로서 9일 19시간 8분이다. 각 규칙을 적용한 이들 집합 중 단일 프로젝트 단위로 가장 지연이 적은 것은 ES, TF, DU, RN 규칙을 적용했을 경우의 P2 프로젝트로서 3일 12시간 17분이다.

그러나 전체 프로젝트의 완료시점은 LALP 규칙을 적용했을 경우의 P4 프로젝트('98. 7/16. 09 : 47분)가 가장 빠르다.

나. 제안 알고리즘에 의한 결과

<표 4>에서 부하평준화로 인하여 초기 일정에서 가장 많이 지연된 프로젝트는 LALP 규칙을 적용했을 경우의 3일 23시간 15분이며, 가장 적게 지연된 프로젝트는 SASP 규칙을 적용했을 경우의 P1, LALP 규칙을 적용했을 경우의 P1, MLPF 조합 규칙을 적용했을 경우의 P1 프로젝트로서 일정지연이 없으며 초기 일정과 같다. 각 규칙을 적용하여 가장 많이 지연된 프로젝트집합들 가운데 가장 지연이 적은 프로젝트는 MLPF 조합 규칙을 적용한 경우의 P3 프로젝트로서 2일 14시간 16분인데, 이것은 MLPF 조합규칙이 프로젝트 전체의 지연이 가장 적으며 또한 가장 효과적이라는 것을 뒷받침한다. 즉 프로젝트 완료시점의 최소화의 기준에서도 MLPF 조합규칙이 가장 효과적이다.

4.4 비교분석 및 고찰

기존 알고리즘을 적용하여 산출한 최종 실험결과와 신규의

표 4. 제안알고리즘에 의한 초기일정과 최종일정의 비교

우선순위규칙	프로젝트	초기일정의 완료시점	최종일정의 완료시점	지연시간
ES, TF, DU, RN	P1	7/13, 19:06	7/15, 06:19	1일 11시간 13분
	P2	7/12, 10:21	7/15, 16:02	3일 5시간 41분
	P3	7/10, 06:22	7/11, 21:27	1일 15시간 5분
	P4	7/10, 07:26	7/12, 08:22	2일 56분
MINSLK, RSM, MINLFT	P1	7/13, 19:06	7/15, 16:02	1일 20시간 56분
	P2	7/12, 10:21	7/15, 06:19	2일 19시간 48분
	P3	7/10, 06:22	7/11, 21:30	1일 15시간 8분
	P4	7/10, 07:26	7/12, 08:22	2일 56분
SASP	P1	7/13, 19:06	7/13, 19:06	0
	P2	7/12, 10:21	7/15, 10:33	3일 12분
	P3	7/10, 06:22	7/10, 23:41	17시간 19분
	P4	7/10, 07:26	7/12, 07:18	1일 23시간 52분
LALP	P1	7/13, 19:06	7/13, 19:06	0
	P2	7/12, 10:21	7/14, 07:54	1일 21시간 33분
	P3	7/10, 06:22	7/13, 04:31	2일 22시간 9분
	P4	7/10, 07:26	7/14, 06:41	3일 23시간 15분
MINSLK	P1	7/13, 19:06	7/15, 16:02	1일 20시간 56분
	P2	7/12, 10:21	7/15, 06:19	2일 19시간 58분
	P3	7/10, 06:22	7/11, 21:30	1일 15시간 8분
	P4	7/10, 07:26	7/12, 08:22	2일 52분
MLPF	P1	7/13, 19:06	7/13, 19:06	0
	P2	7/12, 10:21	7/13, 23:26	1일 13시간 5분
	P3	7/10, 06:22	7/12, 20:38	2일 14시간 16분
	P4	7/10, 07:26	7/10, 21:10	13시간 44분

제안 알고리즘에 의하여 산출한 최종 실험결과를 비교분석해 보면 다음과 같은 사실을 도출해낼 수 있다. 즉 기존의 시스템에서 현재 적용하고 있는 ES, TF, DU, RN 규칙에 의하여 산출된 전체 프로젝트의 일정의 길이의 합은 77, 495분인 반면에, 동일한 규칙을 적용한 신규의 제안알고리즘에 의하여 산출된 전체 프로젝트의 일정의 길이의 합은 71, 770분으로서 제안 알고리즘이 기존 알고리즘에 비하여 전체 일정이 5,725분 적게 소요된다.

ES, TF, DU, RN 규칙을 적용할 경우 이를 프로젝트별로 비교해 보면 <표 5>와 같다.

<표 5>에서 알 수 있듯이, 제안 알고리즘을 적용한 결과 전 프로젝트에서 일정이 단축되었으며 프로젝트 완료시점의 최소화 측면에서도 전체 프로젝트가 1,219분 빨리 완료된다. 또

표 5. 알고리즘 적용 결과의 비교

프로젝트	일정의 합(분)			평균 부하율(%)		
	기존알고리즘	제안알고리즘	차이	기존알고리즘	제안알고리즘	차이
P1	21,861	20,059	1,802	32.4	35.5	3.1
P2	21,038	20,642	396			
P3	17,603	15,207	2,396			
P4	16,993	15,862	1,131			
계	77,495	71,770	5,725			

한 평균부하율도 3.1% 증가하였다. 이것은 동일한 우선순위 규칙을 적용하였기 때문에 우선순위 규칙의 적용 효과는 포함되지 않은, 부하평준화의 가능 및 로직(logic)의 차이에서 오는 개선된 결과이다. 즉 제안 알고리즘의 성능(performance)이 기존 알고리즘의 성능보다 우수하다는 것을 입증하는 것이다.

이러한 결과는 기존 알고리즘의 운영상의 문제점인 특정 활동의 심한 일정지연(jumping)의 방지를 위하여 활동(작업) 분할이 가능하도록 하였고, 또하나의 문제점인 유저자원의 발생을 최소화시키는 동시에 일정지연을 억제시키기 위하여 작업장 간 차입이 가능하도록 한데 기인한 것이다.

5. 결론 및 추후 연구과제

본 연구에서는 복수 프로젝트의 자원제약 일정계획 문제에 실제로 적용 가능한 알고리즘을 개발하였으며, 제안 알고리즘의 성능을 평가하기 위하여 기존 알고리즘에 의한 일정계획과 신규로 개발한 제안 알고리즘에 의한 일정계획을 프로젝트 완료시점 및 평균부하율 측면에서 실험을 통하여 비교 분석하였다.

그 결과 프로젝트 완료시점, 전체 프로젝트 소요기간의 합계, 평균부하율, 최장 프로젝트 일정의 길이 등 어떠한 기준에 의하여 일정을 평가하더라도 제안 알고리즘이 기존 알고리즘보다 성능이 우수하다는 것이 입증되었다.

제안 알고리즘은 프로젝트 일정계획문제에 실제로 적용이 가능하며 작업순서가 정해져 있는 개별공정형 모형 등 다른 모형에도 적용이 가능하다. 원칙적으로 프로젝트의 수, 활동의 수, 자원유형의 수 등에 대한 제한은 없다. 이것은 오히려 데이터베이스 관리상의 문제에 더 가깝다. 그러나 네트워크의 복잡도, 활동의 수, 자원 유형 수 등이 증가할수록 휴리스틱의 성능에는 나쁜 영향을 미칠 수 있다.

본 연구에서는 비교적 소규모의 문제에 대하여 휴리스틱의 성능을 실험하였으나 아주 많은 프로젝트 수와 활동 수를 가지는 대규모 문제를 대상으로 실제 데이터에 의한 다양한 실험을 하지 못하였는데, 이것은 향후의 연구과제로 남는다.

참고문헌

배상운, 김여근(1996), 납기를 갖는 Job Shop 일정계획의 효율적인 발견적 기법, *한국경영과학회지*, 21(3), 76.
 유재건(1997), 유연자원 할당 및 자원제약하의 일정계획을 위한 발견적 알고리즘, *대한산업공학회지*, 23(2), 434.
 이순용(1991), *생산관리론*, 법문사, 521.
 이재동, 홍유신(1997), 블록조립공장의 부하평준화를 위한 생산일정계획, *산업공학*, 10(3), 155-166.
 Bell, C. E. and Han, J.(1991), A New Heuristic Solution Method in Resource-Constrained Project Scheduling, *Naval Research Logistics*, 38, 315-331.
 Burgess, A. R. and Killebrew, J.B.(1962), Variation in Activity Level on a Cyclical

Arrow Diagram, *Journal of Industrial Engineering*, 13(2), 76.
 Davis, E. W. (1973), Project Scheduling under Resource Constraints Historical Review and Categorization of Procedures, *AIIE Transactions*, 5(4), 297-313.
 Davis, E. W. and Patterson, J. H. (1975), A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling, *Management Science*, 21(8), 944-955.
 Kurtulus, I. and Davis, E. W. (1982), MultiProject Scheduling : Categorization

of Heuristic Rules Performance, *Management Science*, 161-172.
 Leachman, R. C. (1993), Multiple Resource Leveling in Construction System Through Variation of Activity Intensities, *Naval Research Logistics Quarterly*, 30, 187-198.
 Levy, F.K., Thompson, G.L. and Wiest, J.D. (1962), Multiship, Multishop, Workload Smoothing Program, *Naval Research Logistics Quarterly*, 9, 37-44.
 Lucas Management Systems (1993), *Artemis 7000 Quick Reference Guide*.
 Wiest, J.D. (1967), A Heuristic Model for Scheduling Large Projects with Limited Resources, *Management Science*, 13(6), B369-B377.



공명달

동아대 산업공학과 공학사
 동아대 산업공학과 공학석사
 동아대 산업공학과 공학박사
 산업체 실무경력 16년
 현재: 영산대학교 경영정보학과 재직
 관심분야: 경영정보시스템 (MIS), 소프트웨어공학, 시스템분석 및 설계, CIM, BPR



김정자

한양대 공업경영학과 공학사
 일본 와세다대학 공업경영학과 공학석사
 일본 와세다대학 공업경영학과 공학박사
 현재: 동아대학교 산업시스템공학과 정교수
 관심분야: 재고·물류관리, 최적화