

프로젝트 스케줄링을 위한 데이터베이스시스템의 설계 및 구현

한재민, 황인수

고려대학교 경영학과

요 약

생산시스템에서의 스케줄링에 관한 연구는 단일기계작업 스케줄링으로부터 유연생산시스템(FMS)에 이르기까지 매우 광범위하게 이루어져 왔으며, 정수계획법, 동적계획법, Branch and Bound, PERT/CPM, 그리고 기타의 휴리스틱 기법 등 많은 연구결과를 찾아볼 수 있다. 그러나, 대부분의 연구는 소규모의 문제를 대상으로 이루어진 것으로 대규모의 현실세계 문제를 풀기에는 부적절한 것으로 판명되었다. 또한, 적용이 가능한 휴리스틱(Heuristic) 기법들도 현실세계와는 맞지 않는 많은 가정하에서 주어진 목적함수를 극대화하도록 개발되었기 때문에 현실적인 문제를 풀기 위해서는 각종 가정을 완화시킴으로써 휴리스틱을 현실에 적용하기 위한 또 다른 연구가 필요하게 되었다.

따라서, 이 연구에서는 새로운 휴리스틱을 개발하기보다는 기존의 휴리스틱 기법을 그대로 이용하면서 현실세계의 문제에서 발생하는 동적(Dynamic) 스케줄링과 재스케줄링(Rescheduling) 문제를 해결하는 시스템을 설계, 구현하는데 초점을 맞추고 있다. 특히, 데이터베이스 접근법을 사용함으로써, 가용자원의 수가 증감하거나 새로운 작업이 추가되는 등 기존의 가정을 상당부분 완화시킨 동적스케줄링환경하에서도 재스케줄링이 가능한 동적 재스케줄링(Dynamic Rescheduling) 시스템을 설계, 구현하였으며, 이의 세부기능을 보완하는 작업이 계속되고 있다.

1. 서 론

스케줄링(Scheduling)이란 주어진 목표를 달성하기 위해 자원을 할당하는 일련의 과정 [4], 또는 필요한 활동들을 할당하며 자원을 통제하는 일련의 의사결정과정 [7]으로 정의 할 수 있으며, Job Shop 스케줄링, 유연생산시스템(FMS) 스케줄링, 로봇(Robot) 스케줄링, 인적자원 스케줄링 등 여러 가지 문제영역에 대해 적용되어 왔다.

일반적으로 스케줄링 문제는 자원제약, 작업간의 선후관계 등과 같은 여러 가지의 제약조건하에서 문제를 풀어야 하며, 이와 같은 제약조건을 고려하지 않은 스케줄링 문제는 현실적으로 아무런 의미를 갖지 못하게 된다. 그러나 한편으로는 이러한 제약조건들로 인하여 탐색공간(Search Space)이 현저히 줄어들기 때문에, 보다 효율적으로 원하는 해를 찾을 수도 있다. 근래에는 서로 다른 여러 가지의 자원을 필요로 하는 요소작업들로 이루어진 일련의 작업을 스케줄링하는 자원제약(Resource-Constrained) 조건하에서의 스케줄링에 관한 연구가 활발히 진행되고 있다.

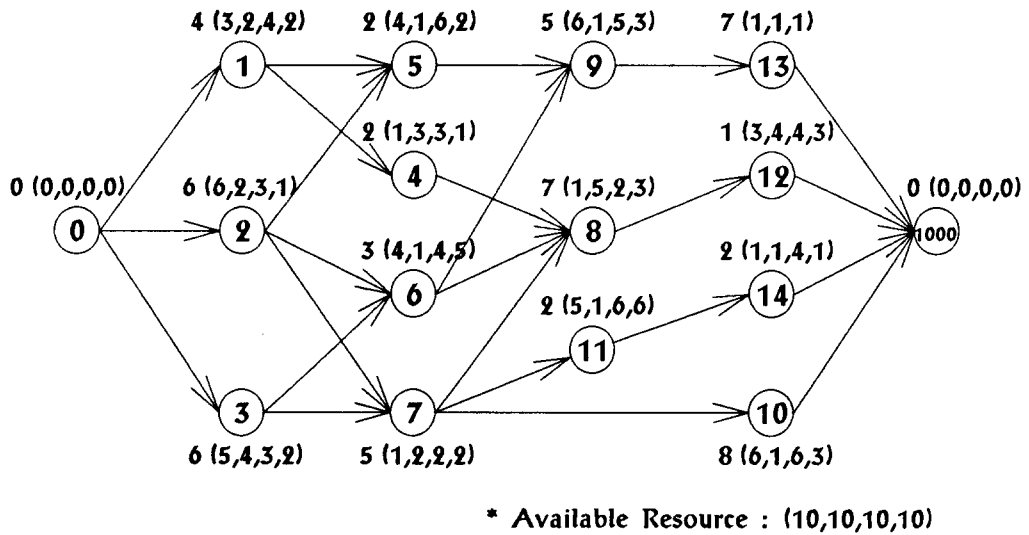
1950년대 말 이후로, 이와 같은 자원제약조건하에서의 스케줄링을 위해 정수계획법(Integer Planning, IP), 동적계획법(Dynamic Programming, DP), Branch and Bound, PERT/CPM 기법, 그리고 여러 가지의 휴리스틱(Heuristic) 기법 등 많은 스케줄링 접근법이 적용되어 왔다. 이들 각 기법에 대해 많은 연구결과가 발표되어 온 것이 사실이나, 휴리스틱기법을 제외한 나머지 접근법들은 대부분이 현실의 문제를 풀기에는 부적절한 것으로 나타났다. 이는 현실적으로 존재하는 대부분의 스케줄링 문제가 NP-Complete, 즉, 문제의 크기가 증가함에 따라 탐색해야 하는 탐색공간이 지수적으로 증가하는 특성을 갖고 있기 때문이다. 따라서, 최적해를 찾기 위해 사용되는 분석적인 기법이나 알고리즘은 점점 더 복잡해지는 현실적인 문제에는 더 이상 적용될 수 없는 것으로 나타나고 있다. 결국 이 후의 대부분의 연구는 스케줄링 문제를 보다 짧은 시간내에 보다 효율적으로 풀기 위한 휴리스틱기법을 개발하는 데 초점이 맞추어졌다. 아울러, 자원제약조건하에서의 스케줄링을 위해 개발된 휴리스틱기법들은 여러 가지의 휴리스틱함수를 이용한 할당(Priority Dispatching) 방식을 사용하였다. 이들 휴리스틱 기법들 중에 작업의 종료시간과 관계 있는 성과척도(Performance Measure)를 갖는 문제를 풀기 위한 기법들은 대부분이 PERT/CPM 네트워크에서 구할 수 있는 임계경로(Critical Path)를 이용하고 있다.

Davis는 여러 가지의 휴리스틱 기법들을 분석한 결과, 특정 휴리스틱 기법이 다른 휴리스틱기법에 비해 보다 효과적 혹은 효율적일 수는 있으나, 이들 대부분도 현실적인 문제를 풀기에는 부적절하다는 결론을 내린 바 있다 [6]. 아울러, 현재 개발되고 있는 휴리스틱 기법들이 현실문제와는 관계없이 서로 다른 환경조건하에서 기존의 다른 기법들보다 주어진 하나의 성과척도에 대해 보다 나은 성과를 나타내는 기법을 개발하는 데 모든 노력을 기울이고 있기 때문에 현실적으로 적용하기에는 부적절한 면이 있음을 인정할 수 밖에 없다. 그러므로, 현실에 적용이 불가능한 연구를 계속하기 보다는 기존의 연구를 바탕으로 현실적인 적용에 대한 연구가 더 깊게 이루어져야만 할 것이다.

따라서, 이 연구에서는 현실적으로 가장 근접한 문제성격을 갖고 있는 프로젝트 스케줄링을 위한 시스템을 개발함에 있어서, 관계형 데이터베이스관리시스템을 이용함으로써 현장의 스케줄링 환경에 보다 능동적으로 대처하여 재스케줄할 수 있는 방안에 대해 연구하였다. 또한, 시스템으로 구축되기 위해서 필요한 데이터베이스 파일의 구성, 네트워크에 대한 각종 사전검사, 초기스케줄 및 재스케줄 작성, 스케줄 결과의 확인 등에 대한 전반적인 틀을 제공하는 것을 목적으로 하고 있다.

2. 프로젝트 스케줄링의 개념

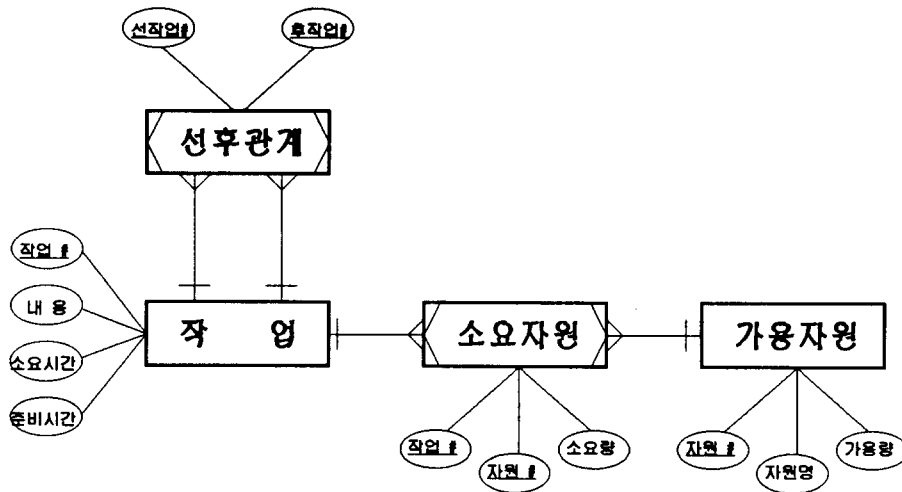
프로젝트 스케줄링은 선후관계를 갖는 일련의 작업으로 이루어지는 프로젝트(Project)를 가용자원을 초과하지 않는 범위에서 스케줄링하는 일련의 과정을 말한다. 일반적으로 스케줄링 문제는 각 세부작업에 관한 기본데이터, 작업간의 선후관계, 가용자원, 작업별 소요자원 등에 대한 각종 데이터를 필요로 하며, [그림 1]과 같은 구조를 갖는다.



[그림 1] 프로젝트 스케줄링의 예

3. 프로젝트 스케줄링을 위한 데이터베이스 설계

위의 스케줄링 문제를 컴퓨터의 데이터베이스 시스템을 이용하여 처리하기 위해서는 데이터 모델링 (Data Modeling) 기법을 이용하여 체계화하여야 하는데, 대표적인 데이터 모델링 기법의 하나인 개체 관계도 (Entity-Relationship Diagram, ERD) 를 이용하여 표현하면 [그림 2]와 같다.



[그림 2] 프로젝트 스케줄링 개체관계도

위의 개체관계도 (ERD)를 바탕으로 관계형 데이터베이스시스템을 구축하기 위해서는 다음과 같이 각 개체 (Entity)가 갖는 속성 (Attribute)을 구체적으로 정의하고, 각각을 하나의 데이터파일로 구축해야 한다.

1) 작업 (JOB) 데이터파일

● <u>작업번호</u> (JobNo)	N (5)	Primary Key
● 작업내용 (Descript)	C (15)	
● 소요시간 (Processing)	N (5)	
● 남은시간 (WorkRemain)	N (5)	
● 방출시간 (Release)	N (5)	
● 작업 중 (InProgress)	L (1)	
● 시작시간 (Start)	N (5)	
● 종료시간 (End)	N (5)	
● 종료여부 (Finish)	L (1)	
● 선작업수 (Pred)	N (5)	

2) 선후관계 (ARC) 데이터파일

● <u>선행작업</u> (JobFrom)	N (5)	Primary Key
● <u>후속작업</u> (JobTo)	N (5)	Primary Key

3) 가용자원 (RESOURCE) 데이터파일

● <u>자원번호</u> (ResourceID)	C (5)	Primary Key
● 자원명 (Name)	N (15)	
● 가용량 (Available)	N (5)	
● 작업율 (Rate)	N (5)	

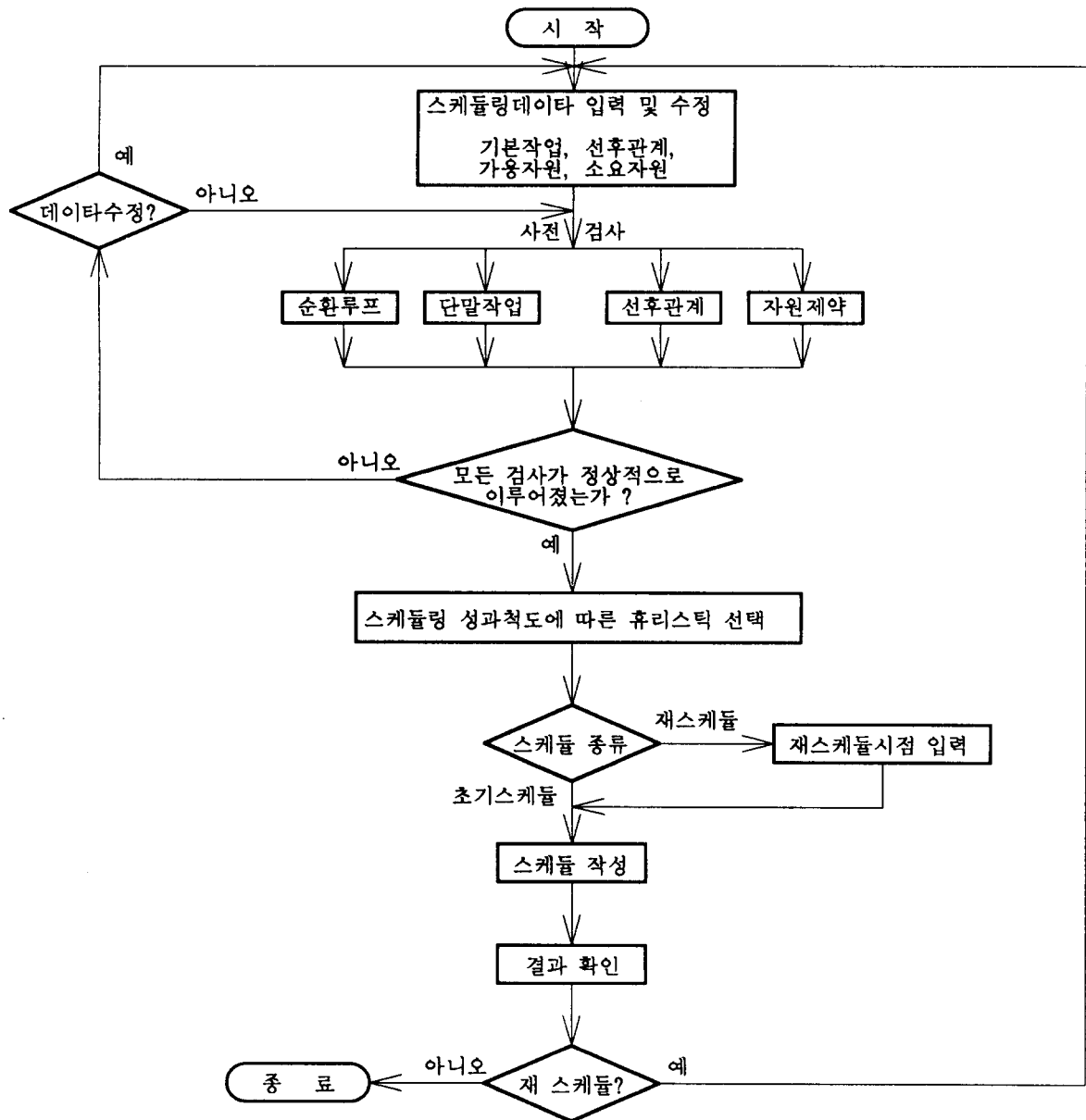
4) 소요자원 (REQUIRE) 데이터파일

● <u>작업번호</u> (JobNo)	N (5)	Primary Key
● <u>자원번호</u> (ResourceID)	C (5)	Primary Key
● 소요량 (Required)	N (5)	

4. 데이터베이스 시스템 설계 및 구현

4.1 프로젝트 스케줄링 절차

프로젝트 스케줄링은 관련 데이터의 입력 및 수정, 사전검사, 스케줄링, 결과 확인 등의 절차를 거쳐 이루어지는 데, 이를 흐름도 (Flow Chart)로 표현하면 [그림 3]과 같다.



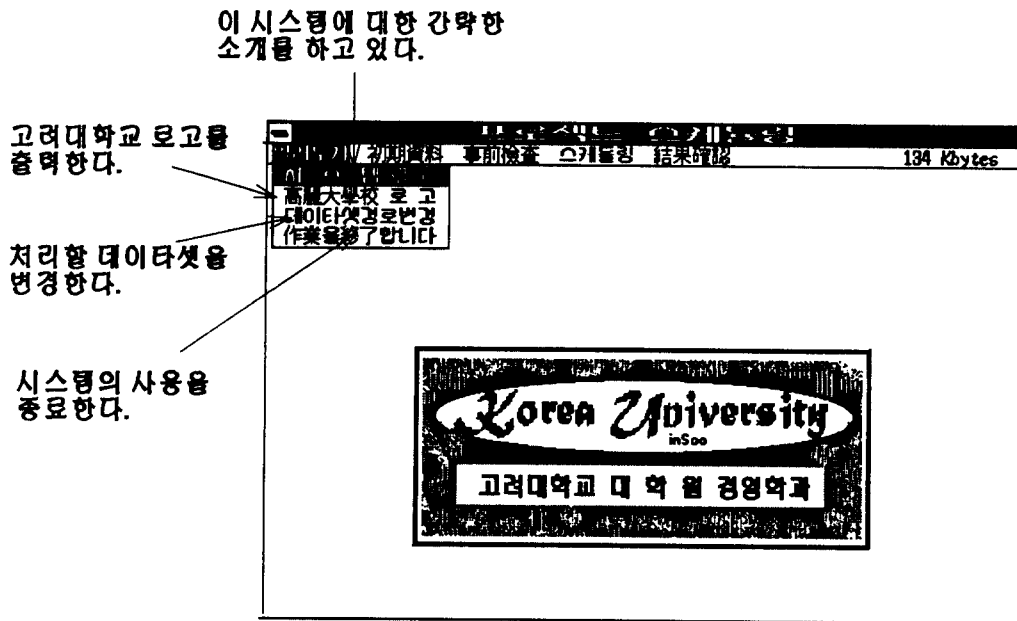
[그림 3] 프로젝트 스케줄링을 위한 흐름도

4.2 메뉴 시스템

이 시스템은 기본적으로 프로젝트 스케줄링을 하기 위한 일련의 작업을 목적으로 하고 있기 때문에, 시스템에 대한 소개, 기초자료관리, 사전검사, 스케줄링, 결과 확인 등의 내용으로 구성되어 있다.

1) 일러두기

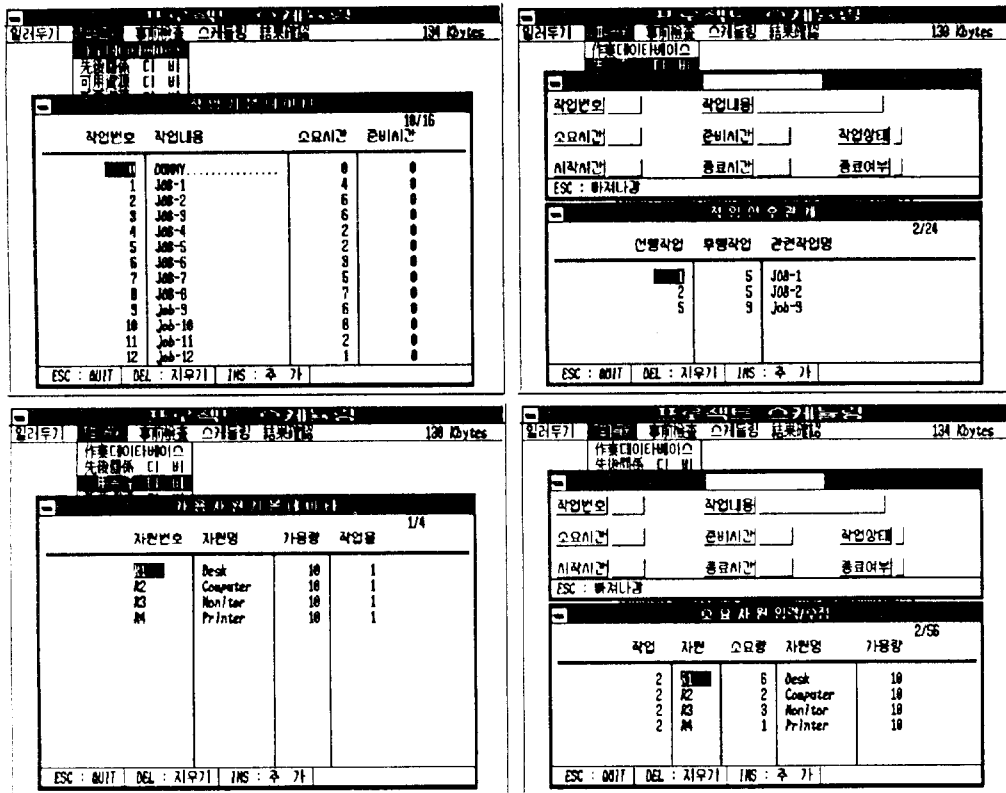
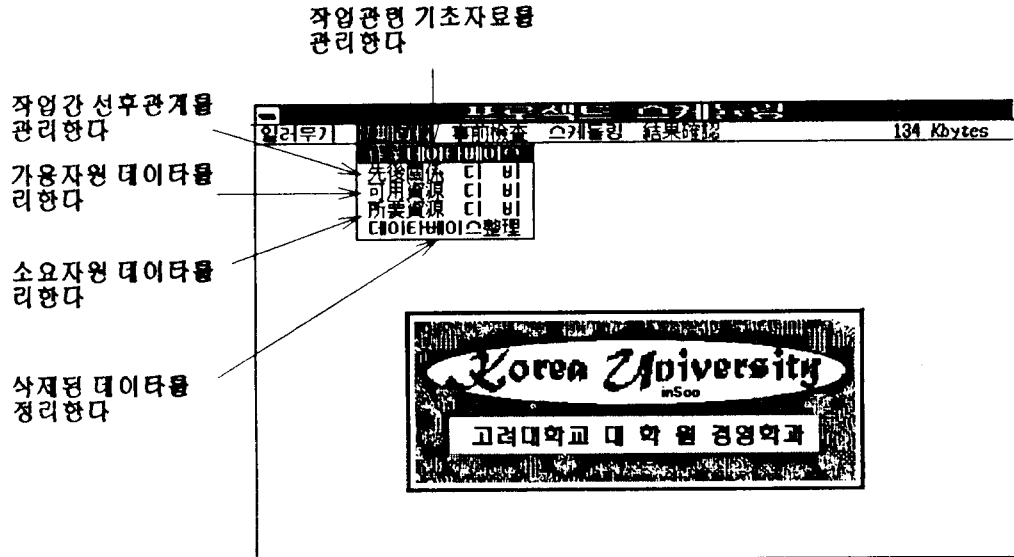
이 시스템에 대한 간략한 소개와 함께 사용법, 프로젝트의 선택, 시스템 사용의 종료 등을 하부메뉴로 갖고 있다.



[그림 4] 프로젝트 스케줄링시스템의 주메뉴 화면

2) 기초자료

프로젝트 스케줄링을 위한 각종 기초자료를 처리하는 부분으로, 프로젝트에 속한 각 작업에 대한 데이터, 작업간 선후관계 데이터, 가용자원 데이터, 그리고 각 작업을 수행하기 위해 필요한 소요자원 데이터 등을 관리한다.

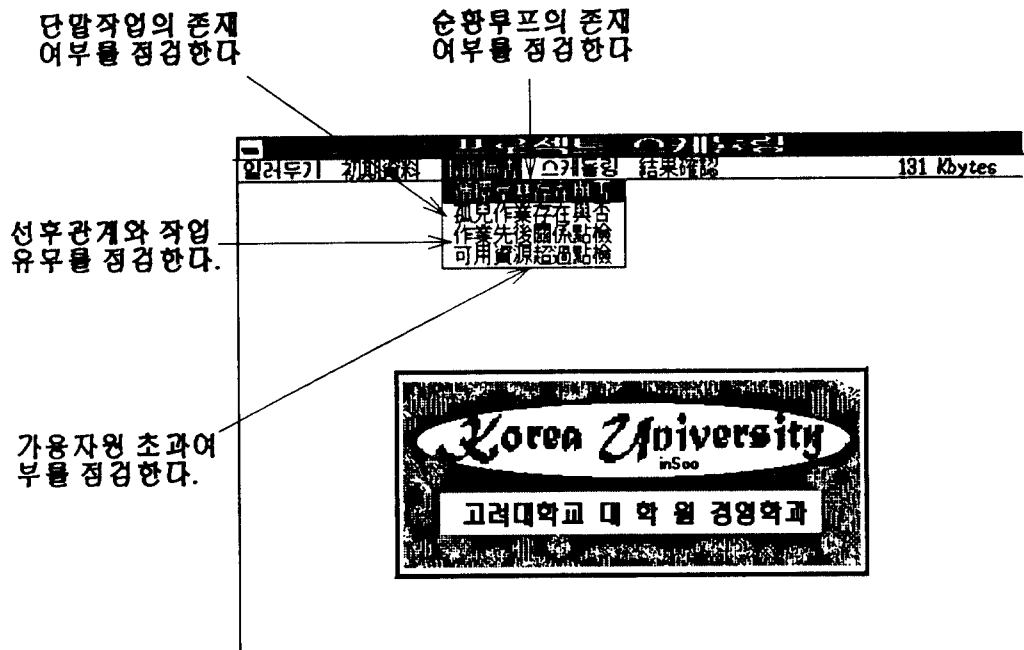


[그림 5] 프로젝트 스케줄링을 위한 기초자료 입력화면

3) 사전검사

사전검사는 스케줄을 만들기 전에 프로젝트가 정상적으로 수행될 수 있는지의 여부를 점검하는 과정으로, 일반적으로 그래프 이론을 이용하며 다음과 같은 사전검사를 수행한다.

- 순환루프 검사 : 순환루프란 네트워크상에서 하나 이상의 작업이 선후관계를 통하여 순환(Circulation)을 이루게 되는 현상으로, 네트워크에 순환루프가 존재하면 스케줄을 만들 수 없다. 이 시스템에서는 선행 혹은 후속 작업이 없는 작업들로부터 차례대로 제거함으로써 순환루프의 존재여부를 검사하는 매우 효율적인 방법을 사용하였다.
- 단말작업 검사 : 단말작업이란 전체 네트워크와는 무관하게 별도로 독립되어 있는 단 하나의 작업 또는 일련의 작업 중에서 Initial 또는 Terminal 작업을 의미하는 것으로, 정상적인 네트워크에서는 각각 하나의 Initial 작업과 Terminal 작업을 갖는다. 이 시스템에서는 선행 혹은 후속작업의 존재유무를 이용하여 단말작업을 식별하고 있다.
- 작업선후관계 검사 : 작업선후관계 데이터파일을 점검하여 선후관계에는 정의되어 있으나, 실제작업이 정의되지 않는 오류가 없는지 점검하며, 이러한 오류가 발생한 경우에는 스케줄을 만들 수 없다. 이 시스템에서는 작업선후관계 데이터베이스와 작업 데이터베이스를 이용하여 작업선후관계를 검사한다.
- 가용자원 초과여부 검사 : 특정 작업을 위해 소요되는 자원의 양이 총 가용자원을 초과하는 경우에는 스케줄을 만들 수 없으므로, 각 작업별로 소요자원의 양이 가용자원을 초과하는지의 여부를 점검한다. 이 시스템에서는 작업별 소요자원 데이터베이스와 가용자원 데이터베이스를 이용하여 가용자원 초과여부를 검사한다.

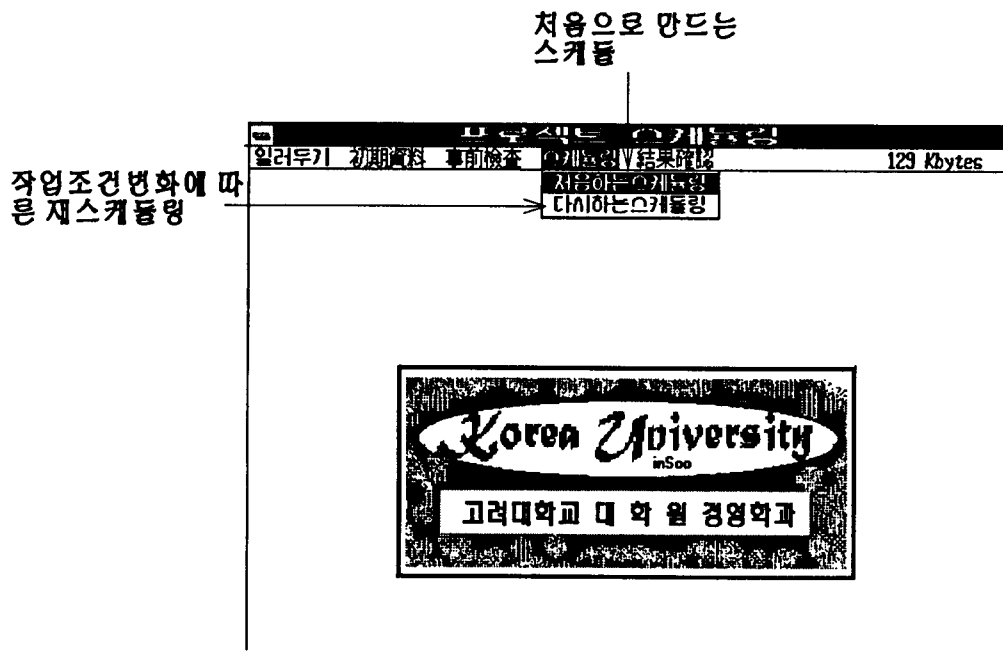


[그림 6] 프로젝트스케줄링을 위한 사전검사화면

4) 스케줄링

스케줄링은 처음으로 만드는 스케줄과 작업조건이 변함에 따라 새로운 환경하에서 다시 스케줄을 만드는 재스케줄링 (Rescheduling)으로 구분된다. 일반적으로 스케줄링은 주어진 시점에서 선행작업이 모두 끝나고 작업이 이미 준비되어 있는 작업들을 찾아낸 후, 이들 작업중에서 주어진 목적함수를 가장 잘 충족시킬 것으로 예상되는 작업 혹은 작업들을 할당하는 일련의 과정으로 이루어진다.

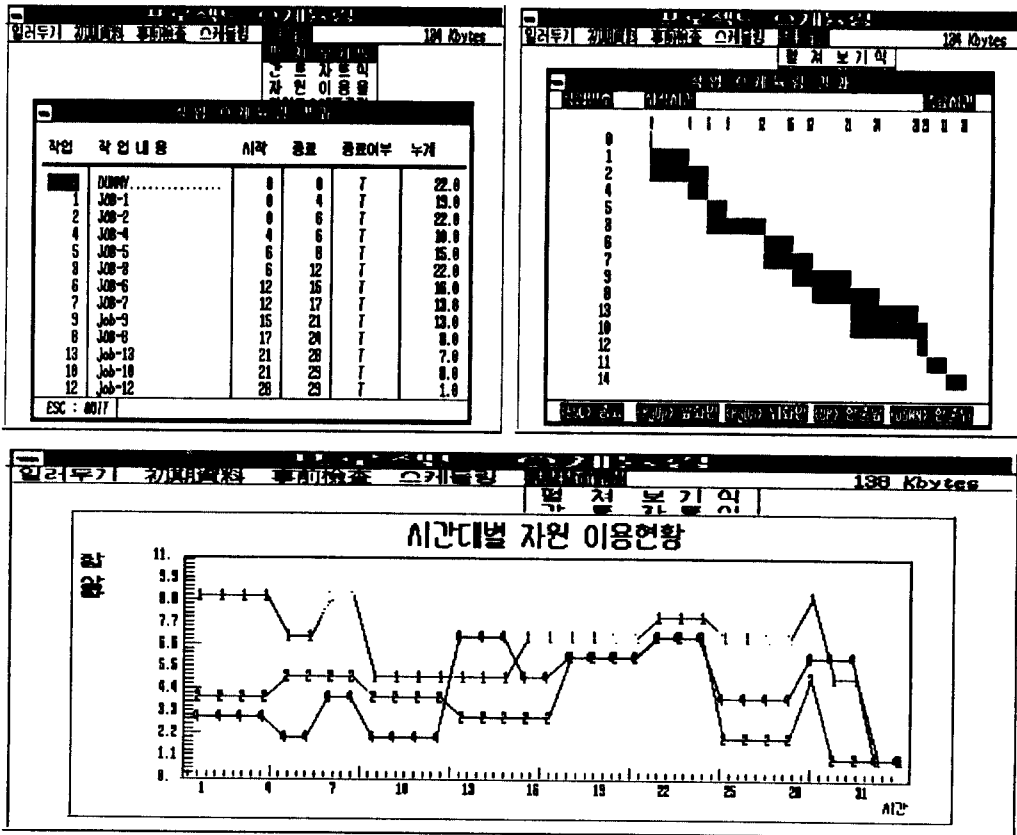
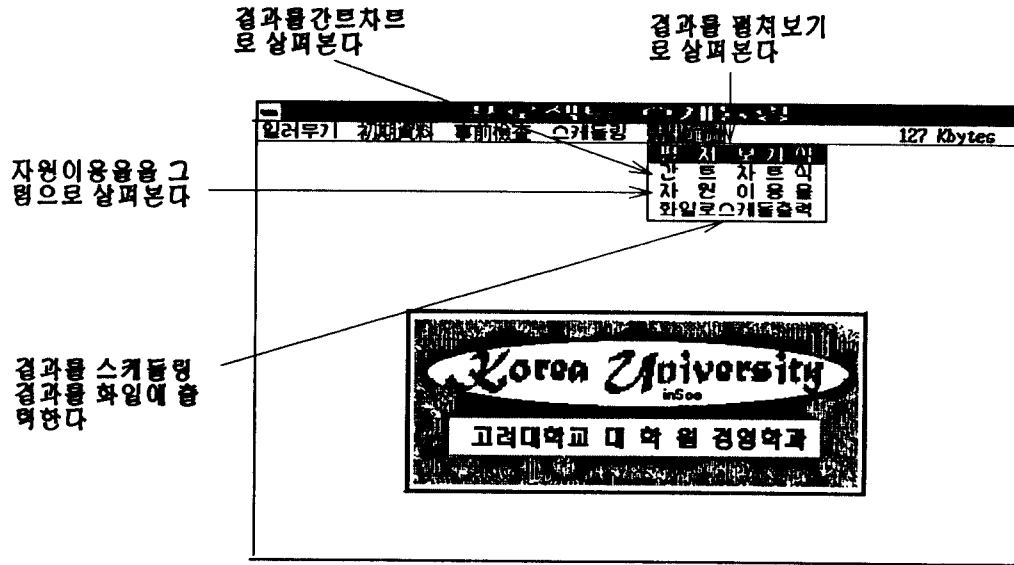
- 스케줄 가능한 작업셋 선택 : 특정 작업이 스케줄 가능하기 위해서는 해당작업이 이미 준비 (Ready) 상태이며, 선행하는 작업들이 모두 종료되어 있어야 한다. 따라서, 데이터베이스의 필터링 (Filtering) 및 인덱싱 (Indexing)기능을 이용함으로써 보다 효율적으로 해당하는 작업들을 선택할 수 있다.
- 스케줄 가능한 작업의 모든 조합을 나열 : 프로젝트 스케줄링이 갖는 특징의 하나는 동일한 시점에 자원이 허락하는 한 서로 다른 작업을 동시에 수행할 수 있다는 것이다. 따라서 앞에서 찾아낸 스케줄 가능한 작업셋들의 모든 조합 (Combination)으로부터 가장 적합한 작업들을 선택해야 하는데, 여기에서는 되부름 (Recursion)을 이용하였다.
- 선택된 작업을 할당 : 위의 각 작업집합들 중에서 주어진 목적함수를 가장 잘 충족시킬 것으로 예상되는 작업을 선택하여 자원 (Machine)을 할당한다. 이 때, 선택을 위해서는 이미 개발되어 있는 많은 알고리즘이나 휴리스틱 기법들을 그대로 사용할 수 있으며, 아울러 인공지능 기법을 이용하여 주어진 문제의 성격과 성과척도에 따라 가장 적합할 것으로 예상되는 휴리스틱을 선택하여 적용하는 방안에 대한 연구를 계속하고 있다.



[그림 7] 프로젝트 스케줄링을 위한 스케줄메뉴 화면

5) 스케줄링 결과 확인

스케줄링이 끝나면, 각 작업이 시작되는 시간과 종료되는 시간에 대한 정보를 테이블이나 간트 차트로 보여주며, 아울러 각 시점에서 자원의 이용율은 꺾은선 그래프의 형태로 보여준다.



[그림 8] 프로젝트 스케줄링 결과 화면

5. 결론 및 향후 연구계획

이 연구에서는 스케줄링에 대한 이론적인 연구를 현실세계에 적용하기 위해서는 데이터베이스 시스템이 반드시 필요함을 강조하고, 데이터베이스시스템 접근법을 사용할 경우에 필요한 처리절차에 대한 개념적 틀을 제공하는 것을 목적으로 하였다. 일반적인 스케줄링 알고리즘이나 휴리스틱에 대한 연구에서는 작업의 종류, 개수, 소요자원의 량, 가용자원의 양, 작업간 선후관계 등이 사전에 알려져 있음을 가정하고, 이 때 주어진 목적함수를 극대화하는 스케줄을 찾는다. 그러나, 이와 같은 환경은 실제로 존재하지 않는다고 해도 과언이 아니며, 작업시간은 작업조건에 따라서 수시로 변경될 뿐만 아니라, 필요에 따라서는 새로운 작업이 추가되기도 하고 또는 기존의 작업이 취소되기도 한다. 이러한 환경하에서는 기존의 알고리즘이나 휴리스틱은 아무런 의미를 갖지 못하게 된다. 따라서, 현장에 적용이 가능한 학문이 되기 위해서는 이와 같은 재스케줄에 대한 연구가 더 많이 이루어져야 하나, 이러한 연구는 거의 찾아 볼 수가 없다.

결국, 현실적인 문제에 적용하기 위한 스케줄링시스템은 현재의 이론적인 배경만으로는 구현될 수 없으며 새로운 패러다임을 이용하여야만 하는데, 이 연구에서 사용한 방법은 사용자들에게 가장 친근감 있는 환경을 제공할 수 있는 데이터베이스시스템을 이용하여 스케줄링 시스템을 설계, 구현하는 것이다. 관계형 데이터베이스시스템이 갖는 고유한 특성으로 인하여 처리속도의 저하를 감수하여야 하나, 데이터베이스시스템이 특정 레코드만을 보여주는 필터링, 검색속도를 향상시켜 주는 인덱싱 기능, 그리고 스케줄링 환경이 바뀔 경우 데이터베이스의 내용을 간단히 수정한 후에 재스케줄하는 사용자 인터페이스를 제공하므로, 데이터베이스의 이용에 따른 장점을 충분히 살릴 수 있을 것이다.

결론적으로, 이 연구는 스케줄링에 대한 기존의 많은 연구들을 데이터베이스라는 도구를 이용하여 현실세계에 적용하기 위해 필요한 개념적인 틀과 절차를 제공하고 있다. 데이터베이스의 질의어(Query language)가 갖는 장점을 살려 보다 편리하게 스케줄링시스템을 이용하며, 스케줄링 환경의 변화에 따른 동적 재스케줄링(Dynamic Rescheduling)이 가능한 시스템을 설계, 구현했다는 점에서 이 연구의 의의를 찾을 수 있을 것이다. 향후의 연구에서는 보다 유연하게 현장의 문제에 적용될 수 있도록 시스템을 개선할 것이며, 비용개념을 도입하여 성과척도로 사용하게 될 것이다. 아울러, 계속되는 연구를 통하여 기존의 연구에서 얻어진 각종 알고리즘 및 휴리스틱을 모듈로 삽입하며, 인공지능분야의 분류 및 학습기능을 첨가하여 새로운 스케줄링 문제를 만날 때 주어진 목표에 가장 잘 부합하는 휴리스틱을 자동적으로 선정하여 스케줄을 작성하는 시스템에 대한 연구가 이루어질 것이다.

6. 참고 문헌

- [1] 아성소프트, 「아성라이브리 3.0 기능설명서」, 1994.
- [2] 황인수, 「클리퍼 5.2 유틸리티」, 정보문화사, 1994.
- [3] 황인창, 「클리퍼 바이블」, 정보문화사, 1992.
- [4] Baker, K. R., *Introduction to Sequencing and Scheduling*, Wiley, New York, 1974.
- [5] Computer Associates, *CA-Clipper Reference Guide (Version 5.2)*, 1994.
- [6] Davis, E. and J. Patterson, "A Comparison of Heuristic and Optimal Solutions in Resource-Constrained Project Scheduling," *Management Science*, Vol. 21(1975), pp. 944-955.
- [7] Han, J., "From Automated Planners to Intelligent Schedulers," *The Journal of Management*, Korea Univ., Vol. 35(1991), pp. 181-200.
- [8] McFadden, F.R. and J.A. Hoffer, *Database Management*, 3rd ed., Benjamin/Cummings, 1991.
- [9] Rich, E. and K. Knight, *Artificial Intelligence*, McGraw-Hill, 1991.