Journal of the Korean
OR/MS Society
Vol. 15, No. 2, Dec., 1990

學位論文 審査스케쥴링에 관한 硏究***

梁光敏* · 申承澈**

A Thesis Committee Scheduling

Kwang Min Yang* and Seung-Chul Shin**

Abstract

The problem analyzed in this paper is to specify a schedule for thesis committee allowing maximum committee members' preference over thesis topics and meeting time-slots while satisfying other related scheduling requirements such as prohibiting simultaneous assignment of a committee member to more than one committee at a time.

Two mathematical programming approaches are presented to solve the thesis committee scheduling problem in a professional graduate school. They include LP-based branch-and-bound approach with linear subproblems with tighter added constraints and branch-and-bound approach with network subproblems. Characterization of the problem is analyzed to develop an efficient solution algorithm. Implementation and computational experiments are also performed for real size problems on an IBM PC/AT to show the relative performance of the proposed approaches along with an ordinary ILP solution approach.

---

* 中央大學校 經營大學 教授

** 現代經濟社會硏究院 硏究員

*** 1989年度 文教部 學術硏究助成費에 의한 自由公募課題로 選定되어 硏究되었음.

I. 序 論

1960년대 초부터 연구되기 시작한 여러가지 시간표작성(time-tabling)문제는 개설 강의과목의 결정, 강의실 배정, 시험시간표 작성 등에 관한 여러가지 접근방법을 제시하였다. 본 연구는 이와 같은 시간표 작성문제의 일종인 學位論文 審査스케쥴의 작성에 관한 연구이다. 우리나라의 경우 學位 取得資格要件에 論文提出이 필수적이므로 한정된 심사위원이 일정기간내에 논문심사를 마치도록 하는 효과적인 論文審査 스케쥴링방법이 긴요하다. 특히 심사대상 논문이 많은 專門大學院의 경우 논문 심사위원들의 選好度를 고려한 최적 스케쥴을 작성한다는 것은 일상적으로 해결해야 하는 문제이다.

본 연구는 論文審査時 심사위원들의 主題 및 審査時間에 대한 選好度를 기초로 심사위원들 전체의 選好度를 최대화하기 위한 심사스케쥴을 마련하는 것을 主題로 한다. 스케쥴 작성에서 고려해야 할 사항으로는 한 審査委員이 여러심사에 동시에 참여할 수 없다든가, 되도록 심사위원이 원하는 시간에 심사를 배정하며, 또한 심사가 특정 심사위원에게로 지나치게 편중되는 것을 止揚하는 것 등을 생각해 볼 수 있다. 그러나, 이러한 諸 與件을 모두 고려하다 보면 자연히 심사기간이 길어지게 되며, 주어진 기간내에서 임의적인 방법으로 이러한 조건들을 모두 만족시키면서 선호도를 최대화하는 것은 쉽지 않다. 따라서 본 연구는 주어진 條件을 만족시키면서 전체 審査委員들의 選好度를 最大化하는 審査時間 配定模型을 제시하고, 이에 대한 最適解를 도출하는 방법에 관해 연구하고자 한다.

초기의 시간표작성에 대한 연구는 강의 시간표와 강의실 배정, 시험시간표의 작성에 관한 實務的인 것들이 대부분이었다. 그 결과 특정 경우에는 매우 실질적인 도움을 줄 수 있었으나 解 接近方法上의 근본적인 변화없이 단순히 手作業으로 수행하던 것을 컴퓨터로 대체하기만 한 초기의 노력은 그리 만족스러운 성과를 가져오지 못했다.

이러한 가운데 Gotlieb[19]은 시간표 작성에 대한 새로운 접근방법을 제시하였고, Sherman[30]은 대학 강의시간표 작성에 대한 연구를 발표하였다. 1964년에는 Csima는 Gotlieb[13]와 함께 종전의 모형을 수정하여 보다 개선된 시간표 작성방법을 제시하였고, Cole[12]은 整列(sorting)기법을 사용하여 소용량의 컴퓨터에서 실행가능한 시간표 작성방법을 연구하였으며, Broder[11]는 學期末 試驗時間表 작성에 관한 것을 數理計劃模型으로 제시하였다.

1965년 Barraclough[8]는 英國의 대학들이 美國의 대학들과 비교하여 규모가 비교적 작고, 학생들이 특정 강의에 편중되는 현상이 거의 없는 특징을 고려하여 시간표작성에 적용할 수 있는 다양한 적용방법들에 대해 종합적인 연구를 하였고, 1966년 Almond[3]는 기존의 연구들에 비해 보다 범용성이 있고 소용량의 컴퓨터에서 실행가능한 대학 강의시간표 작성에 관한 간단한 휴리스틱을 제시하였다.

한편 Welsh와 Powell[33]은 彩色方法(coloring)을 이용하여 인접한 두개의 頂點은 같은 色

으로 칠할 수 없다는 조건하에 色彩의 數를 최소화하는 上限設定方法을 시간표 작성문제에 적용한 연구를 발표하였으며, Lions[26]는 헝가리(Hungarian)방법을 기초로 한 보다 효율적인 알고리즘을 이용하여 Ontario 교육대학의 스케쥴을 작성하였다. 1968년 Yule[39]은 Almond[3]가 제시한 휴리스틱보다 적용성이 뛰어난 휴리스틱을 개발하여 대학시간표를 작성하였고, Wood[36]는 기존의 시험시간표 작성프로그램들이 작은 크기의 문제에나 적용가능했던 것에 비해 6,000명의 학생과 1,000여 시험과목으로 구성된 시험시간표 작성프로그램을 개발했으며, Foxley와 Lockyer[17]는 Cole[12]의 優先順位賦與에 의한 방법을 이용하여 시험시간표 작성프로그램을 개발하였다. 또한, Wood[37]는 彩色方法을 이용하여 대규모의 시간표작성에 대한 방법을 제시했고, Lawrie[25]는 整數計劃模型을 이용한 시간표 작성방법을 제시하였으며, de Werra[34]는 彩色技法을 이용한 圖表理論模型(graph-theoretical model)을 제시하였다.

이러한 시간표 작성에 대한 연구는 1970년을 지나면서 침체된 경향을 보여 1971년 Andrew와 Collins[6]는 講義 配定의 制約과 각 講師에 대한 제약을 추가한 모형을 제시했는데, 이는 모형의 구조상의 특수성으로 인하여 整數條件이 없이도 정수해를 구할 수 있었다. 1976년 Aust[7]는 불가능한 시간표 작성문제를 3단계에 걸쳐 講義資源과 不可能性을 감소시키는 개량해법을 제시하였고, Tillett[31]는 Andrew와 Collins의 모형을 개선하여 강사의 과목에 대한 선호도는 강사에게 주어진 과목의 數만큼 다양할 수 있다는 가능성을 반영하여 각 강사의 差別的인 選好度를 포함할 수 있도록 모형을 확장하고, 각 강사가 담당할 수 있는 최대의 科目數에 관한 제약을 추가한 整數計劃模型을 제시했다. 1976년 Breslaw[10]는 이 문제를 선형계획모형으로 제시하였다.

1980년 이후는 시간표작성보다는 jobshop, flowshop 스케쥴링 등에 대한 연구가 많이 이루어졌는데, 시간표작성에 대한 최근의 논문에는 시험시간표를 작성하기 위한 Laporte와Desroches[23]의 논문이 있다. 이들은 기존의 시험시간표 작성기법이 실제에 적용하기에는 적용성이 부족했던 것에 비해 보다 적용성이 높은 해법을 제시하였고, 1985년에 Ferland와 Roy[15]는 각 자원에 활동을 할당하는 방법으로 대학의 강의시간 스케쥴과 강의실 할당문제를 해결하기 위하여 0-1 2次割當技法(quadratic assignment problem)을 이용한 해법을, 그리고 1986년에는 대규모 대학에서 발생되는 강의스케쥴에 대한 특수성들을 반영한 알고리즘이 Laporte와 Desroches[24]에 의해 제시되었는데, 이는 학생들의 시간표를 작성한 후, 이들을 균형화시키고, 다음으로 강의실 受容能力을 고려하는 段階別 접근으로 되어 있다.

국내의 연구로는 신영수[1, 2]의 대학강의시간표 작성에 관한 것이 시간표 작성에 관한 것으로는 유일한 연구이다.

위에서 살펴본 시간표 작성문제에 관한 해법은 크게 나누어 하나는 문제의 屬性(大規模性)에서 오는 組合的 特性과 실제 이용면에서 최적해 아

닌 可能解 만으로도 충분하다는 점을 고려한 특수 휴리스틱에 기초한 實用的 接近方法과, 數理計劃模型을 세우고 이에 대한 解(또는 可能解)를 구하는 알고리즘(부분적 휴리스틱)적 접근방법인 整數計劃, 非線型 네트 웍, 2次割當模型 등으로 구성되는 實用性보다 學問的 우아함에 치중하는 접근방법으로 나누어 볼 수 있다.

본 연구의 주제인 學位論文 審査스케쥴링에 관한 연구는 국내외 문헌에서 전혀 찾아볼 수 없다. 이는 지금까지 대부분의 경우 논문심사가 간헐적-非정규적으로 이루어져 왔으므로 현재 우리의 專門大學院의 경우에서와 같이 學位論文 審査件數가 短期間에 大量인 경우가 적었던데 기인한다. 본 연구의 주제와 같은 參席人員, 時間, 講義場所 등이 관련한 資源 制約下에서의 스케쥴링은 그 응용분야가 광범위하다고 하겠다. 歐美와 달리 자원이 제한되어 있는 우리의 경우 특히 그러하다고 판단된다.

본 연구의 구성은 위에서 기술한 연구 접근방법 중 後者에 속하는 것으로 문제의 定式化와 이에 따르는 여러가지 알고리즘적 解 接近方法에 있어 문제의 特性分析과 이에 대한 計算實驗을 比較·分析하는 것으로 이루어진다.

Ⅱ. 模　　型

본 연구에서 사용된 모형은 배정여부가 2進變數로 표현되는 二進整數計劃模型에 속하는 割當模型이다. 통상적 할당모형과 다른 점은 보통 할당모형의 첨자에 期間이 추가되어 있고 할당이 되는 경우라면 반드시 심사에 필요한 n명의 심사위원이 배정되어야 한다는 추가 제약이 있다.

한편, job-shop 스케쥴링과 비교하면 각각의 作業 所要時間이 불규칙할 수 있는데 비해 본 모형은 일정한 것이 특징이다. 일반적인 job-shop 스케쥴링의 可能解는 다음과 같은 조건이 만족되어야 한다. 즉, 두개의 작업이 동시에 하나의 기계에서 행해질 수 없으며, 각 job의 모든 작업은 중복됨이 없이 작업순서대로 이루어져야 한다는 것이다. 그러나 심사 스케쥴링의 경우는 작업이 중복되면 안 되는 것은 동일하나 작업순서의 제약이 없는 반면 할당되는 경우에는 필요한 n명의 심사위원들 모두가 동시에 두 학생에게 배정될 수 없다는 특징이 있다. 이러한 특징으로 인하여 모형의 구조상 제약식에 2進數 이외의 係數를 가짐으로해서 整數制約이 없이는 整數解를 얻기가 힘든 특징 즉, 整數갭(integrality gap)이 존재하는 문제이다.

본 모형의 目的函數는 각각의 심사위원의 審査時間 및 論文에 대한 選好度를 측정하여 이를 목적함수의 계수로 사용한다. 즉,

$$\text{Maximize} \sum_i \sum_j \sum_k p_{ij} q_{jk} x_{ijk}$$

i : 論文

j : 審査委員

k : 審査時間(time-slot)

따라서 각 심사위원의 주어진 기간내의 매 時間에 대한 可用與否(q_{jk})와 각 論文에 대한 選好度(p_{ij})를 設問 등의 형식을 통하여 측정한 후, 이들의 積을 目的函數의 係數로 사용한다. 이렇게

해서 측정된 값이 $p_{ij} \cdot q_{jk}$이며 논문, 심사위원 및 시간으로 구성된 변수가 x_{ijk}이다. 즉 $p_{ij} \cdot q_{jk}$란 i논문을 k시간에 심사하고자하는 j심사위원의 선호도이며, x_{ijk}변수는 0 또는 1을 가지며 1일 경우 i논문을 j심사위원이 k시간에 심사함을 나타내며 0은 그 반대의 경우이다. 표기의 편의상 다음부터 $p_{ij} \cdot q_{jk}$는 p_{ijk}로 표기한다.

본 연구에서는 다음의 조건들을 고려하도록 한다.

- o 심사위원은 동시에 두 논문을 심사할 수 없다.
- o 심사시간에는 심사에 필요한 n명의 심사위원이 모여야 한다.
- o 모든 논문은 심사기간중 반드시 한번의 심사를 받아야 한다.

다음에 위의 제약들을 하나씩 定式化하도록 한다.

(1) 심사위원은 동시에 두 논문을 심사할 수 없다.

어느 심사위원이고 특정시간에 동시에 두편 이상의 논문을 심사하는 것은 불가능하므로 이로부터 다음의 제약식을 쓸 수 있다.

$$\sum_{i} x_{ijk} \leq 1 \qquad \forall\ j,\ k$$

(2) 심사시간에는 심사위원 全員이 배정되어야 한다.

논문이 심사받게되는 시간에는 n명의 심사위원 전원이 참석하여야 한다. 즉, 각 시간마다 한 논문의 삼사에 참가하는 심사위원수는 0 또는 n이 되어야 하는데, 이는 0-1변수를 추가하여 다음과 같이 定式化 할 수 있다.

$$\sum_{j} x_{ijk} - n \cdot y_{ik} = 0 \qquad \forall\ i,\ k$$

(3) 모든 논문은 심사기간중 반드시 한번의 심사를 받아야 한다.

어느 논문이건 주어진 심사기간중에 반드시 한번의 심사를 받아야 하는데, 이는 각 논문마다 다음과 같이 표현된다.

$$\sum_{k} y_{ik} = 1 \qquad \forall\ i$$

시간배정을 나타내는 변수는 두가지 값만을 가질 수 있으므로 二進整數로 표현한다. 즉,

$$x_{ijk},\ y_{ik} = 0 \text{ 또는 } 1 \quad \forall\ i, j, k$$

위의 목적함수와 제약식들을 정리한 模型(P)은 다음과 같다.

$$
\begin{array}{llll}
\text{(P)} & \text{Maximize} \ \sum_{i}\sum_{j}\sum_{k} p_{ijk} \cdot x_{ijk} & & \\
\text{s.t.} & \sum_{i} x_{ijk} \leq 1 & \forall\ j, k & (1) \\
 & \sum_{j} x_{ijk} - n\ y_{ik} = 0 & \forall\ i, k & (2) \\
 & \sum_{k} y_{ik} = 1 & \forall\ i & (3) \\
 & x_{ijk} = 0 \text{ or } 1 & \forall\ i, j, k & (4) \\
 & y_{ik} = 0 \text{ or } 1 & \forall\ i, k & (5)
\end{array}
$$

여기서 n은 심사에 필요한 심사위원의 수이다.

Ⅲ. 解接近方法

序論에서 기술한 바와 같이 그 간의 시간표작성 연구의 접근방법은 임의적인 휴리스틱 방법과 알고리즘적 방법으로 대별된다. 이중 後者의 방법은 문제의 屬性(NP-complete)上 그 대상이 소규모의 假想的 문제에 국한되었다. 본 연구에서는 이를 활용하거나 극복하기 위한 두가지의 修正模型을 제시하기로 한다.

Ⅲ.1 問題의 特性

가) 問題 行列의 크기

모형에서 보듯이 I편의 심사대상논문, J명의 심사위원, K개의 심사시간의 경우 필요한 정수변수의 수는(I·K+I·J·K)로, 가령 논문, 심사위원, 시간이 모두 두배가 될 경우 필요한 변수의 수는 8배 이상으로 급격히 증가함을 알 수 있다.

나) 低密度 問題行列

대부분의 스케쥴링 문제에서 나타나는 바와 같이 이 문제 역시 모형에서 보는 바와 같이 대단히 성근(sparse) 문제행렬을 갖고 있으며, 그 係數도 0, 1, -n으로만 구성되어 있다. 따라서 문제를 표현하기 위한 특별한 자료구조를 고려할 수 있다. I편의 논문, J명의 심사위원, K개의 심사시간인 경우 非零密度는 2/(J·K+I·K+I)로 표시되므로 문제가 커짐에 따라 더욱 성글어짐을 알 수 있다.

다) 退化 問題

모형에서 알 수 있듯이 제약식의 대부분의 계수가 0, 1, 특히 제약식의 右項이 0이거나 아니면 모두 동일한 1로 구성되어 있으므로 통상적 심플렉스 방법을 적용할 경우 계산반복 과정의 많은 부분이 목적함수의 개선이 없는 退化基底의 연속일 것이 예상되므로 이에 대한 효과적인 대처방안이 이 문제를 효율적으로 풀 수 있는 관건이 된다는 것을 알 수 있다.

라) 小規模 커널 크기

최적해를 구했을 경우 원래 문제의 변수로서 값을 갖는 변수(=1)의 수는 (n+1)·I개 뿐이다. 즉, 통상적 심플렉스 방법의 基底의 많은 부분이 원래 변수 이외의 변수(여유변수 등)만으로 이루어 진다. 다시 말해 통상적 基底의 크기(J·K+I·K+I ; 원문제의 제약식 수)에 비해 束縛제약식(binding constraints)만으로 만들어지는 커널(kernel) [21]은 훨씬 작다.

마) 整數 갭

整數 조건을 弛緩한 문제의 解(대응되는 선형문제의 解)와 원문제의 최적해의 목적함수의 값과의 차이(integrality gap)가 매우 크다. 이는 통상적 방법에서의 分枝트리가 매우 커질 수 있음을 시사하는 동시에 효율적인 解를 위해서는 有效切面(cut) 추가에 의한 보다 조여진 定式化(tight formulation)가 효과적일 수 있음을 보여주는 것이다.

Ⅲ.2 修正 模型

위에서 살펴 본 문제의 특성에 비추어 解를 구하기 위한 두가지 접근방법을 제시하도록 한다. 그 하나는 문제행렬의 성김과 大規模 退化로 인해 통상적 심플렉스 방법에 의한 해법이 非효율적이라는 것에 촛점을 둔 네트웍 모형에 기초한 접근방법이고 다른 하나는 整數갭이 커서 탐색 트리의 크기가 방대해져 최적해의 탐색이 비효율적인 것을 개선하기 위해 보다 더 조여진 修正模型을 구축하여 分枝 탐색하는 방법의 두가지 접근을 한다.

가) 線型計劃 중심 接近方法

이 문제에 관한 선형계획 중심의 전통적인 탐색

방법은 탐색해야 할 整數변수의 수가 매우 많고 整數갭이 매우 크므로 해서 分枝트리의 크기가 커져 매우 비효율적이다. 따라서 정수 변수의 축소와 有效切面의 도입으로 보다 조여진(tight) 모형을 작성한 후 分枝 탐색하는 접근방법을 취한다. 이 방법은 또한 최적해에서의 커널크기가 문제행렬의 크기에 비해 작은 특성을 이용하여 전통적 基底가 아닌 커널만을 유지·갱신하는 방법을 이용할 수 있는 장점이 있다.

원 모형(P)와 아래에 제시한 모형(P_R)이 동일한 解를 가짐을 보인다.

(P_R) Maximize $\sum_i \sum_j \sum_k p_{ijk} \cdot x_{ijk}$

subject to (1), (2), (3) 및 (5)

$x_{ijk} \le y_{ik} \quad \forall\ i, j, k$

$\Sigma y_{ik} \le \lceil |J|/n \rceil \quad \forall\ k$

$x_{ijk} \ge 0 \quad \forall\ i, j, k$

여기서 $|J|$는 첨자 j가 속해 있는 집합 J의 크기를 나타낸다.

(정리) 가능해 집합을 F라고 하면, $F(P) = F(P_R)$

(증명) 〈필요조건〉 $F(P) \supseteq F(P_R)$이므로 自明.

〈충분조건〉

1) $x_{ijk} \ge 0$ 부분

$y_{ik} = 0$ 또는 1이면 (2)식은 $\sum_j x_{ijk} = 0$ 또는 n이 되므로 (1)과 (2)로 구성되는 제약식은 bipartite(식(3)은 disjoint). 따라서 (1)과(2)로 구성되는 제약식은 totally unimodular이고 右項이 모두 정수이므로 x_{ijk}는 정수값만을 가지고, 식(1)에 의해 Σx_{ijk}가 上限값을 가지므로 $x_{ijk} > 0$과 더불어 x_{ijk}는 0, 1값만을 가진다.

2) $x_{ijk} \le y_{ik} \quad \forall\ i, j, k$ 부분

(2)式과 (4)式에 의해

$y_{ik} = 0 \Rightarrow x_{ijk} = 0$

$y_{ik} = 1 \Rightarrow x_{ijk} = 0$ 또는 1

따라서 $x_{ijk} \le y_{ik} \ \forall\ i, j, k$

3) $\Sigma y_{ik} \le \lceil |J|/n \rceil \ \forall\ k$ 부분

(2)식을 i에 관해 합하면,

$\Sigma\Sigma x_{ijk} = n\, \Sigma y_{ik} \quad \forall\ k$

또는 $\Sigma y_{ik} = (\Sigma\Sigma x_{ijk})/n$

(1)식에서 $\Sigma x_{ijk} \le 1 \quad \forall\ j, k$이므로

$\Sigma y_{ik} \le |J|/n$

y_{ik}는 정수이므로

$\Sigma y_{ik} \le \lceil |J|/n \rceil$ Q. E. D.

나) 네트웍 중심 接近方法

Ⅱ장의 모형 (P)는 (1)식과 (2), (3)식의 두 부분으로 2분(bipartite)할 수 있다. 따라서 다음과 같이 변수치환에 의해 모형 (P)를 (P_N)으로 변환한다.

$\Sigma y_{ik} = 1 \ \forall\ i \Rightarrow \Sigma - n \cdot y_{ik} = -n \ \forall\ i$

$\Sigma x_{ijk} - n \cdot y_{ik} = 0 \ \forall\ i, k$

$\Rightarrow \Sigma - x_{ijk} + n \cdot y_{ik} = 0 \ \forall\ i, k$

$n \cdot y_{ik} = y'_{ik}$로 놓으면 (P)는 다음과 같이 쓸 수 있다.

(P_N) Maximize $\sum_i \sum_j \sum_k p_{ijk} \cdot x_{ijk}$

s. t. $\sum_i x_{ijk} \le 1 \quad \forall\ j, k$

$\sum_j - y'_{ik} = -n \quad \forall\ i$

$\sum_k - x_{ijk} + y'_{ik} = 0 \quad \forall\ i, k$

$x_{ijk}=0$ 또는 1

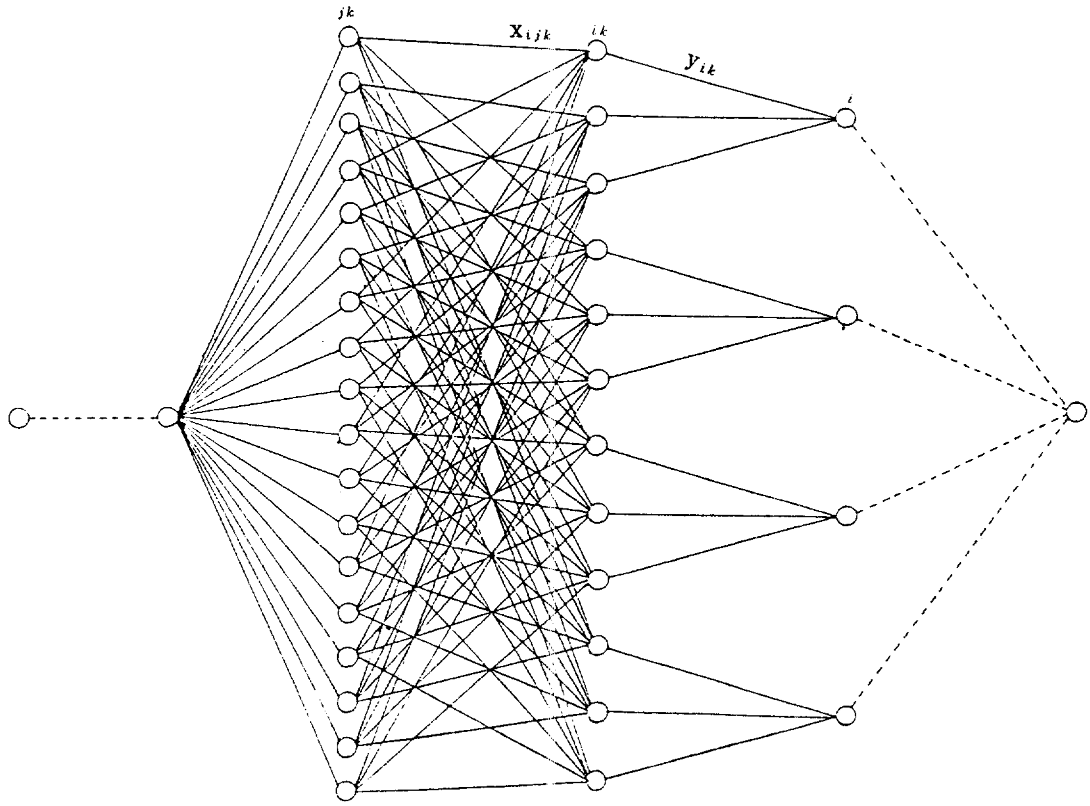
$y'_{ik}=0$ 또는 n

$|I|=4$, $|J|=6$, $|K|=3$의 경우를 圖示하면 〈그림-1〉과 같다.

마지막 제약식 ($y'_{ik}=0$ 또는 n)만 없다면 純粹 네트웍 모형이 되므로 y'_{ik}변수가 0 또는 n이 되도록 y'_{ik}변수에 관해 分枝탐색하는 접근방법이다. 이 네트웍 중심의 탐색방법에서는 통상적 가지 자르기(fathoming)로서 ① 整數解(0 또는 n), ② 不可能解 및 ③ 上限값에 의한 방법 이외에 추가적으로 다음과 같은 간편한 가지 자르기 방법을 추가할 수 있는 장점이 있고, 작은 할당문제를 풀므로 해서 (P)의 비교적 좋은 初期 가능해를 얻을 수 있는 장점이 있다.

모형 (P)의 초기 가능해를 간단히 얻을 수 있는 방안이 있다면 이는 제안한 두 가지 解접근 방법이 모두 分枝限界法에 의하므로 가지 자르기에 유용하게 쓰일 수 있다. 물론 최적해에 가까운 값일수록 효과가 있을 것이다.

다음의 할당문제(P_A)를 고려해 보자.



〈그림-1〉 4-6-3 네트웍 모형

(P_A) Maximize $\sum_i \sum_k p^{\circ}_{ik} \cdot y_{ik}$

s. t. $\sum_k y_{ik} = 1 \quad \forall\, i$

$\sum_i y_{ik} \leq \lceil |J|/n \rceil \quad \forall\, k$

$y_{ik} = 0, 1$

여기서 $p^{\circ}_{ik} = \sum_{i=1} p'_{ijk} \quad \forall\, i, k$

(p'_{ijk}는 p_{ijk}를 j에 관해 내림차順(descending order)으로 정렬한 것임)

(정리) F를 가능해의 집합이라고 하면

$y^{\circ}_{ik}\ \varepsilon\ F(P_A) \iff y^{\circ}_{ik}\ \varepsilon\ F(P)$

(증명) 〈충분조건〉

(P)의 식 (2)가 등식이므로 i에 관해 합하면 (P_A)의 두번째 제약식이 된다.

$$\sum_i y_{ik} = (\sum_i \sum_j x_{ijk})/n = (\sum_j \sum_i x_{ijk})/n \leq |J|/n$$

〈필요조건〉

(P)의 (3)式은 (P_A)에 있으므로 자연히 만족하고 (2)式중 $y_{ik}=0$일 경우는 $x_{ijk}=0$이므로 (1)식을 만족하며, $y_{ik}=1$일 경우는 n개의 x_{ijk}가 1의 값을 갖는다. 이때 (P)의 (1)식이 만족하지 않는 경우는 $\Sigma\Sigma x_{ijk}$가 $n \cdot \Sigma y_{ik}$를 감당하지 못하는 경우이다. 이 경우는 (P_A)의 두번째 제약식에 의해 방지된다.

Q. E. D.

위의 (P_A)의 최적해 (y^*_{ik})를 구해 이값을 (P) 또는 (P_r)에 제약식으로 추가하여 최적해를 구한 후 이를 (P) 또는 (P_r)의 초기 가능해로 사용 分枝탐색을 계속한다.

네트웍 중심 分枝탐색 접근방법의 경우 i°노드에서 $0 < y_{i^{\circ}k} < n$이라고 하면 y_{ik}를 0이나 n으로 固定시켜야 한다. 이때 변수 고정후 후보문제가 가질 수 있는 목적함수의 上限을 다음과 같이 간단히 계산하여 지금까지의 가장 좋은 整數解(incumbent)의 목적함수($=v^{\circ}$)와 비교하여 열등할 경우 후보문제를 풀지 않고 가지자르기를 행할 수 있다.

지금 문제의 목적함수의 값을 v라고 하고 $y_{i^{\circ}k}*$를 固定을 고려하는 변수라고 하고 K를 지금까지 고정 안 된(自由) 변수의 집합을 나타내고, $v_{i^{\circ}}$이 다음과 같이 정의된다고 하자.

$$v_{i^{\circ}} = \Sigma\Sigma p_{i^{\circ}jk} \cdot x_{i^{\circ}jk}$$

(방법-1) $v - v_{i^{\circ}} + p^{\circ}_{i^{\circ}k}* \leq v^{\circ}$ 이면

$y_{i^{\circ}k}* = n$은 fathom,

(방법-2) $v - v_{i^{\circ}} + \max\ p^{\circ}_{i^{\circ}k',k}* \leq v^{\circ}$이면

$y_{i^{\circ}k}* = 0$은 fathom.

Ⅳ. 具現 및 計算實驗

Ⅳ. 1 具現 및 計算實驗 結果

제Ⅲ장에서 기술한 解접근 방법의 성과를 평가하기 위해서 컴퓨터상에 구현하는데 있어 선형계획에 해당하는 부분은 商用 패키지인 LINDO[29]를 채택·이용하였다. Graves의 알고리즘은 前章의 문제특성에서 언급하였듯이 퇴화문제와 커널 표현을 동시에 해결해 줄 수 있기 때문에 기대되었으나 본 연구에서는 사용할 수 없어 제외되었다. 네트웍에 기초한 방법에서의 네트웍 코드는 Bradley[9]의 방법을 이용하고 分枝탐색 부분에서는 탐색트리 이외에 변수 固定어레이를 추가로 사용하였다.

문제발생기는 Borland社의 Turbo Basic으로 짜여져 있으며, 난수발생에 의해 목적함수의 계수를 0, ±4, ±8로 발생 사용하였다.

계산실험에 사용된 컴퓨터 시스템은 IBM PC/AT 호환기종(12MHz)으로 LINDO를 제외한 모든 코드는 DOS 3.1상에서 MS FORTRAN V.5.0으로 컴파일하여 사용하였다.

아래의 〈표-1〉에 세가지 접근방법에 의한 계산결과를 차례로 보인다.

Ⅳ.2 計算實驗 結果分析

〈表-1〉에서 알 수 있듯이 세 가지 접근방법(원래의 모형(P), 切面이 추가된 모형(P_R) 및 네트웍을 기초로 한 모형(P_N))에 의한 解 계산속도는 (P_N)이 예외없이 타 방법에 비해 월등히 빨랐으며, (P_R)도 (P)와 비교하여 작은 문제의 경우를 제외하고 빨랐다. 이러한 현상은 문제크기가 커짐에 따라 더욱 두드러 졌다.

모형(P)와 모형(P_N)은 동일한 트리탐색 방법과 가지자르기를 적용한다면 각 탐색트리 노드에서 풀어야 할 후보문제와 數(Brn으로 표시)에는 차이가 없어야 한다. (P_N)에서 특별히 고안한 트리탐색 방법은 적용하지 않았으므로 계산속도의 주된 차이는 후보문제를 푸는 속도에서 오는 차이라고 보아야 한다. 피봇 數의 차이는 후보문제를 풀 때의 進入변수 선정(pricing)전략이 다른데서 오는 결과이다.

(P)의 경우 整數갭이 매우 컸으나 (P_R)의 경우는 갭이 매우 줄어져 정수조건을 이완한 선형문제의 解가 바로 정수해가 되는 경우도 많았으나 추가 제약식으로 인하여 問題行列의 크기가 커져 풀어야 할 후보문제의 數는 매우 줄었으나 후보문제당 解계산 소요시간이 매우 길어졌다. 그러나 전반적으로 문제의 크기가 커짐에 따라 (P_R)이 (P)에 비해 時間 効率的임을 알 수 있다.

(P_N)모형에 의한 접근방법이 타 방법에 비해 시간 효율적인 이유는 후보문제가 네트웍으로 표현되므로 基底변환이 효율적으로 이루어 질 수 있었음에 기인한다. 반면에 정수갭을 줄여 보다 더 좋은 후보문제의 上限값을 얻어 가지자르기를 보다 빨리 할 수 있는 制約式의 추가는 일반적으로 네트웍을 구성하지 않으므로 이를 적용하기 곤란한 것이 이 방법의 단점이다.

(P_N)의 Brn*난은 최적해가 얻어진 후보문제의 순서 위치를 나타낸다. 최적해와 목적함수가 같은 整數解가 얻어진 이후의 계산은 더 좋은 解가 존재하지 않는다는 것을 확인하는 과정에 불과하며, 표에서 볼 수 있듯이 이 이후의 계산시간이 많은 부분을 점한다는 사실에 비추어 실제문제를 푸는데 있어서는 정수해를 一定數 이상 얻었을 경우 중단하는 방법을 고려해 보는 것(suboptimal)도 실용성 있는 방법임을 시사해 준다.

分枝탐색법의 효율성은 가지자르기의 효율성에 의하는 바 (P_N)모형에 의한 계산실험의 경우 3가지 방법(不可能解, 整數解, 目的函數 값의 上限) 중 거의 모두가 목적함수 값에 의한 上限 가지자르기였다는 점이 특징적이다. (不可能解에 의한 가

표-1　計算 實驗 結果

論文	教授	時間	경	模型 (P)							模型 (P_K)							模型 (P_N)					
文	授	間		線型		整數					線型		整數					整數					
I	J	K	우	Piv	OFV	No	Brn	Pivot	OFV	所要時間	Piv	OFV	No	Brn	Pivot	OFV	所要時間	No	Brn	Brn*	Pivot	OFV	所要時間
4	6	2	1	39	72.0	2	3	180	56	00 : 00 : 25	105	56.00	1	0	105	56	00 : 00 : 43	1	4	2	213	56	00 : 00 : 01
			2	41	52.0	2	4	251	40	00 : 00 : 30	100	40.00	1	0	100	40	00 : 00 : 41	1	6	5	270	40	00 : 00 : 01
			3	49	72.0	1	3	134	56	00 : 00 : 19	71	56.00	1	0	71	56	00 : 00 : 27	1	2	2	115	56	00 : 00 : 01
4	6	3	1	59	84.0	3	22	738	60	00 : 01 : 23	111	60.00	1	0	111	60	00 : 00 : 51	2	14	6	749	60	00 : 00 : 01
			2	70	92.0	2	13	433	68	00 : 00 : 55	127	69.71	1	2	168	68	00 : 01 : 20	1	19	6	1033	68	00 : 00 : 01
			3	56	80.0	2	12	401	64	00 : 00 : 54	117	64.00	1	0	117	64	00 : 01 : 02	1	13	11	784	64	00 : 00 : 01
6	6	3	1	127	112.0	2	52	3458	64	00 : 07 : 42	316	68.00	1	3	522	64	00 : 07 : 06	1	91	35	5954	64	00 : 00 : 06
			2	120	108.0	3	20	1146	80	00 : 02 : 50	214	80.00	1	0	214	80	00 : 02 : 39	1	45	33	2990	80	00 : 00 : 03
			3	108	144.0	2	23	1574	108	00 : 03 : 41	243	108.00	1	0	243	108	00 : 03 : 13	1	34	4	2541	108	00 : 00 : 03
6	6	4	1	119	140.0	3	120	5683	112	00 : 14 : 22	270	115.00	1	1	390	112	00 : 05 : 19	2	136	34	9850	112	00 : 00 : 11
			2	128	128.0	3	129	6304	92	00 : 15 : 42	296	94.00	1	1	376	92	00 : 05 : 03	2	324	273	23476	92	00 : 00 : 26
			3	130	140.0	1	58	2549	108	00 : 06 : 33	245	110.67	1	4	619	108	00 : 07 : 33	3	86	20	7243	108	00 : 00 : 08
8	6	4	1	192	184.0	3	131	11596	132	00 : 34 : 45	411	134.00	1	2	735	132	00 : 13 : 21	2	268	16	25640	132	00 : 00 : 32
			2	168	168.0	5	306	26697	116	01 : 18 : 02	613	116.00	1	0	613	116	00 : 13 : 06	2	351	111	33206	116	00 : 00 : 42
			3	195	168.0	3	176	15707	136	00 : 45 : 57	461	136.00	1	0	461	136	00 : 08 : 36	2	154	12	15587	136	00 : 00 : 20
8	6	5	1	190	192.0	2	25	909	144	01 : 40 : 26	578	150.67	1	3	3491	144	01 : 08 : 29	2	1703	144	181310	144	00 : 04 : 01
			2	227	192.0	6	353	24087	152	01 : 17 : 30	614	152.00	1	0	614	152	00 : 13 : 01	3	317	193	35679	152	00 : 00 : 48
			3	180	180.0	5	426	27250	116	01 : 41 : 55	573	131.33	2	2	3224	128	01 : 06 : 27	1	2509	281	256653	128	00 : 05 : 38
9	9	3	1	249	204.0	3	99	10332	176	00 : 31 : 11	680	176.00	1	0	680	176	00 : 19 : 28	3	109	41	12398	176	00 : 00 : 18
			2	204	196.0	5	167	18462	172	00 : 55 : 27	924	172.00	1	0	924	172	00 : 26 : 19	4	101	79	11757	172	00 : 00 : 17
			3	227	184.0	3	87	9302	148	00 : 28 : 24	568	148.00	1	0	568	148	00 : 15 : 11	2	158	147	16540	148	00 : 00 : 24
9	9	4	1	198	216.0	3	402	30824	188	01 : 42 : 50	975	197.00	1	1	1487	196	00 : 44 : 56	3	1008	28	121275	196	00 : 03 : 09
			2	261	216.0	2	89	4320	192	01 : 47 : 43	765	198.00	1	27	6297	192	02 : 26 : 55	5	1721	78	214694	192	00 : 05 : 35
			3	221	216.0	3	32	1229	168	01 : 41 : 58	844	182.00	2	31	8543	176	03 : 28 : 05	2	2238	186	260995	176	00 : 06 : 44
10	6	5	1	245	236.0	3	55	6009	172	01 : 59 : 48	811	184.00	1	1	855	184	00 : 25 : 43	2	2298	1291	299183	184	00 : 07 : 37
			2	228	220.0	3	227	26673	140	01 : 56 : 44	913	160.00	1	0	913	160	00 : 34 : 18	5	7417	6642	907245	160	00 : 22 : 46
12	9	4	1	328	276.0	2	40	3155	220	02 : 02 : 03	1764	236.00	1	0	1764	236	01 : 36 : 00	2	3732	1273	581732	236	00 : 17 : 01
			2	295	272.0	3	126	17991	228	02 : 00 : 31	2090	248.00	3	12	22966	244	21 : 13 : 55	2	2330	813	352880	244	00 : 10 : 16
12	12	3	1	324	288.0	1	11	713	240	01 : 54 : 03	2459	242.00	1	2	3765	240	03 : 28 : 08	4	2316	47	352344	240	00 : 10 : 10
12	12	4	1	290	288.0	5	62	4634	264	02 : 11 : 15	2464	280.00	4	36	22982	276	16 : 51 : 04	8	8000	4077	1341361	276	00 : 42 : 59

주) 1. ■ 표시 부분은 pivot수가 32767(2^{15}-1)이 되도록 enumeration이 끝나지 않은 경우임.

2. Piv는 pivot 수; OFV는 목적함수 값; No는 정수해의 수; Brn은 계산한 후보문제의 수; Brn*는 최적해 발견 후보문제의 위치.

지자르기는 計算實驗한 문제에서는 존재할 수 없다. |J|가 n의 배수가 아닌 문제에서만 존재함.)

V. 結 論

본 연구는 一定 심사기간內에서 심사위원들의 심사대상 논문 및 시간에 대한 選好度를 최대로 하는 시간표 작성문제의 일종인 학위논문 심사스케쥴 작성에 관한 것으로, 이에 대한 數理計劃模型을 작성하고, 이 모형의 解를 구하기 위한 선형계획 중심과 네트웍 중심의 두가지 解 접근방법을 제시하고, 이를 個人用 소형 컴퓨터 상에서 구현, 계산실험을 행해 비교 분석한 것이다. 계산실험 결과 두가지 제안방법 모두가 원래의 모형에 비해 계산속도면에서 탁월했으며, 그 중 네트웍 중심 分枝탐색방법은 선형계획중심 탐색방법에 비해 월등히 효과적이었다. 그러나 문제가 NP-complete 이므로 해서 네트웍 중심 분지방법도 문제의 크기가 증가하면서 解 소요시간의 급격한 증가를 보였다. 그러나 실제로 부딛히는 실질적인 대규모 문제에 대해서는 다음과 같은 방안을 생각해 볼 수 있다. 그 하나는 解 발견 반복과정에서 볼 수 있듯이 많은 경우 반복과정의 대부분이 이미 발견된 解가 최적인가를 확인하는 과정이며 문제의 屬性上 최적해를 구하는 것이 목표라기 보다 최적해에 근사한 亞최적해(sub-optimal solution)라도 실제 이용면에서는 적절하다고 보여지므로 일정 시간내에 반복절차를 마치는 방법을 고려할 수 있다. 대형 컴퓨터를 이용한다면 실질적 규모의 문제를 허용시간내에 최적으로 푸는 것도 가능하리라 판단된다.

본 연구는 특정 스케쥴링 문제에 관한 접근방법을 제시한 것이기도 하나 특수구조를 갖는 組合문제에서 쉬운 副문제(sub problem;이 경우 네트웍)중심의 방법과 切面을 많이 추가하여 整數 多面體에 가까운 문제로 만들어 접근하는 것의 두가지 접근방법을 적용했을 경우를 비교·계산해 보았다는 데서도 의의를 가질 수 있다.

본 연구는 학위논문 심사스케쥴링 문제만을 다룬 경우이나 이는 유사한 학술대회 日程작성 등에도 적용 가능한 방법이므로 새로운 적용문제의 개발에 힘 쓸 것이다.

參考文獻

〔1〕 신영수, "강의 시간표 작성 기법에 관한 연구," 『한국 OR학회지』, 8권 2호(1983. 10), pp. 62~72.

〔2〕 _____, "PC를 이용한 대학 강의 시간표 작성에 관한 연구," 「경영학 연구」, 17권 1호(1987. 9), pp. 125~140.

〔3〕 Almond, Mary, "An Algorithm for Constructing University Timetable," *The Computer Journal*, Vo. 8, No. 4(January 1966), pp. 331~340.

〔4〕 ______, "A University Faculty Timetable," *The Computer Journal*, Vo. 12, No. 3 (August 1968), pp. 215~217.

〔5〕 Andreu, Rafael and Albert Corominas, "SUCCCES92 : A DSS for Scheduling the Olympic Games," *Interfaces*, Vol. 19, No. 5 (September-October 1989), pp. 1~12.

〔6〕 Andrew, G. M. and R. Collins, "Matching Faculty to Courses," *College University*, 46 (1971), pp. 83~89.

〔7〕 Aust, A. J., "An Improvement Algorithm for School Timetabling," *The Computer Journal*, Vol. 19, No. 4 (November 1976), pp. 339~343.

〔8〕 Barraclough, Elizabeth D., "The Application of a Digital Computer to the Construction of Timetables," *The Computer Journal*, Vol. 8, No. 2 (July 1965), pp. 136~146.

〔9〕 Bradley, Gordon H., Gerald G. Brown and Glenn W. Graves, "Design and Implementation of Large Scale Primal Transshipment Algorithms," *Management Science*, Vol. 23, No. 1 (September 1977), pp. 1~34.

〔10〕 Breslaw, Jon A., "A Linear Programming Solution to the Faculty Assignment Problem," *Socio.-Econ. Plan. Sci.*, Vol. 10, No. 6 (1976), pp. 227~230.

〔11〕 Broder, Sol, "Final Examination Scheduling," *Comm. ACM* Vol. 7, (August 1964), pp. 494~498.

〔12〕 Cole, A. J., "The Preparation of Examination Time-Tables Using a Small-Store Computer," *The Computer Journal*, Vol. 7, No. 2 (July 1964), pp. 117~121.

〔13〕 Csima, J. and C. C. Gotlieb, "Tests on a Computer Method for Constructing School Timetables," *Comm. ACM*, Vol. 7, No. 3 (1964), pp. 160~163.

〔14〕 Dyer, James S. and John M. Mulvey, "An Integrated Optimization/Information System for Academic Departmental Planning," *Management Science*, Vol. 22, No. 12 (August 1976), pp. 1332~1341.

〔15〕 Ferland, Jacques A. and Serge Roy, "Timetabling Problem for Univeristy as Assignment of Activities to Resources," *Comput. & Ops. Res.* Vol. 12, No. 2 (1985), pp. 207~218.

〔16〕 Fisher, Marshall L. and Alexander H. G. Rinnooy Kan, "The Design, Analysis and Implementation of Heuristics," *Management Sceience*, Vo. 34, No. 3 (March 1988), pp. 263~265.

〔17〕 Foxley, E. and K. Lockyer, "The Construction of Examination Time-tables by

Computer," *The Computer Journal*, Vol. 34, No. 3 (November 1968), pp. 264～268.

[18] Glassey, C. Roger and Michael Mizrach, "A Decision Support System for Assigning Classes to Rooms," *Interfaces*, Vol. 16, No. 5 (September-October 1986), pp. 92～100.

[19] Gotlieb, C. C., "The Construction of Class-Teacher Timetable," *Proc. IFIP Congress 62*, North Holland Pub., 1963.

[20] Graves, Glenn W., *Mathematical Programming*, forthcoming.

[21] ______ and R. D. McBride, "The Factorization Approach to Large-Scale Linear Programming," *Mathematical Programming*, Vol. 10, No. 1 (February 1976), pp. 91～110.

[22] Junginger, Werner, "Timetabling in Germany - A Survey," *Interfaces*, Vol. 16, No. 4 (July-August 1986), pp. 66～74.

[23] Laporte, Gilbert and Sylvain Desroches, "Examination Timetabling by Computer," *Compul. & Ops. Res.*, Vol. 11, No. 4 (1984), pp. 351～360.

[24] ______, "The Problem of Assigning Students to Course Sections in a Large Engineering School," *Comput. & Ops. Res.*, Vol. 13, No. 4 (1986), pp. 387～394.

[25] Lawrie, N. L., "An Integer Linear Programming Model of a School Timetabling Problem," *The Computer Journal*, Vol. 12, No. 4 (November 1969), pp. 307～316.

[26] Lions, J., "The Ontario School Scheduling Program," *The Computer Journal*, Vol. 10, No. 1 (May 1967), pp. 14～21.

[27] Mehta, Nirbhay K., "The Application of a Graph Coloring Method to an Exam Scheduling," *Interfaces*, Vol. 11, No. 5 (October 1981), pp. 57～65.

[28] Pruul, E. A., G. L. Nemhauser and K. A. Rushmeier, "Branch-and-bound and Parallel Computation : A Historical Note," *Operations Research Letters*, Vol. 7, No. 2 (April 1988), pp. 65～69.

[29] Schrage, Linus, *User's Manual : Linear, Integer and Quadratic Programming with LINDO* (2nd ed.), The Scientific Press, 1985.

[30] Sherman, G. R., "A Combinatorial Problem Arising from Scheduling of University Classes," *Journal of the Tennessee Academy of Science*, Vol. 38, No. 3 (1963), p. 115.

[31] Tillett, P. I., "An Operations Research Approach to the Assignment of Teachers to Courses," *Soc-Econ. Plan. Sci.* 9 (3) (1975), pp. 101～104.

〔32〕 Tripathy, Arabinda, "School Timetabling - A Case in Large Binary Integer Linear Programming," *Management Science*, Vol. 30, No. 12 (December 1984), pp. 1473~1489.

〔33〕 Welsh, D. J. A. and M. B. Powell, "An Upper Bound for the Chromatic Number of a Graph and Its Application to Timetabling Problem," *The Computer Journal*, Vol. 10, No. 1 (May 1967), pp. 85~86.

〔34〕 de Werra, D., "Balanced Schedules," *INFOR J.*, Vol. 9, No. 3 (November 1971), pp. 230~237.

〔35〕 Winter, W. K., "A Scheduling Algorithm for a Computer Assisted Registration System," *Commun. ACM*, Vol. 14 (1971), pp. 166~171.

〔36〕 Wood, D. C., "A System for Computing University Examination Timetables," *The Computer Journal*, Vol. 11, No. 1 (May 1968), pp. 41~47.

〔37〕 ______, "A Technique for Colouring a Graph Applicable to Large Scale Timetabling Problems," *The Computer Journal* Vol. 12, No. 4 (November 1969), pp. 317~319.

〔38〕 Yang, Kwang Min, "On Academic Resource Scheduling," *Korea Management Review*, Vol. 11 (February 1982), pp. 221~242.

〔39〕 Yule, A. P., "Extension to the Heuristic Algorithm for University Time-tables," *The Computer Journal*, Vol. 10, No. 4 (February 1968), pp. 360~364.