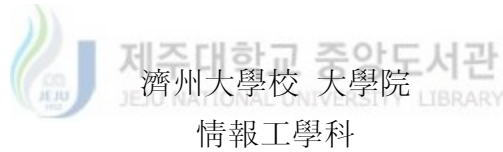


碩士學位論文

경험적 탐색기법을 이용한
시간표 자동화 시스템 개발에 관한 연구



金 京 熙

1998年 12月

경험적 탐색기법을 이용한
시간표 자동화 시스템 개발에 관한 연구

指導教授 金 壯 亨
金 京 熙

이 論文을 工學 碩士學位 論文으로 提出함

1998年 12月



金京熙의 工學 碩士學位 論文을 認准함

審査委員長 印

委 員 印

委 員 印

濟州大學校 大學院

1998年 12月

A Study on Development of a Timetable
Automatic System using the Heuristic Search
Method

Kyung-Hee Kim

(Supervised by professor Jang-Hyung Kim)



A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
ENGINEERING

DEPARTMENT OF INFORMATION ENGINEERING
GRADUATE SCHOOL
CHEJU NATIONAL UNIVERSITY

1998. 12.

목 차

Summary	1
I. 서론	2
II. 이론적 배경	
1. 탐색	
1) 맹목적 탐색	5
2) 경험적 탐색	7
2. 제약 조건	10
3. 자료 흐름	10
III. 시스템 구성 및 기능	
1. 구현 환경	13
2. 이벤트 목록	14
3. 프로세스 계층	15
4. 지식 데이터베이스 구축	17
5. 최적해 탐색	18
6. 클라이언트 어플리케이션	23
IV. 실험 및 고찰	
1. 모의 실험	
1) 모의 데이터	24
2) 적용1	28
3) 적용2	30
4) 적용3	30
5) 수작업 시간표	33
2. 비교 분석	34



V. 결론 36

참고문헌 38



Summary

A college's most essential obligation is to fulfill its educational function. This educational function can be fully achieved in the classroom.

The three necessary components for a lesson - students, an instructor and a classroom - are intricately interwoven. Due to this complexity, it has been difficult to computerize a timetable preparation program.

This paper uses the concept of Problem Solving to draw up a timetable through which a lesson can be efficiently taught and all conditions desired for the above-mentioned three components can be maximally satisfied.

This program is operate on the basis of "Search Method" which finds the optimal solution through a process of trial and error in cyberspace where as many solutions as possible are tested by the Problem Solving method.

This paper also provides an efficient and optimal timetable that meets the conditions of the three components in the following two ways. One is to establish an evaluation function knowledge database on search path guide information through a breadth-first search which is a blind search method from the artificial intelligence basic search method. The other is to appropriately use an evaluation function which is set up by a best-first search, an empirical search method. Furthermore this system offers class time coordination and management convenience.

I. 서 론

대학의 여러 가지 기능 중 가장 중심이 되는 것은 교육기능이라 할 수 있다. 그리고 교육의 내실화는 학교 수업을 통해 이루어진다.

수업이 이루어지는데는 학생과 교수, 강의실 3가지 요소가 상호 복잡하게 연관되어 있다. 학생들은 수강하고 싶은 교과목을 수업시간 중복 없이 모두 이수할 수 있기를 바라고, 교수들은 수업 가능시간이 각기 달라 교수 개개인의 요구되는 시간에 강의를 할 수 있기를 바라는데 이러한 요구들을 충족하는데는 학생이 요구하는 시간과 교수의 수업시간 중복을 고려해야 하며, 수업이 이루어질 수 있는 강의실은 충분한가를 고려하여야 한다. 이렇게 학생·교수 개개의 아주 다양한 요구와 수업에 필요한 강의시설을 충족시키는데는 많은 제약조건들 즉 전체 학생이 선택 이수할 수 있는 교양선택 강의와 실험실습 등 연속강의가 필요한 특수교과목에 대한 시간분배, 학생과 교수의 수업시간 중복관계 등 고려해야 될 문제들이 수없이 많다. 이와 같은 수많은 제약 조건들을 수렴해야 하는 복잡한 문제들 때문에 시간표 작성의 전산화가 어렵다. 또한 이러한 이유 때문에 시간표 작성을 전산화해야 할 필요성이 요구된다.

시간표 자동화에 관한 프로그램은 경희대학교 인공지능연구실에서 개발한 것 외에 일부 몇 가지가 있으며, 학위논문으로는 한국해양대학 정 운성의 “시간표작성 전문가 시스템 구현에 관한 연구”와 공주대학교 박 경하의 “시간표평가 전문가 시스템의 구현에 관한 연구” 등이 있다(정,1995, 박,1997). 발표된 논문들은 시간표 자동화 부분이 미약하며 전체적인 통계를 구하는 선에서 평가가 이루어 졌고, 경희대학교 인공지능연구실에서 개발된 프로그램은 최적의 시간표를 산출해 내지만 개인 PC용으로 개발되어 기존 학사DB와의 연계가 안되어 자료관리의 어려움이 많다(경희대 인공지능연구실,1996).

본 논문에서는 학생·교수·강의실 3가지의 요소들이 요구되는 조건들을 최대한 충족하면서 수업이 효율적으로 이루어 질 수 있도록 하는 종합시간표 작성 시스템을 구축한다. 이 시스템에서는 3가지 요소가 요구되는 아주 다양한 조건과 그

요구조건들이 복합되어 복잡한 상관관계를 이루기 때문에 이러한 문제들을 해결하기 위해 문제풀이 개념을 이용하였다. 문제풀이 방식은 수많은 해들이 놓인 공간에서 시행착오적 방법으로 최적의 해를 찾는 탐색에 근거를 두고 해결해 나갈 수 있다(Kevin,1991).

본 논문에서는 이러한 인공지능의 기본 탐색방법중 하나인 맹목적탐색방법의 넓이 우선 탐색을 이용하여 탐색 경로를 안내할 수 있는 정보인 평가함수 지식 데이터베이스를 구축하여 경험적탐색방법의 최적 우선 탐색에서 구축된 평가함수를 적절히 사용하여 학생·교수·강의실 3요소의 요구조건이 최대한 수용되어 효율적이고 최적화된 시간표를 만들어 내고 있다. 또한 수업시간의 편성과 관리의 편리성도 제공하고 있다.

본 논문의 구성은 다음과 같다.

2장에서는 탐색방법의 이론적 배경을 살펴보고, 3장에서는 본 논문에서 제안하는 경험적 탐색방법을 이용한 종합수업시간표 자동 시스템을 설계하고, 4장에서는 본 시스템의 모의실험을 통하여 결과를 고찰하였다. 마지막으로 5장에서는 결론과 향후 연구방향을 제시하였다.



II. 이론적 배경

1. 탐색

인공지능의 대부분의 영역에서는 문제를 해결하기 위하여 탐색을 이용하고 있다. 예를 들어, 탐색은 여러 가지로 해석될 수 있는 단어의 문장 내에서의 의미를 파악하기 위해서, 주어진 질의어에 답하기 위하여 어떠한 규칙들을 먼저 부합시켜야 하는가를 결정하기 위해서, 목표지점까지 도달하기 위한 경로를 찾기 위해서, 원하는 정보를 어떠한 과정을 거쳐서 검색할 것인가를 판단하기 위해서 등, 인공지능의 거의 모든 부분에 공통적으로 관계되는 기초 분야라고 할 수 있다. 탐색을 위해서는 탐색공간을 정의하는 것이 필요한데, 탐색공간은 일반적으로 상태의 집합과 연산자의 집합으로 정의된다. 주어진 문제는 초기상태와 목표상태로 표현되며, 초기상태에 연산자를 적용시켜 다음 상태로 변환하면서 목표상태에 이르러 문제를 해결하게 된다. 여기서의 문제는 현재상태에서 어떠한 연산자를 적용하여 다음 상태로 갈 것인가를 결정하는 것이며 이것이 바로 탐색 문제의 핵심이 된다 (Luger,1989).

일반적으로 탐색은 주어진 문제영역에서 현재 상황에 적합한 연산자를 평가할 수 있는 함수가 존재하느냐에 따라 맹목적 탐색과 경험적 탐색으로 구분된다.

탐색을 구현하기 위해서 생각해야 할 문제는 어떠한 탐색 알고리즘을 사용할 것인가 하는 것 외에 어떠한 방법으로 탐색공간을 줄일 수 있는가 하는 점이다. 일반적으로 인공지능에서는 탐색공간의 크기가 주어진 문제의 크기(또는 입력자료의 크기)에 따라 지수 적으로 늘어나는 문제들을 그 대상으로 하고 있다. 이와 같은 문제의 접근방법은 좋은 평가함수 또는 탐색제어규칙을 사용하여 지수 적으로 늘어나는 탐색공간을 지수 적으로 줄이는 방법이다. 이러한 방법을 통해서 탐색공간이 비지수적으로 늘어나게 할 수는 없지만 해결 불가능한 문제를 해결가능한 문제로 변환시킬 수 있게 된다.

1) 맹목적 탐색

노드의 확장 순서가 목표의 위치에 대하여 무관한 맹목적 탐색은 평가함수를 정의하기 어려울 때 사용하는 방법으로서 노드를 생성된 순서에 따라서 확장시키는 것을 깊이우선탐색이며 가장 최근에 생성된 노드를 먼저 확장시키는 것을 넓이우선탐색이 그 대표적인 예이다(이와 김,1996).

(1) 깊이 우선 탐색

깊이 우선 탐색방법은 노드들이 생성된 순서에 의거하여 확장시키는 것으로 출발노드가 목표 노드가 아닌 것으로 가정한 것이다. 만일 출발 노드가 목표 노드인 경우도 있을 수 있다면 이에 대한 테스트를 첨부하면 된다. 탐색 과정에 의하여 생성된 노드와 포인터들은 암시적으로 정의된 전체상태공간 트리의 부분 트리를 형성하게 된다. 이러한 부분 트리를 탐색 트리라고 부른다.

만일 해가 존재한다면, 출발 노드에서 목표 노드까지의 최단경로길이, 즉 연산자의 적용 횟수를 최소로 하는 경로를 찾는 것을 보장한다. 만일 해가 존재하지 않는다면 유한 그래프의 경우에는 실패로 끝난다.

(2) 넓이 우선 탐색

넓이 우선 탐색방법은 가장 최근에 생성된 노드를 가장 먼저 확장시킨다. 따라서 탐색 트리 에서 현재 가장 깊은 노드는 바로 확장되기 위하여 선택된 노드인 것이다. 이 방식은 경우에 따라서 해가 없는 경로를 계속해서 따라갈 수 있으므로 필요에 따라 위의 노드로 되돌아가는 백트래킹 할 수 있는 방도를 마련해야 한다.

이를 위하여 통상적으로는 적당한 깊이제한을 주어 어떤 노드가 그 이상의 깊이 갖게 되면, 이 노드를 더 이상 확장시키지 않고 깊이제한을 넘지 않는 것 중에서 가장 깊은, 즉 가장 최근에 생성된 노드를 골라서 이것을 확장시킨다.

넓이 우선 탐색 알고리즘의 흐름도는 Fig. 1 이다.

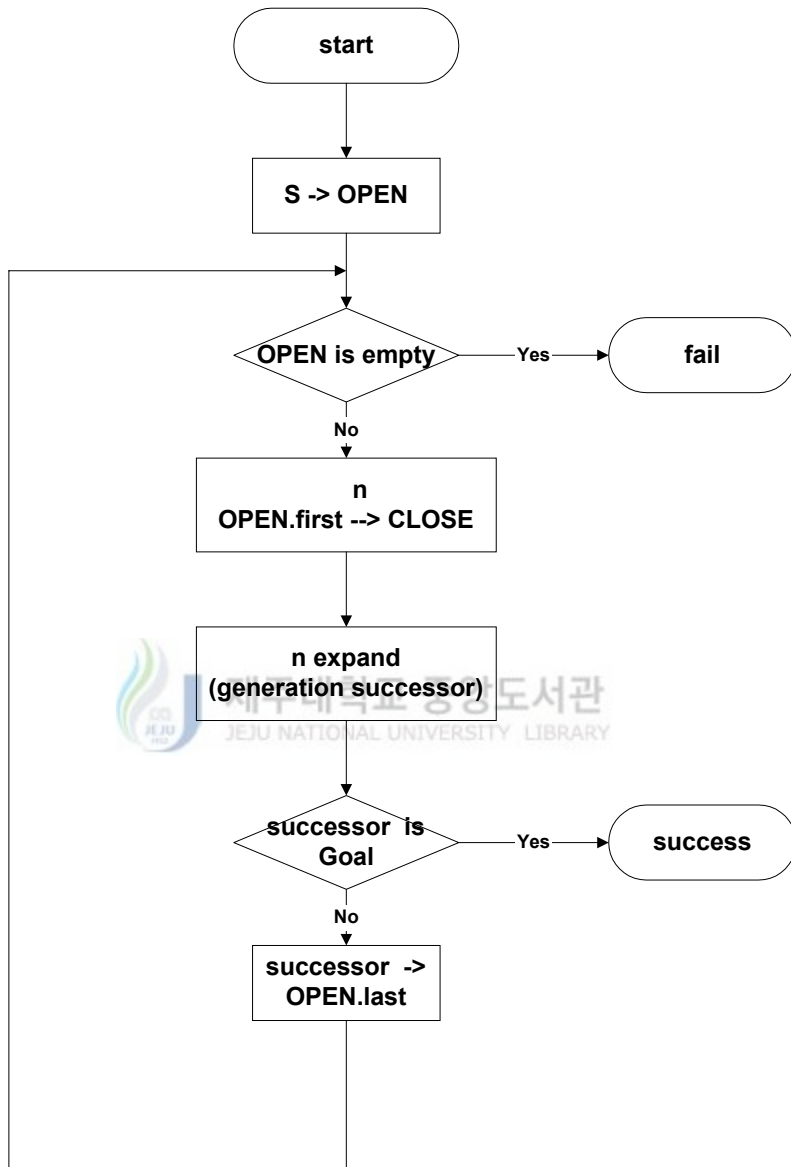


Fig. 1 Breadth-first search algorithm

2) 경험적 탐색

맹목적인 방법으로 모든 노드를 탐색하는 것은 목표에 근접하고 있는가를 알 수 없는 무모한 탐색방법이다. 목적지가 어디에 있는지 알지 못하고 무의미한 탐색을 방지하기 위해서는 탐색 경로를 안내할 수 있는 정보가 필요하다. 주어진 정보를 통하여 다음 단계에 탐색할 노드를 선택하기 위해서 평가 함수가 필요하다.

경험적 탐색은 탐색하고자 하는 노드들의 탐색 순서를 정하기 위하여 평가함수를 적용하여 현재상황에 가장 적합한 연산자를 선택하게 되는데, 고려 대상이 되는 연산자를 어떻게 한정하느냐에 따라 빔 탐색, 산 오름 탐색 및 최적우선탐색으로 나눌 수 있다(유,1988, 이와 김,1996).

(1) 빔 탐색

빔 탐색은 상태공간을 탐색 도중에 해가 될 가능성이 적은 노드를 잘라버리는 가지치기 방식의 하나로서 일반적으로 탐색 도중에 해가 될 후보가 몇몇 있는데, 항상 한정된 수의 후보만 남기고 그 이외의 것은 버리는 방식을 말한다. 이렇게 하면 탐색이 진행되더라도 후보의 수가 불어나지 않아 마치 빛의 빔이 크게 확산하지 않고 목표를 향해 나아가듯이 탐색을 진행시킬 수 있다.

(2) 산 오름 탐색

산 오름 탐색은 상태공간내에서 목표점에 가까워진다고 추정되는 방향으로 나아가는 탐색으로서 마치 산꼭대기에 오를 경우, 목표점은 위쪽에 있으므로, 위쪽으로 올라가는 데 대응한다. 그러나 실제 목표에 못 미쳐 있는 극대점에 도달해 버리는 경우도 있다. 그러므로 목표점의 높이를 미리 알고 있으면 극대점에서 다시 내려와서 다른 방향으로 나아가거나 다른 출발점부터 다시 시도할 수 있다.

(3) 최적 우선 탐색

최적 우선 탐색은 상태공간을 탐색하는 효율을 높이기 위하여 평가함수를 정의하여 어떤 기준을 설정하고, 또 탐색하지 않는 어떤 문제 중에서 최소(또는 최대)의 평가 값을 갖는 것을 선택하는 방법을 취하는 탐색방법이다.

경험적 지식에 사용되는 적절한 평가 함수를 노드에 적용하여, 지금까지 만들어진 노드 중 목표로 이끌 가능성이 가장 높은 노드를 택한다. 후계 노드를 만들기 위해 규칙을 이용하여 선택된 노드를 전개, 확장한다. 만약 이들 후계 노드 중에

원하는 목표 상태가 포함되어 있다면 탐색을 중단하고, 그렇지 않다면 새로 만들어진 후계 노드를 지금까지 만들어낸 노드로 구성된 집합에 첨가한다. 다시 목표로 이끌 가능성이 가장 높은 노드를 찾아 최적 우선 탐색을 계속 진행한다.

최적 우선 탐색 알고리즘의 흐름도는 Fig. 2 이다.



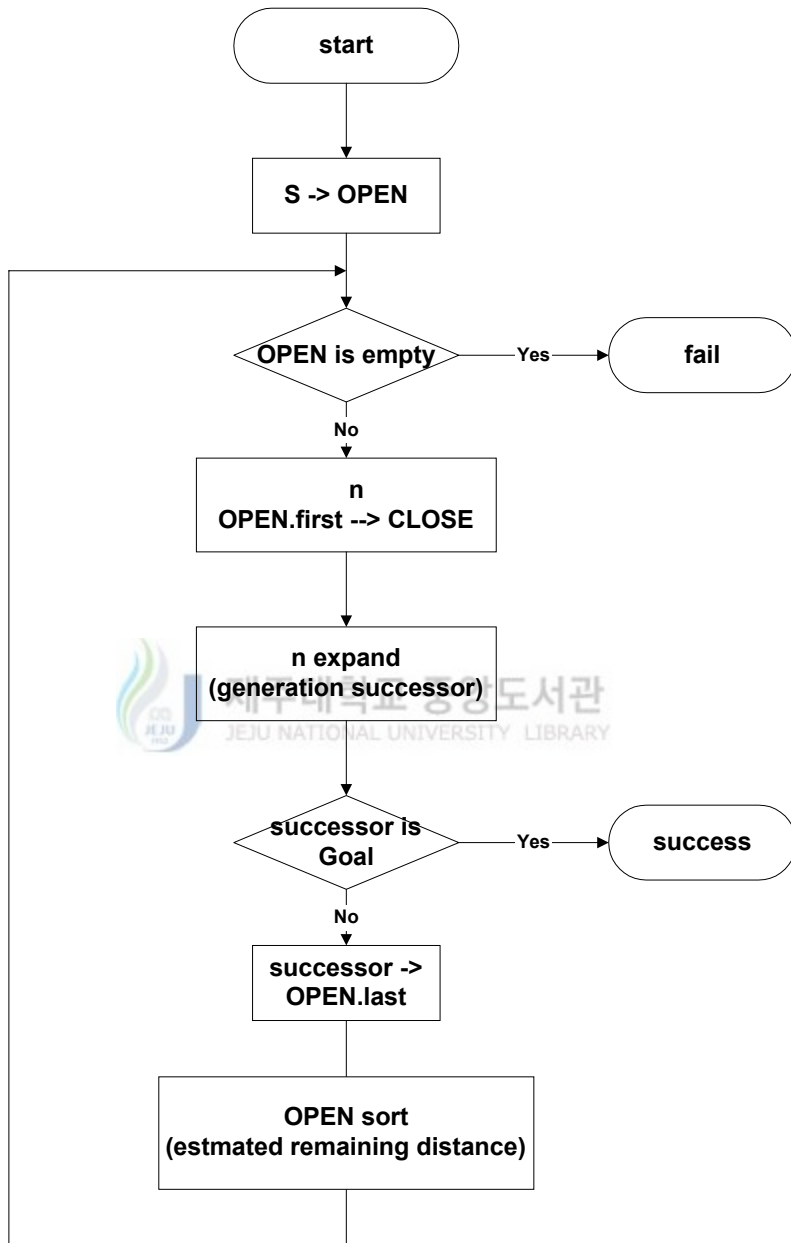


Fig. 2 Best-first search algorithm

2. 제약 조건

시간표와 관계된 학생·교수가 만족하는 시간표란 이들의 요구하는 다양한 요구조건들을 모두 수용한 상태의 시간표가 될 것이다. 그러나 이러한 제 조건들을 모두 수용한 시간표 작성은 거의 불가능하다. 또한 수업시간표 작성에는 이들 요구와는 별개의 대학의 특성에 따른 교과과정이나 강의실 등의 제약조건들이 주어진다. 그러므로 시간표 전산화에 따른 강의실 및 강의시간의 제약조건은 Table 1과 같다.

3. 자료 흐름

교과과정은 학과에서 편성된 교과과정을 기초로 최종적으로 교무과에서 확정시키면서 전산처리를 위해 신규과목에 대해 교과목번호를 부여한다. 확정된 교과과정은 요청에 의하여 수시로 변경할 수도 있다. 교과과정을 기초로 학과에서는 매학기 개설과목을 확정시키고 과목에 대한 담당교수를 선정한다.

그리고 합강여부, 연강여부, 과목의 특성에 따라서 부득이 강의시간과 강의실을 지정해야할 과목 등과 같은 자료를 확정시킨다. 담당교수는 교원인사DB에서 교수에 대한 정보를 받아오는데 신규교수 및 시간강사 위촉 등은 실제 학기초에 이루어지는데 시간표 작성은 임용 전에 이루어지기 때문에 임시번호를 부여하여 처리할 수밖에 없다.

이러한 모든 자료를 기초로 하여 시간표 작성이 이루어지며, 최종 시간표가 완성되면 시간표 검색 및 수정을 할 수 있으며, 교수별 강의시수 일람표, 강의실활용 현황, 학과별시간표, 강의실별 시간표, 교수별 시간표 등 수업에 필요한 자료를 출력한다. 이러한 자료 흐름과 시스템이 수행하는 기능들간의 상호작용, 변환 등의 자료흐름도를 Fig 3에서 보여주고 있다.

Table 1 Restriction for room and time

강의실 제약조건
<ul style="list-style-type: none"> • 한 강좌당 수용인원에 알맞는 강의실을 배정한다. • 수용인원에 알맞는 강의실이 부족할 경우에는 규모가 큰 강의실 중 수용인원에 근사한 강의실부터 배정한다. • 학생이 최소한 움직이도록 배정한다. • 연속강의 강좌는 같은 강의실을 배정한다. • 실험·실습과목과 같은 특수강의실은 지정된 강의실을 배정한다. • 일부 학과를 제외하고 학과별 전용강의실 배정은 무시한다.
강의시간 제약조건
<ul style="list-style-type: none"> • 특정요일이나 교시에 편중되지 않도록 균등히 배정한다(토요일 제외). • 전임교원은 주 3일 이상 강의를 배정하고, 강의가 없는 날이 연속 3일이 되도록 배정 할 수 없다. • 학문관심과목(교양선택), 전공과목은 2시간 연속강의로 배정한다. • 학과별, 담당교수별 점심시간을 보장. (3,4,5교시중 1시간이상배정제외). • 1, 2학년 학과별 시간편성시 교양선택과목의 비지정시간대와 중복되지 않아야 한다. • 한 강좌에 2명 이상의 교수가 담당하는 특수한 교과목은 담당교수를 같은 시간에 편성. • 교수별 대학원 시간과 학부 시간이 중복되어서는 안 된다. • 학과별 전공필수 교과목은 학년 구분 없이 필수교과목간 시간 중복을 지양. (해당 학과 학생들의 재수강 고려). • 교직과목 개설학과는 교직과목시간과 전공과목 시간이 중복되지 않게 시간을 배정. • 교수별 1일 4교시이상 연속강의 배정과 강의시간과 강의시간 사이 4교시 이상 비는 시간이 있어서는 안 된다.

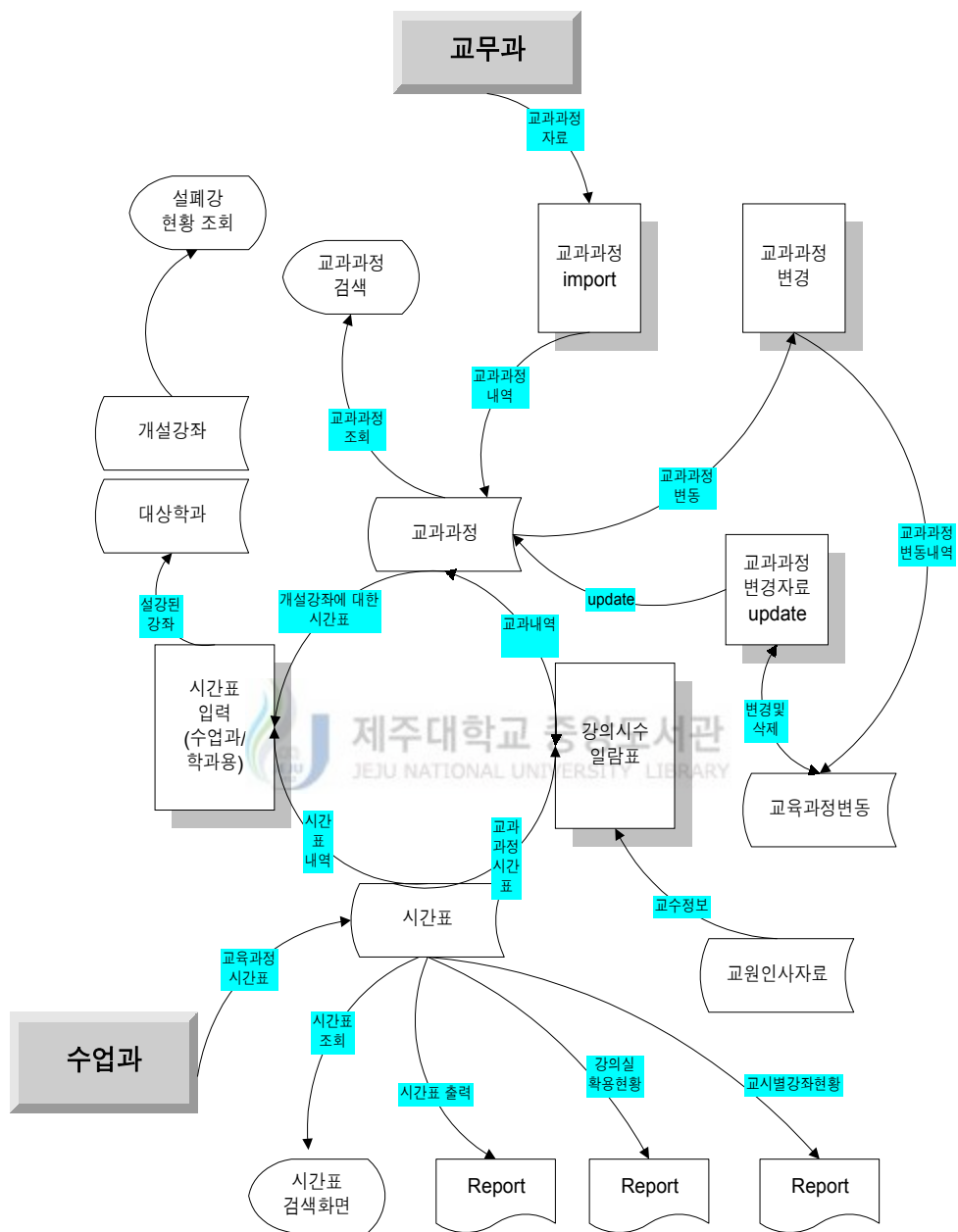


Fig. 3 Data flow diagram

III. 시스템 구성 및 기능

지금까지 탐색방법과 시간표 작성시 강의실과 강의시간에 대한 제약 조건 및 자료 흐름에 대하여 살펴보았다.

시간표 작성은 이와 같은 자료 흐름과 여러 가지 제약조건 때문에 강의실 활용을 최적화와 합리적이고 효율적인 시간표 작성에는 많은 어려움이 있다. 그러므로 시간표 작성은 이와 같은 조건하에서 학생·교수·강의실 3가지 구성요소가 유효 적절하게 배치되어야만 한다.

본 시스템은 이와 같은 어려움을 해결하기 위하여 인공지능의 기본 맹목적 탐색방법중 하나인 넓이 우선 탐색을 이용하여 탐색 경로를 안내할 수 있는 정보인 평가함수 지식 데이터베이스를 구축하였다.

구축된 지식 데이터베이스와 경험적탐색방법의 하나인 최적 우선 탐색에서 구축된 평가함수를 적절히 사용하여 학생·교수·강의실 3요소의 요구조건이 최대한 수용되어 효율적이고 최적화된 시간표를 만들어 내고 있다. 또한 수업시간의 편성과 관리의 편리성도 제공하고 있다.

본 장에서는 시간표자동화 시스템에서 설계된 이벤트 목록, 프로세스 모델들의 기능 및 계층화에 대해 살펴본다.

1. 구현 환경

시간표 자동화 시스템에 구현한 운영체제는 Unix(HP-UX B.10.20)이고 사용된 서버급 컴퓨터는 HP 9000/K250이며, Oracle 7.3.4 데이터베이스로 자료를 구축하였다.

NC기반의 프로그램을 사용한 컴퓨터는 펜티엄 166MHz 개인용 컴퓨터이며, 운영체제는 Microsoft Windows를 사용했고 오라클사의 Developer/2000 개발툴을 이

용하여 구현했다(Oracle,1996, 한국오라클,1997).

2. 이벤트 목록

이벤트 목록은 시스템을 둘러싸고 있는 모든 행동·사건을 의미하며 이벤트는 프로세스를 발생시킨다. 이벤트 목록은 엔티티가 존재 할 수 있는 여러 상태와 이러한 상태들 사이의 전이를 야기시키는 이벤트를 상세히 나타내며 시스템이 이벤트가 발생했음을 알게되는 자극과 시스템이 그 결과를 행하는 반응을 지정한다. 그리고 이벤트는 프로세스를 발생시킨다(한국산업기술원,1994).

우선적으로 현시스템이 사용자의 요구를 충족시키기 위해 어떤 일을 수행해야 하는지를 파악해야 한다. 이러한 자료수집 기법으로는 면담, 질의서 작성, 문서검토, 관찰 등이 있다.

본 시스템은 “수업시간 작성 지침“문서와 일부학과 시간표 작성자와의 면담을 통해서 우선 전산화가 가능한 영역과 불가능한 영역을 분석하였다. 전산화가 불가능한 영역으로는 예체능계학과의 실습과목(4교시이상 연장)이나 의학과 의 실습과목, 현장실습과목 등이 있는데 그러한 과목은 강의시간과 강의실을 미리 지정하는 것으로 하였다. 시간표관리 이벤트 목록은 Table 2와 같다.

Table 2. Event list

EVENT	PROCESS
Init_State1()	자동처리 불가능한 영역 초기화
Init_State2()	교양선택 및 교직시간대 선정
Department_Curriculum()	학과별 교과과정 입력 및 교육과정개설
CourseNum_Select()	개설강좌 및 담당교수 선정 및 개설강좌번호 부여
Classroom_List()	강의실정보 최종 점검 및 강의실 목록 작성
Professor_RateTime()	교수별 미지정 시간대 결정
TogetherContinue_Check()	합강여부 및 연강여부 결정
TogetherTime_Select()	합강과목에 대한 시간표 편성
ContinueTime_Select()	연강과목에 대한 시간표 편성
BookTime_Select()	잔여과목에 대한 시간표 편성
InitState_ErrorCheck()	제약조건에 위배 과목 확인
Double_ErrorCheck()	중복여부 확인
Classroom_Select()	강의실 배정
Time_Change()	특정과목의 시간표 변경
Classroom_Change()	특정과목의 강의실 변경

3. 프로세스 계층

프로세스 모델은 실제 업무처리를 전산화하기 위해서 시스템처리로 변환하는 것으로서 우선 프로세스를 업무처리 특성 및 단계에 따라 계층화하고 상호 의존성(선행-후행관계)을 검토한다.

프로세스의 계층화 작업은 크게 기능, 단위업무, 기초업무, 업무절차로 분류할 수가 있으며 시스템의 외부에서 발생하는 모든 행동들의 내역을 검토하여 실행할 수 있는 최소단위의 업무활동인 기본 프로세스들로 분해한다. 데이터의 흐름, 기능들간의 상호작용을 분석하여 프로세스 계층도를 작성한다(한국산업기술원,1994).

시간표자동화 기능을 살펴보면 크게 기초작업, 시간표작성, 강의실배정 등의 단

위업무로 분류 할 수가 있다. 각각의 단위업무에는 여러 개의 기초업무로 분류할 수가 있는데, 예를 들어 합강과목 시간표 작성 기초업무는 강의시간탐색, 강의시간표 중복 검사 등으로 세분화하여 업무처리를 하였다. 이러한 시간표자동화 시스템에 대한 프로세스의 계층도를 Fig 4에서 보여주고 있다.

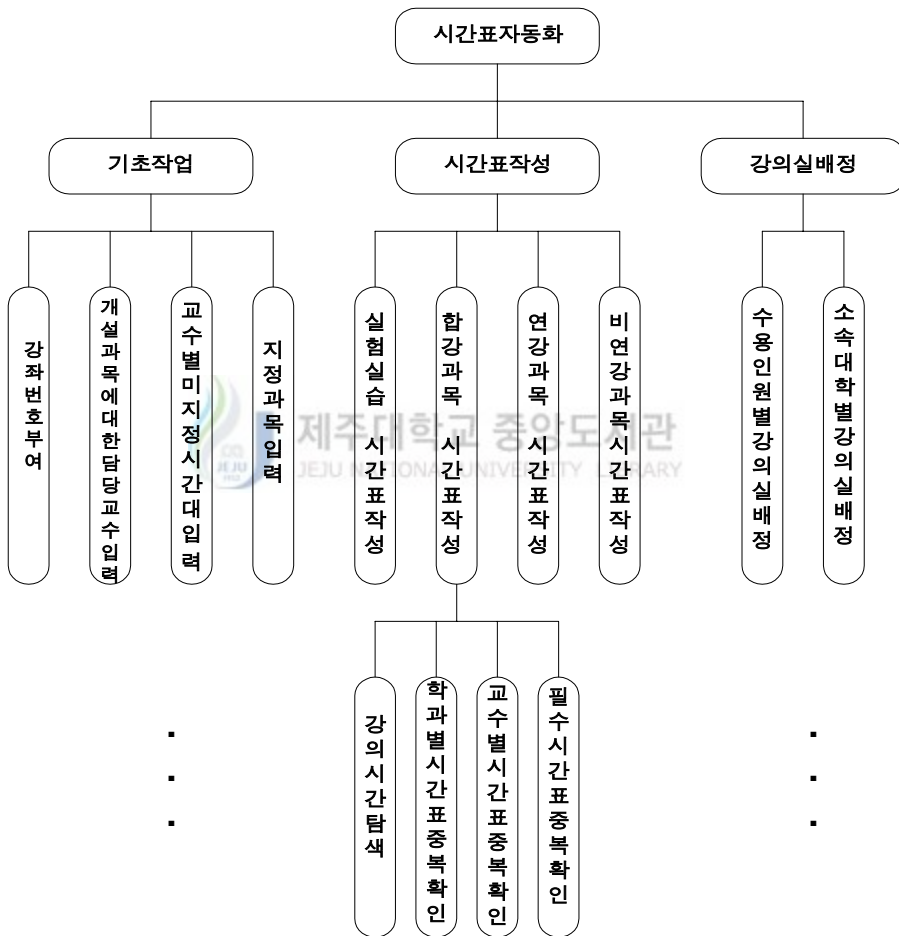


Fig. 4 Process layer diagram

4. 지식 데이터베이스 구축

평가함수 지식 데이터베이스는 맹목적 탐색 방법의 하나인 넓이우선탐색 시스템에 의해 구축되었다. 본 연구에서 사용된 평가함수는 크게 두 가지로 나누어 적용하였는데, 첫째는 학생 시간표에 적용되는 평가함수이다. 이는 학생들의 수업시간이 4시간 이상 연강수와 4시간 이상 공강수의 데이터로 이루어져 있으며, 둘째 교수 시간표에 적용되는 평가함수는 교수 수업시간이 4시간 이상 연강수와 4시간 이상 공강수의 데이터로 이루어져 있다. 이러한 평가함수에 나올 수 있는 경우의 수(시간표정렬순서, 과목정렬순서)는 무한히 많은데 전부 실험하기에는 너무 무의미한 것 같아 한 학과 개설과목을 대상으로 모의실험을 한 최적의 결과를 전체시스템에 적용하고자 한다.

Table 3은 한 학과를 대상으로 17개 과목 총 51시수, 7명의 교수 데이터를 가지고 맹목적 탐색방법에 의해서 구축된 평가함수 데이터의 예이다.

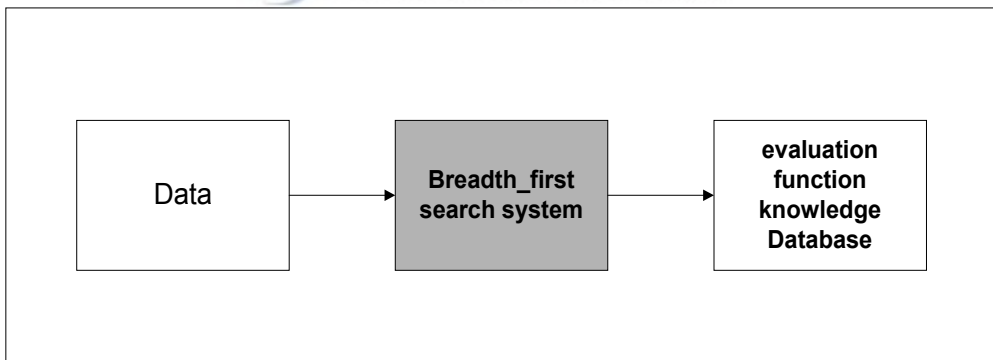



Fig. 5 Breadth-frist search system

Table 3 Evaluation function

과 목	시 간	평가함수
A,B,C... G	월1,월2, ...금9	학과별 4시간이상연강수: 5 학과별 4시간이상공강수: 0 교수별 4시간이상연강수: 1 교수별 4시간이상공강수: 1
B,C, ...G,A	월1,월2, ... 금9	학과별 4시간이상연강수: 7 학과별 4시간이상공강수: 0 교수별 4시간이상연강수: 2 교수별 4시간이상공강수: 0
:		
:		
E, ...A,G,F	수2,월6,화2	학과별 4시간이상연강수: 1 학과별 4시간이상공강수: 0 교수별 4시간이상연강수: 0 교수별 4시간이상공강수: 0
 제주대학교 중앙도서관 JEJU NATIONAL UNIVERSITY LIBRARY		
G,F,...B,A	금8, ...월1,금9	학과별 4시간이상연강수: 7 학과별 4시간이상공강수: 0 교수별 4시간이상연강수: 2 교수별 4시간이상공강수: 2
G,F, ...B,A	금9,금8, ... 월1	학과별 4시간이상연강수: 8 학과별 4시간이상공강수: 0 교수별 4시간이상연강수: 2 교수별 4시간이상공강수: 2

5. 최적해 탐색

시간표작성을 하는데 예는 여러 가지 제약조건들도 상호 충돌이 발생하게 되며, 최적화된 시간표는 이러한 제약조건 충돌문제뿐만 아니라 요구조건과 제약조건과의 충돌, 요구조건간(학생·교수) 상호 충돌되는 문제들을 적절히 조화 있게 수용

하여야 최적에 가까운 시간표를 작성할 수 있다.

이러한 수업시간표의 문제들이 최적 해를 얻기 위한 탐색은 맹목적 탐색을 기초로 경험적 탐색이 가미되는데 경험적 탐색은 연속강의시간 처리 혹은 필수과목의 학과별 중복을 피하는 처리 등을 위해서 우선적으로 탐색되며 그 나머지는 맹목적 탐색이 적용된다.

Table 4 Result of search

강좌번호	과목명	담당 교수	학과 학년	필수 여부	학점 시수	연강 여부	시간표
1121	국어음성학	교수1	A학과 2학년	필수	3학점 3시수	연강 아님	월4,화6,목4

전체 수업시간이 특정요일과 특정시간에 편중되지 않고, 월요일 1교시부터 금요일 9교시까지(45개의 시간)의 시간대를 균등히 분배하기 위해서는 『월1,월2,월3 . . .』 등의 순서로 탐색하는 것보다는 『화1 -> 수2 -> 수3 -> 월4 -> 목8 -> 금5』와 같이 평가함수의 최적결과에 의한 정렬순으로 탐색하는 게 좋다. Table 4는 구축된 평가함수를 이용한 탐색결과 예시로서 이를 이용하여 최적 시간표를 만들기 위한 탐색경로는 Fig 6에서 보여주고 있고 Fig 7은 시간표 탐색 프로그램에 대한 순서도이다.

A학과 2학년 시간표

요일 교시	월	화	수	목
1		1122	****	
2		1122		
3				
4				

교수1 시간표

요일 교시	월	화	수	목
1	1144			1163
2			1144	
3		0072		
4				



A학과(전학년) 필수과목 시간표

요일 교시	월	화	수	목
1		1122		1141
2		1122		
3		1111	1141	
4		1111	1141	

Fig. 6 Example of search

1) '화요일 1교시' 탐색

학과별 시간표에서 이미 강좌번호가 1122인 과목이 배정되어 있어 실패

2) '수요일 2교시' 탐색

학과별 시간표는 비어 있어서 성공, 그러나 담당교수(교수1) 시간표에 다른 과목이 있어 실패

3) '수요일 3교시' 탐색

학과별 시간표와 담당교수 시간표는 비어 있어서 성공, 그러나 필수과목 시간표에 같은 학과 3학년 필수 과목이 들어가 있어 실패

4) '월요일 4교시' 탐색

학과별 시간표, 담당교수시간표, 필수과목 시간표 모두 비어 있어 성공.

이와 같은 경로로 탐색하는 과정에서 학과별 시간표와 담당교수별 시간표의 점
심시간(3, 4, 5교시중 1교시 이상의 공강이 필요함) 검사가 동시에 이루어진다.



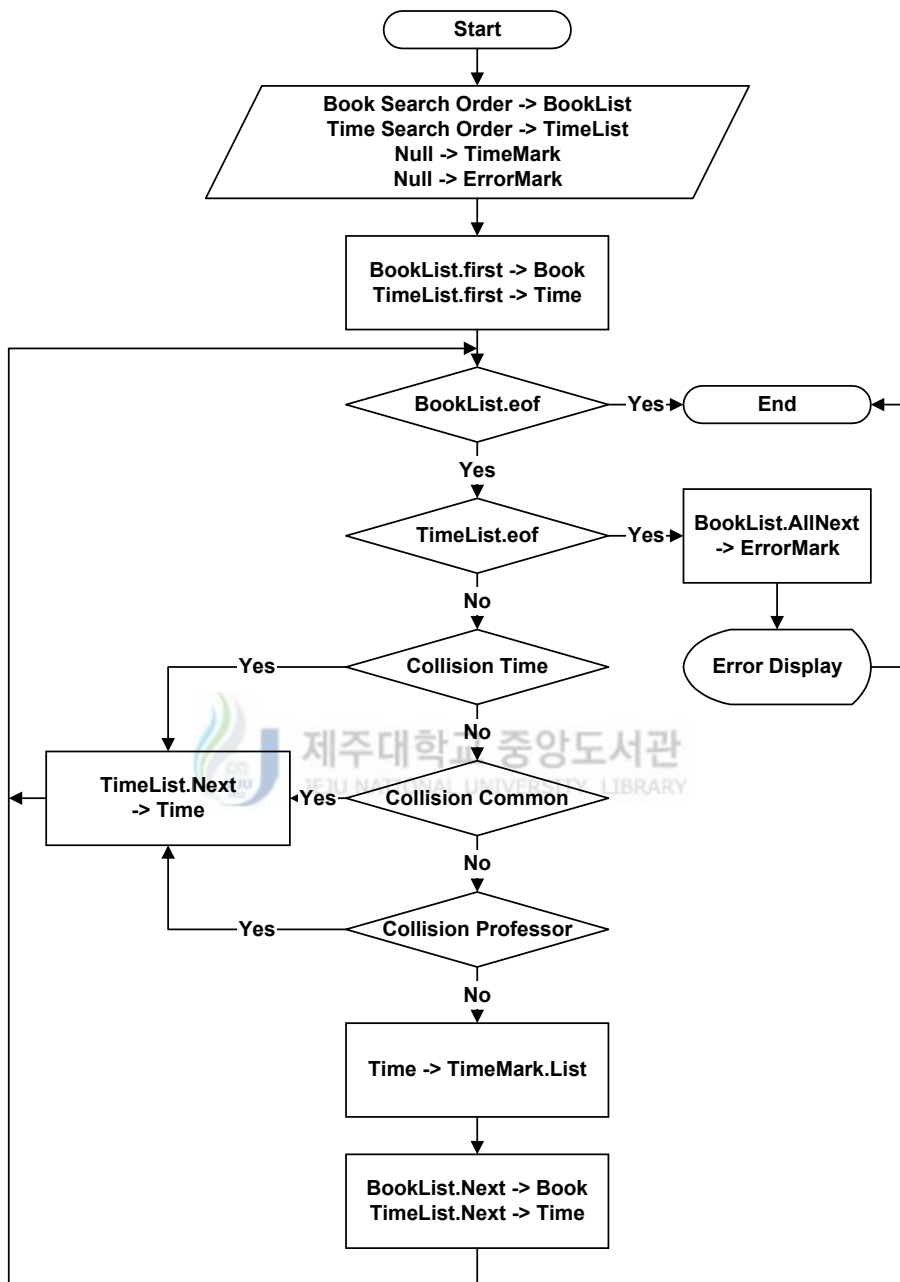


Fig. 7 Search program flowchart

6. 클라이언트 어플리케이션

시간표가 완성된 후에도 구성요소들의 사정에 의하여 부득이 변경해야 할 경우가 생긴다. 이런 한 변경을 위해 NC기반의 클라이언트 어플리케이션을 구축하였다 (David 1998). 완성된 시간표를 종합DB에 연계하면 사용자가 즉시 시간표와 강의실을 검색함으로써 변경하고 싶은 시간대와 강의실을 쉽게 찾을 수가 있다. 모든 검색화면은 Developer/2000 Tool을 사용하여 사용자에게 보다 편리하고 신속한 서비스를 제공하기 위해 그래픽 적인 어플리케이션과 GUI환경을 기반으로 검색시스템을 설계 및 구현을 하였다(조 등,1998, 정과 진,1998).

변경하려는 구성요소가 교수, 학생, 강의실 등 어느 것이든 비어있는 조건만 만족하면 변경이 가능하며 시간표 데이터를 체계적으로 관리할 수가 있으며 강의실 부족을 해결하는데 큰 도움이 된다.



IV 실험 및 고찰

1. 모의 실험

시간표 작성에 있어서 최적의 결과는 모든 교수들의 요구조건을 충족시키면서 주어진 강의실 한도 내에서 학생들의 수업효과를 극대화하는 것이라 할 수가 있다. 그러나 최적의 해에 접근한 시간표를 작성하기에는 제약조건이 너무 다양하고 이 해가 상반되는 경우가 많기 때문에 매우 어렵다.

최적의 시간표를 얻기 위한 검증 결과가 있어야 한다. 그러나 최종 평가는 수업 담당자의 경험적 지식을 바탕으로 막연히 이루어지기 때문에 구체적이지가 못하고 과학적인 분석이 어려우며, 교수나 학생들에 대한 만족도는 대상자가 너무 많아 평가하기가 힘들고 개인별 만족도의 차이로 정확한 평가가 어렵다.

본 연구에서는 본 시스템에서 구축된 평가함수에서 나온 탐색순서들에서 최적의 해를 찾아 시간표를 작성하여 문제를 해결하고자 한다.

1) 모의 데이터

모학과 전공과목을 대상으로 시간표를 작성을 한다. Table 5와 같이 이 학과의 교수는 8명이며, 2학년 전공 과목은 총 6명의 교수가 8과목에 26시간, 3학년 전공 과목은 총 5명의 교수가 7과목에 21시간, 4학년 전공과목은 총 2명의 교수가 6시간이다.

시간표 작성순서는 2학년, 3학년, 4학년 순으로 작성한다.

우선 교수별 만족도를 최대한으로 반영하기 위하여 교수 개인별로 배정되지 않아야 할 시간대를 적용하는데 있어서 최대한으로 6시간까지만 허용하였다. 본 모의실험에서는 Fig 8에서와 같이 교수별 미지정 시간은 A교수는 금1~금6까지, B교수는 금5~금9까지, C교수는 월1~월6까지, D교수는 화1~화5까지, E교수는 월1~월6까지, F교수는 수1~수6까지 및 G교수는 목1~목6까지로 미리 선정하였다.

이는 학과별 과목수 및 교수수에 따라 유동적으로 적용할 수가 있다. 그리고

이론 강의실은 2개만 배정하였다.

2학년 학생들은 교양선택과목을 이수할 수 있도록 교양선택 과목시간대를 적용 배정하였는데 Fig 9에서와 같이 2학년 교양선택 및 교직 과목시간은 월1, 월2, 화1, 화2, 수1, 수2, 수9, 목1, 목2 및 금1 이다. 또한 이 학과의 모든 교수는 전공과목 이외의 교양과목을 강의한다. A교수는 월6, 월7, 화3 및 목5, B교수는 월6, 목5, 금3 및 금4, C교수는 화4, 수3 및 수4, D교수는 월3, 화7, 화8 및 수6, E교수는 화3, 화4, 화6, 수3, 수4, 목3, 목4 및 금7, F교수는 월3, 월6, 수7, 수8, 목6, 목7, 금5 및 금6, G교수는 월3, 월4, 월6, 화6, 수5 및 목7이다.



Table 5 Sample data

반번호	교과목명	학점-이론-실습	담당교수	학년	비고
7021	전자기학및연습I	3-3-1	C교수	2	연강(2)
7022	물리실험II	1-0-2	C교수	2	실험
7023	수리물리학II	3-3-0	시간	2	연강(2)
7024	역학및연습II	3-3-1	D교수	2	//
7025	전산물리학및실습II	3-3-1	F교수	2	//
7027	통계열역학	3-3-0	A교수	2	//
7028	현대물리학및연습II	3-3-1	B교수	2	//
7029	물리실험II	1-0-2	C교수	2	실험
7041	물리실험IV	1-0-2	D교수	3	//
7042	물리실험IV	1-0-2	D교수	3	//
7043	전자기학II	3-3-0	G교수	3	연강(2)
7044	핵물리학I	3-3-0	시간	3	//
7045	고체물리학I	3-3-0	E교수	3	//
7046	광학II	3-3-0	G교수	3	//
7048	양자역학I	3-3-0	B교수	3	//
7064	반도체물리학II	3-3-0	A교수	4	//
7068	물리학세미나	3-3-0	시간	4	//
계	17과목	43학점 51시수	총8명		

	월	화	수	목	금	토
1교시	C교수 E교수	D교수	F교수	G교수	A교수	
2교시						
3교시						
4교시						
5교시						
6교시					B교수	
7교시						
8교시						
9교시						

Fig. 8 Initial state(professor)

	월	화	수	목	금	토
1교시						
2교시						
3교시	D교수 F교수 G교수	A교수 E교수	C교수 E교수	E교수	B교수	
4교시	G교수	C교수 E교수	C교수 E교수	E교수	B교수	
5교시			G교수	A교수 B교수	F교수	
6교시	A교수 B교수 F교수 G교수	E교수 G교수	D교수	F교수	F교수	
7교시	A교수	D교수	F교수	F교수 G교수	E교수	
8교시		D교수	F교수			
9교시						

Fig. 9 Initial state(common)

2) 적용 1

과목별 탐색순서는 Table 6이며, 시간별 탐색순서는 Fig 10 이다. 적용1의 탐색순서는 과목번호와 시간을 오름차순으로 정렬된 탐색순서를 이용하여 모의 실험을 하였다. 이때 사용된 평가함수는 학과별 4시간 이상 연속강의수만 5개이며 교수별 4시간이상 연강수와 공강수는 각각 1개이다.

적용1의 탐색순서로 본 논문에서 구현된 자동화 시스템에 적용한 결과는 Fig 11이다.

Table 6 Book search order for application 1

탐색순서	과목코드
1	7021
2	7022
3	7023
4	7024
5	7025
6	7027
7	7028
8	7029

	월	화	수	목	금	토
1교시	1	10	19	28	37	46
2교시	2	11	20	29	38	47
3교시	3	12	21	30	39	48
4교시	4	13	22	31	40	49
5교시	5	14	23	32	41	
6교시	6	15	24	33	42	
7교시	7	16	25	34	43	
8교시	8	17	26	35	44	
9교시	9	18	27	36	45	

Fig 10 Time search order for application 1 and 2

교시	월	화	수	목	금	토
1	****	****	****	****	역사통사(1) 07214	
2	****	****	****	****	통사(1) 07212	
3		역사통사(1) 07201	역사통사(1) 07204	조선통사(1) 07207	조선통사(1) 07212	
4	역사통사(1) 07204	조선통사(1) 07201	역사통사(1) 07204	조선통사(1) 07207	조선통사(1) 07212	
5	조선통사(1) 07201	조선통사(1) 07201	조선통사(1) 07201	조선통사(1) 07207	조선통사(1) 07212	
6	조선통사(1) 07201	조선통사(1) 07201	조선통사(1) 07201	조선통사(1) 07207	조선통사(1) 07212	
7	조선통사(1) 07201	조선통사(1) 07201	조선통사(1) 07201	조선통사(1) 07207	조선통사(1) 07212	
8	조선통사(1) 07201		조선통사(1) 07201	조선통사(1) 07207	조선통사(1) 07212	
9	조선통사(1) 07201		****	조선통사(1) 07207	조선통사(1) 07212	
10						
11						
12						
13						

Fig. 11 Department timetable for application 1

3) 적용 2

적용2의 탐색순서는 과목만 Table 7에 의한 순서를 이용하였고 시간별 탐색 순서는 적용1과 같은 순서를 이용하였다. 이때 사용된 평가함수는 학과별 4시간 이상 연속강의수만 4개이며 교수별 4시간이상 공강수는 1개이다.

적용2의 탐색순서로 본 논문에서 구현된 자동화 시스템에 적용한 결과는 Fig 12이다.

교시	월	화	수	목	금	토
1	****	****	****	****		
2	****	****	****	****	물리실험11 07022	
3	컴퓨터공학실험11 07023		컴퓨터공학실험11 07023		물리실험11 07022	
4		수업실용11 07025	컴퓨터공학실험11 07023	컴퓨터공학실험11 07023	컴퓨터공학실험11 07023	
5	컴퓨터공학실험11 07023	수업실용11 07025		컴퓨터공학실험11 07023	물리실험11 07022	
6		컴퓨터공학실험11 07023	컴퓨터공학실험11 07023	컴퓨터공학실험11 07023	물리실험11 07022	
7	물리실험11 07022	컴퓨터공학실험11 07023	컴퓨터공학실험11 07023	물리실험11 07022	물리실험11 07022	
8	물리실험11 07022		수업실용11 07025	물리실험11 07022	물리실험11 07022	
9	컴퓨터공학실험11 07023		****	컴퓨터공학실험11 07023		
10						
11						
12						
13						

Fig. 12 Department timetable for application 2

4) 적용 3

경험적 탐색방법에 의하여 평가함수 지식 데이터베이스를 이용하여 탐색한 결과 최적의 탐색경로를 얻었다. 과목별 탐색순서는 Table 7과 같으며, 시간별 탐색순서는 Fig 13 이다.

이때 사용된 평가함수는 학과별 4시간 이상 연속강의수만 1개이며 나머지 평가함수 값은 '0' 이었다. Fig 14는 7029 및 7021 과목을 적용3에 의한 정렬방식으로 탐색한 순서의 예이다.

적용3의 탐색순서로 구현된 결과는 Fig 15이다.

Table 7 Book search order for application 2 and 3

탐색순서	과목코드
1	7029
2	7021
3	7025
4	7028
5	7023
6	7024
7	7027
8	7022

	월	화	수	목	금	토
1교시	15	34	42	28	39	46
2교시	37	45	1	6	43	47
3교시	3	31	32	14	36	48
4교시	9	26	5	33	17	49
5교시	11	41	12	21	18	
6교시	44	7	22	24	20	
7교시	23	8	29	10	4	
8교시	19	2	38	40	30	
9교시	25	13	27	16	35	

Fig 13 Time search order for application 3

	월	화	수	목	금	토
1교시						
2교시			① 교양영역	⑥ 교양영역		
3교시	③ 교수					
4교시	⑨ 교수		⑤ 교양			
5교시						
6교시		⑦ 7021				
7교시		⑧ 연강안됨		⑩ 7021	④ 7021	
8교시		② 7029				
9교시						

Fig. 14 Example of searching for application 3



교시	일	화	수	목	금	토
1						
2					물리실험 II 67022	
3			전공필수역물 II ST024	전공필수역물 II ST024	물리실험 II 67022	
4		수업필수역 I ST025	전공필수역 I ST024	전공필수역물 II ST024	전공필수역물 II ST024	
5	전공필수역물 II 67024	수업필수역 I ST025				
6	전공필수역물 II 67024	전공필수역물 I ST021	전공필수역물 II ST024	전공필수역물 II ST024		
7			수업필수역 I ST025	전공필수역물 I ST021	전공필수역물 I 67021	
8	전공필수역물 II 67024	물리실험 I 67020	물리실험 I 67020	물리실험 I 67021	물리실험 I 67021	
9	전공필수역물 II 67024	물리실험 I 67020		물리실험 I 67021	물리실험 I 67024	
10						
11						
12						
13						

Fig. 15 Department timetable for application 3



5) 수작업 시간표

모의실험에서 적용한 학과의 조교와 교수들에 의해서 작성된 실제 시간표를 분석해 보았다. 2학년 전공시간표에서 월요일의 경우 2교시부터 8교시까지 연속으로 편성되어 있어 학생에는 최악의 시간표이다.

평가함수는 학과별 4교시 이상 연속강의수는 4개이고 공강강의수는 3개이다. 그리고 교수별 4교시 이상 연속강의수는 3개이고 공강강의수는 1개로써 수기로 작성된 시간표의 평가함수는 최적의 결과가 아니었다. 실제 수작업으로 생성된 시간표 결과는 Fig 16이다.

	월	화	수	목	금	토
1교시	****	****	7028	****	7021	
2교시	7022	****	7028	7028	7021	
3교시	7022		7024		7024	
4교시	7028	7024			7024	
5교시	7029					
6교시	7029			7023	7023	
7교시	7025	7021		7023	7025	
8교시	7025		7021	7027	7025	
9교시			****	7027	7027	

Fig. 16 Department timetable for person

2. 비교 분석



한 학과를 대상으로 모의실험을 실시한 시간표 중에서 최적의 결과인가를 가려 내는데는 어려움이 많다고 본다. 본 연구에서는 실제 수작업으로 만들어낸 시간표와 다른 평가함수에서 나온 시간표를 비교 분석하고자 한다. 많은 평가함수를 비교 분석하기는 어렵고 크게 2가지 과목정렬과 시간표정렬에 의한 평가함수를 적용하였다.

본 연구에서는 교수 개인별 제약조건을 일부 수용하였으므로 교수별 시간표는 비교분석하지 않고 학과별 시간표만 비교분석 하고자 한다. 적용1과 적용2의 시간표를 보면 4교시이상 연장되는 경우의 수가 많으며 개설강좌가 적은 3학년과 4학년 시간표는 시간표 정렬을 하지 않은 경우에는 특정요일과 교시에만 집중되는 경향이 많았다.

수기로 작성된 시간표는 교양선택 시간대에도 편성되었고 학생들의 점심시간도 전혀 고려를 하지 않았으며 4교시이상 연장되고 공강되는 경우도 발생하였다. 이는

수기로 시간표를 작성할 경우에는 교수 개인의 요구사항을 전부 수용하다 보면 이와 같이 학생들에게는 불합리한 시간표가 만들어질 수밖에 없다.

본 논문에서 최적의 시간표로 평가되는 적용3의 시간표는 교수 개인의 요구사항을 최대한으로 수용하고 학생들에게도 적절한 시간 편성을 할 수가 있다.

Table 8은 모의실험을 한 평가함수 비교표이다 .

Table 8 Comparison table of evaluation function

	과목정렬	시간표정렬	평가함수결과
적용1	정렬안함	정렬안함	학과별 4시간이상 연강수: 5 학과별 4시간이상 공강수: 0 교수별 4시간이상 연강수: 1 교수별 4시간이상 공강수: 1
적용2	정렬함	정렬안함	학과별 4시간이상 연강수: 4 학과별 4시간이상 공강수: 0 교수별 4시간이상 연강수: 0 교수별 4시간이상 공강수: 1
적용3	정렬함	정렬함	학과별 4시간이상 연강수: 1 학과별 4시간이상 공강수: 0 교수별 4시간이상 연강수: 0 교수별 4시간이상 공강수: 0
수작업	없음	없음	학과별 4시간이상 연강수: 4 학과별 4시간이상 공강수: 3 교수별 4시간이상 연강수: 3 교수별 4시간이상 공강수: 1

V 결 론

대학의 교육기능이 원활하게 수행되기 위해서는 수업시간표의 효율적인 편성운영이 그 중요도에서 비중을 크게 차지한다. 즉 수업의 구성 요소인 학생·교수들의 요구조건이 수용되지 않음으로서 수업을 듣는 자와 수업을 하는 자들이 불편으로 이어져 수업에 효율을 기할 수 없게 된다.

그러나 수업시간표 작성에 있어 학생들의 요구와 교수들의 요구가 서로 상충되고 이러한 상충되는 부분들을 적절한 시간대를 찾아 움직여 보면서 문제를 해결한다 하더라도 수업에 필요한 적절한 규모와 수업성격에 맞는 강의실이 있는지를 찾아야 하며, 적당한 강의실이 없을 경우는 다시 수업시간을 움직여 적절한 강의 시간과 강의실을 찾아 이를 해결하여야 한다.

따라서 본 연구에서는 이러한 문제들을 해결하기 위해 문제풀이 개념을 이용한 맹목적 탐색과 경험적탐색방법을 이용하여 최적화된 해를 얻는 과정을 거치므로서 최대한 합리적이고 효율적인 시간표를 작성하였다.

그러나 완벽에 가까운 알고리즘을 구현하기에는 어렵고 여러 가지 제약조건과 요구조건을 충분히 수렴하는 데는 어려움이 많아 조건 값에서 크게 벗어나지 않는 범위 내에서 시간표를 최적화 할 수밖에 없는 점도 있다. 특히 전공과목 대부분이 연속강의를 원하기 때문에 강의실배정이나 시간표 조정이 매우 어려워 시간표 작성에 일부 조건에 벗어날 수밖에 없었다.

이번 연구의 결과로는 수작업으로의 시간표작성에 따른 인력과 시간 소모를 크게 줄이는 효과가 있었으며, 수업시간표 작성과 관리 주체의 일원화와 체계적인 자료관리의 효율 증대, 강의실 활용을 제고, 학생들의 교육 이수기회 균등제공, 교수 시간표의 형평성 있는 분배와 학부제, 최소전공인정학점제 등 다양한 교육제도를 수용할 수 있게 되었다. 또한 이러한 수업시간표 전산화로 시간표 중복으로 인한 시간표변경이나 수강신청 변경이 없어지고 이에 따른 인력소모와 불편이 해소되는 기대효과도 가져 올 것이다.

그러나 앞으로 다양한 요구조건과 여러 가지 제약조건들을 모두 수렴한 시간표

작성은 불가능하므로 이러한 조건들을 가능한 최소화하는 방안과 병행하여 전산화에 의한 수업시간표 시행과정에서 발생하는 문제와 수업시간 운영과 관리의 보다 나은 편리성을 제공하기 위해 지속적인 프로그램 보완 개발이 필요하다.

그리고 이번 연구에서 나타난 전공교과목 연속강의에 따른 문제점과 기초자료에 개설교과목, 담당교수 선정, 실험실습시간 지정, 합강 등을 명확하게 표기하지 않음으로서 수업시간표 전산처리시 오류가 발생, 수업시간표 전산화 처리에 많은 장애를 가져오는 등의 문제점을 개선해 나가야 한다.



참 고 문 헌

박경하, 1997, “시간표평가 전문가시스템의 구현에 관한 연구” 공주대학교 대학원 석사학위논문

정운성, 1995, “시간표 작성 전문가 시스템 구현에 관한 연구” 한국 해양대학교 대학원 석사학위논문

한국산업기술원, 1994, “시스템분석 및 설계(전문가 양성과정)”

경희대학교 전자계산공학과 인공지능연구실, 1996, “경희대학교용 종합 강의 시간표 자동 작성 시스템 개발(연구개발 보고서)”

Kevin Knight, 1991, “Artificial Intelligence Second Edition”

G. F. Luger, and W.A.Stubble field, 1989 “Artificial Intelligence and The Design of Expert System”

이광형,김병래, 1996, “인공지능”, 한국방송통신대학교출판부

유석인, 1988, “인공지능원론” 교학사

조이기,김대영,김원중, 1998, “인트라넷 환경의 학교 정보시스템 구축방안” 한국정보과학회 호남.제주지부 학술발표논문집 Vol. 10, No. 1

한국오라클(주), 1998, “Oracle Magazine”, pp96-103

Oracle, 1996, "Programmer's Guide to the Oracle Call Interface"

한국오라클(주), 1997, "개발자를 위한 Oracle Developer/2000"

정대홍,전주예, 1998, "Oracle Web Application Sever 3" 도서출판 삼각형

David Lockman, 1998, "Oracle 8 Database Development in 21 days"

