

정수 일반네트워크문제를 위한 분지한계법의 개선

주재훈* · 김기석**

Improvements of Branch and Bound Algorithm for the Integer Generalized Network Problem

Jaehun Joo* and Kiseog Kim**

Abstract

A generalized network problem is a special class of linear programming problem whose coefficient matrix contains at most two nonzero elements per column. A generalized network problem with 0-1 flow restrictions is called an integer generalized network(IGN) problem. In this paper, we presented a branch and bound algorithm for the IGN that uses network relaxation. To improve the procedure, we develop various strategies, each of which employs different node selection criterion and/or branching variable selection criterion. We test these solution strategies and compare their efficiencies with LINDO on 70 randomly generated problems.

1. 서론

네트워크문제는 선형계획문제의 특수한 형태로서 제약조건식의 계수행렬(coefficient matrix)에서 각 열(column)의 0이 아닌 원소(nonzero elements)가 많아야 두개라는 일반적인 특성을 갖고 있다. 특히 계수행렬의 0이 아닌 원

소가 +1과 -1로 구성되어 있는 네트워크는 순수 네트워크(pure network)라 하며, 이때 0이 아닌 원소의 값에 대한 제한이 없는 네트워크가 일반 네트워크(generalized network)이다. 한편 일반 네트워크에서 어떤 가지(arc)상의 흐름이 0 혹은 1로 제약되어 있는 경우를 정수 일반네트워크문제(integer generalized network problem)라 하며, 아래의 (P)와 같이 표현된다.

* 동국대학교 정보산업학과

** 부산대학교 경영학과

$$(P) \quad z = \min C_1 X_1 + C_2 X_2$$

$$\text{s. t. } A_1 X_1 + A_2 X_2 = b$$

$$0 \leq X_2 \leq u$$

$$X_1 : 0 \text{ 혹은 } 1 \text{의 값을 갖는 이진변수벡터}$$

$$\text{단, } X = (X_1 \mid X_2)' = [x_j], j \in J = \{1, 2, \dots, n\}$$

$$J = J_1 \cup J_2, J_1 : \text{이진변수의 인덱스집합}$$

$$J_2 : X_2 \text{변수의 인덱스집합}$$

$$X_1 = [x_j], j \in J_1$$

$$A = [A_1 \mid A_2] = [a_{ij}], i \in I = \{1, 2, \dots, m\}, j \in J$$

A는 제약식의 좌변계수행렬로서 각 열에서 0이 아닌 원소가 많아야 2개이다.

$$b = [b_i], i \in I$$

$$C = [C_1 \mid C_2] = [c_j], j \in J$$

$$u : \text{연속변수의 상한벡터}$$

정수 일반네트워크모형은 지금까지 다양한 실제 문제에 적용되어 왔다. 예컨대 Crum과 Klingman 그리고 Tavis[8]는 확실성하에서의 투자 및 자본예산문제가 정수 일반네트워크로 표현될 수 있음을 제시한 바 있다. Glover 등[13]은 다품목생산시스템에서 최적 로트량 결정과 기계부하의 결정문제를, Bai[4]는 산림관리를 위한 일정계획문제를, Steinberg와 Napier[21]는 MRP시스템에서 최적 로트량 결정문제를 정수 일반네트워크문제로 표현한 바 있다. Barzilai와 Cook 그리고 Kress[5]는 가능한 대안의 집합(예를 들면, 프로젝트집합, 선거에서 후보자집합)에 대한 개개인의 선호(preference)가 쌍비교의 형태(pairwise comparison type)로 주어진 경우, 합의에 의한 대안의 우선순위를 결정하는 문제를 정수 일반네트워크 모형으로 표현한 바 있으며, 주재훈과 김기석[3]은 정수 일반네트워크모형을 유조선의 운항 일정계획문제에 적용한 바가 있다.

뿐만 아니라 Glover와 Mulvey[17]는 모든

순수 0-1 정수계획문제가 정수 일반네트워크로 표현될 수 있음을 증명한 바 있고, Ignizio[18]는 가중치가 부여된 정수목표계획문제가 정수 일반네트워크로 표현될 수 있고, 또한 목표간의 우선순위가 주어진 정수 목표계획문제가 가중치 정수 목표계획문제를 순차적으로 해결함으로써 해결될 수 있음을 보인 바 있다. 또한 정수계획문제를 제약식이 2개로 구성된 부분제로 분해하면 모든 정수계획문제가 정수 일반네트워크문제로 표현될 수 있다(Glover와 Klingman[14]은 이를 layering strategy이라 하였으며, 이는 적절한 Lagrangian Decomposition에 의한 모든 부분제가 정수 일반네트워크로 표현됨을 의미한다).

본 연구에서는 이러한 정수 일반네트워크문제를 위한 개선된 분지한계법을 제시하며, 특히 분지과정에서의 여러 분지전략들을 살펴보고 그 효율성을 검증하기 위하여 두 유형의 정수 일반네트워크를 대상으로 LINDO패키지와 비교실험을 하였다.

2. 정수 일반네트워크문제의 분지한계법

정수 일반네트워크문제는 정수계획모형으로 표현되므로 분지한계법을 적용하여 그 해를 구할 수 있다. 그러나 정수 일반네트워크문제의 경우에는 네트워크 구조를 이용한 분지한계법을 사용하면 더욱 효율적인 것으로 알려져 있다. 예컨대 Glover 등[13]은 0-1 정수계획문제로 정식화된 미공군조종사의 비행훈련 일정계획문제를 정수 일반네트워크로 표현하고, 분지한계과정에서 마디선택기준(node selection criterion)으로 LIFO전략을 사용하는 분지한계법으로 이 문제를 효율적으로 해결하였다. 또한 Glover와 Klingman 그리고 Phillips[15]는 혼합정수계획문제로 정식화된 원자로 연료보급 일정계획문제를 비선형 제약식이 포함된 네트워크문제로 변환하여 분지한계법으로 해를 구하였다. 이 연구에서 제시한 분지한계법은 문제에 주어진 비선형 제약식을 효율적으로 이용하고 있다. 정수 일반네트워크문제를 해결하기 위한 분지한계법의 분지한계과정은 이진트리 형태로 표현되므로 부문제를 마디라 정의하고 부문제의 해에서 이진변수 x_k 값을 f_k 라 정의하면 해산출과정은 아래와 같다.

단계 I : 마디의 선택(subproblem or node selection)

마디리스트(node list : L)에서 z (하한 : 부문제의 목적함수값) < \bar{z} (상한 : incumbent solution)인 마디(혹은 부문제) 중의 하나를 선택한다. 그러한 마디가 없으면 종료한다.

단계 II : 분지변수(branching variable)의 선택

단계 I에서 선택한 부문제에서 $f_k \neq 0$ 인 변수중의 하나를 분지변수로 선택한다. 그 변수를 x_k 라 둔다.

단계 III : 분지(branching)

다음 제약식을 단계 I에서 선택한 마디에서 다음 조건을 만족하는 두개의 부문제(마디)를 생성한다.

$$x_k = 0, x_k = 1$$

일반네트워크의 원단체해법에 의하여 이들 부문제의 해를 구하고 마디리스트에 삽입한다. 여기서 \bar{z} 보다 좋은 정수해가 산출되면 \bar{z} 를 갱신한다. 그리고 단계 I로 간다. 마디선택과정에서 $z \geq \bar{z}$ 인 부문제는 더 이상 고려할 필요가 없다(탐색되었다).

정수 일반네트워크문제를 해결하기 위한 위의 분지한계법의 효율성은 부문제의 해산출방법, 마디의 선택기준, 분지변수의 선택기준에 의존한다.

본 연구에서는 부문제의 네트워크구조를 이용할 수 있는 네트워크 이완(network relaxation)에 의한 부문제의 해, 즉 일반 네트워크문제의 해를 산출하는데 있어서는 주재훈[1]의 연구에 기초를 두고 있다. 여기서 부문제의 기저 그래프를 표현하기 위한 자료구조로는 EATI (extended augmented threaded index)법을 사용하였고[19]. 초기 실행가능 기저그래프는 여유마디(slack node)를 정의하고 필요한 경우 인위가지(artificial arc : 비용이 big-M이고, 승수가 1인 가지)를 도입하여 산출된다. 또한 진입가지결정방법으로는 크기가 N^2 인 한 개의 리스트(list)를 이용하였다. N^2 개의 비기저가지를 리스트에 채워두고, 이들 중에서 최선의 가치를 선택한다. 리스트에 진입가지로 선택할 가치가 없거나 리스트를 N^1 번 이상 사용하게 될 경우

에는 다시 N2개를 리스트에 채운다. 또한 진입 가지에 의하여 현 기저그래프에서 흐름이 변화하는 가지의 흐름변화량을 산출하기 위해서는 진입가지가 서로 다른 하위그래프를 연결하는 경우와 같은 기저그래프의 두 마디를 연결하는 두 경우에서 각각 사이클이 포함되는 경우와 사이클이 포함되지 않는 경우로 구분되어 흐름 변화량이 계산되어 진출변수의 결정과정이 단순화되었다.

또한 분지한계법의 분지전략으로 기존의 정수계획문제에서 종종 사용되어 왔던 LIFO규칙이나 페널티(penalty)에 의한 방법은 계산시간 관점에서 비효율적이다[12]. 따라서 본 연구에서는 다양한 부문제(즉, 마디)의 선택기준과 주어진 부문제에서의 다양한 변수선택기준을 사용하고자 하였다. 또한 다수의 활동집합 중에서 하나의 활동을 선택해야 하는 문제에서 종종 나타나는 다중선택제약식(multiple choice constraints)을 효율적으로 처리하기 위하여 special ordered sets의 개념을 적용하고자 하였다.

3. 정수 일반네트워크 분지한계법의 개선전략

본 연구에서는 정수 일반네트워크의 분지한계법을 개선하기 위하여 다음 기준들에 대한 다양한 전략들을 개발하여 이들의 여러 조합간의 효율성을 비교하고자 한다.

- ① 마디의 선택기준
- ② 분지변수의 선택기준

3. 1 마디선택기준

Forrest 등[12]에 의하면 LIFO규칙이나 페널티를 사용하는 기준은 분지한계과정을 통제하는 기준으로 적절하지 못하다는 것이 입증되었고, best projection과 pseudo-cost기준이 그들보다 효율적이라는 사실이 밝혀진 바 있다.

따라서 본 연구에서는 Forrest 등의 기존 정수계획문제를 해결하기 위한 분지한계전략에서 이용된 best projection기준과 pseudo-cost기준을 기초로 하고 또한 이들의 변형방법을 사용하여 정수 일반네트워크문제를 해결하고자 한다.

(1) BP기준

BP기준(best projection criterion:이하에서는 BP기준이라 함)에서는 마디 p에서 기대할 수 있는 정수해의 근사적인 목적함수값의 추정치(\hat{z}_p)를 구하여 이 값이 최소인 마디를 선택한다.

$$\hat{z}^p = z^p + \lambda \cdot s^p \quad (1)$$

단, z^p : 부문제(마디 p: 일반네트워크로 이완한 문제)의 최적 목적함수값

$$s^p = \sum_{j \in J_p} \min\{f_j, 1-f_j\}, \quad \forall p: \text{마디 } p \text{ 에}$$

서의 모든 정수변수(J_p)의 정수실행불가능성의 합(sum of integer infeasibilities)

λ : $(\bar{z}-z^p)/s^p$: z^p 에서 \bar{z} 까지 목적함수값의 증가율에 대한 경험적인 추정치로서, 정수실행불가능성의 합이 s^p 에서 0까지 단위감소당 목적함수값의 증가치

z^0 : 초기 부문제(마디 0), 즉 최초의 정수 일반네트워크를 일반네트워크로 이완한 문제의 목적함수값

s^0 : 초기 부문제의 정수실행불가능성의 합

\bar{z} : 최적 정수해의 목적함수값의 상한(초기에는 입력값으로 제공)

따라서 (1)식의 \hat{z}^p 는 z^p 를 하한으로 하여 정수실행불가능성의 함(s^p)과 선형관계에 있다고 가정하고 구한 정수해의 목적함수값에 대한 휴리스틱 추정치라 할 수있다.

(2) PC기준(pseudo-cost criterion)

BP기준에서는 모든 정수변수가 $x_j=f_j$ 에서 $x_j=0$ 으로 단위 감소에 소요되는 비용과 $x_j=1$ 로 단위 증가에 소요되는 비용이 같다고 가정하고 있다. 그러나 사용자가 그러한 단위비용에 관한 정보를 제공할 수 있거나 문제의 구조에서 단위비용에 관한 정보를 얻을 수 있는 경우, 각 정수변수가 0과 1로 단위감소 혹은 증가되는 경우에 소요되는 개개 변수의 pseudo-cost를 이용하여 마디 p에서의 추정치를 (2)식과 같이 산출할 수 있다. 따라서 (2)식에서는 개개 정수변수의 단위증가와 단위감소에 따른 목적함수값의 변화를 달리 반영하고 있다.

$$\hat{z}^p = z^p + \sum_{j \in J} \min\{D_j f_j, U_j (1-f_j)\} \quad (2)$$

단, D_j : 정수변수 x_j 의 단위감소당 추정된 비용
 U_j : 정수변수 x_j 의 단위증가당 추정된 비용

사용자가 D_j 와 U_j 에 관한 정확한 정보를 제공할수록 추정치 \hat{z}^p 는 빨리 최적해에 근사해 간다고 할 수 있다. 그러나 사용자가 D_j 와 U_j 에 관한 정보를 제공할 수 없는 경우, (3)식과 같이 이들의 초기값이 추정된다. (3)식에서는 최초의 D_j 와 U_j 는 BP기준의 λ 와 같다고 가정한다.

$$D_j = U_j = (\bar{z} - z^p) / s^p \quad (3)$$

단, \hat{z} : 정수해의 목적함수값에 대한 추정치

그러나 분지한계과정의 마디 p에서 분지변수 $x_k(=f_k \neq 0)$ 에 의한 두개의 부문제, 즉 $x_k=0$ 과 $x_k=1$ 에 의해 생성된 부문제를 각각 마디 p+1과 p+2라 하면 D_j 와 U_j 는 (4)식과 (5)식과 같이 갱신된다. 따라서 (4)와 (5)식에서는 이미 주어진 D_j (혹은 U_j)와 분지과정에서 얻는 새로운 정보에 의한 정수실행불가능성 단위당 목적함수값의 변화량을 평균하여 D_j (혹은 U_j)를 수정하고 있다.

$$D_j = [D_j + (z^{p+1} - z^p) / f_j] / 2 \quad (4)$$

$$U_j = [U_j + (z^{p+2} - z^p) / (1-f_j)] / 2 \quad (5)$$

(3) PC기준의 변형

초기 실행가능 정수해를 산출하기까지는 이진트리리의 깊이우선탐색(depth first search)을 기초로 마디를 선택하되, 어떤 마디의 left branch와 right branch의 두마디 중에서는 PC기준에 의하여 부문제를 선택한다. 따라서 여기서는 마디의 선택범위를 제한하므로써 가능한 빨리 초기 실행가능해를 산출하고자 한다. 일단 초기 실행가능 정수해가 산출되면 탐색이 끝나지 아니한 부문제의 집합(L)에서 현 정수해를 가장 빨리 개선시킬 가능성이 있는 마디 p^* 를 선택한다. $\bar{z} > z^p$ 인 탐색되지 않은 마디 p에서 가능한 빨리 목적함수값을 향상시키는 마디를 선택하는 기준으로 (6)식에서는 $\bar{z} - z^p$ 가 크고, 추정치(\hat{z}^p)가 그 하한(z^p)에 가까운 마디 p^* 를 선택하고 있다.

$$p^* = \min_{p \in L} (\bar{z} - z^p) / (z^p - z^p) \quad (6)$$

(4) SOS에서의 BP기준

SOS(special ordered sets)의 개념은 Beale과 Tomlin[6]에 의해 비볼록 최적화문제(nonconvex optimization problem)의 해를 산출하는 데 적용되었다. SOS에는 S1집합(special ordered sets of type 1)과 S2집합(special ordered sets of type 2)이라는 두 유형이 있다. S1집합이란 최종해(final solution)에서 집합내에 있는 많아야 한 변수가 0이 아닌 경우를 의미하고, S2집합이란 많아야 집합내의 인접한 두개의 변수가 0이 아닌 경우이다. 두번째 유형의 SOS개념은 주로 비선형계획문제의 최적화기법에 사용되고 있다.

많은 현실 문제에서는 어떤 활동(activities) 집합에서 정확히 하나 이하의 활동을 선택해야 하는 경우가 발생하는 데, 이러한 문제는 (7)식과 같은 형태로 표현된다. 이러한 제약식의 형태로 성립하는 변수집합은 첫번째 유형의 SOS이다. 이러한 경우에는 분지한계과정에서 개변수보다는 변수집합을 대상으로 하여 분지한계과정을 처리하는 것이 보다 효율적이다.

$$\begin{aligned} x_1 + \dots + x_2 + \dots + x_{n'} &\leq 1, \\ x_j &= \{0, 1\}, j = 1, \dots, n' \end{aligned} \quad (7)$$

(7)식에서 변수의 인덱스를 임의로 $j=1, \dots, n'$

이라 하면 n' 개로 구성된 어떤 변수집합 중에서 많아야 하나가 1의 값을 취하게 된다. 3.2의 SOS에 의한 변수선택기준의 조건 3)에서 같이 n' 개의 변수 중에서 j_0 번째 변수가 분지 변수로 선택되었다고 하면, 분지과정의 현 마디에서 $x_1=x_2=\dots=x_{j_0}=0$ 인 left branch의 부문제에서는 $x_{j_0+1}+\dots+x_n \leq 1$ 을 충족하는 해가 산출된다. 그 후 left branch의 부문제를 계속 분지해가면 (7)식을 만족하는 해가 산출되든가 혹은 실행불가능해가 산출된다. 한편 $x_{j_0+1} \dots x_n \leq 0$ 으로 고정되는 right branch의 부문제에서는 $x_1+\dots+x_{j_0} \leq 1$ 을 만족하는 해가 산출되고 그 이후의 과정은 left branch의 경우와 같다. 따라서 분지한계법에서는 x_1, \dots, x_{j_0} 모두가 0이거나 x_{j_0+1}, \dots, x_n 모두가 0이 되도록 분지한다. 즉, SOS개념을 사용하여 분지한계트리(branch and bound tree)의 임의의 마디에서 개변수에 대하여 분지하지 않고 변수집합을 대상으로 하여 분지한다.

(8)식에서는 정수실행불가능성 단위당 목적함수값의 증가율(λ)에 변수집합을 대상으로 산출한 어떤 마디의 정수실행불가능성의 합(W)을 곱하고 여기에 그 하한을 더한 것이 최소인 마디를 선택하는 기준이다.

$y_k = \max_{j \in s_k} \{x_j\}$, 단, s_k : k번째 SOS에 있는 변수의 인덱스집합

$S_{sk} = 1 - y_k$: k번째 변수집합의 정수실행불가능성

$$N_{sk} = \begin{cases} 1 : y_k \text{가 분수값을 갖는 경우} \\ 0 : y_k \text{가 0 혹은 1의 값을 갖는 경우} \end{cases}$$

SS = $\sum S_{sk}$: 변수집합의 정수실행불가능성의 합

SN = $\sum N_{sk}$: 분수값을 갖는 변수집합의 수

$W = u \cdot SN + (1-u) \cdot SS$

단, u : $0 \leq u \leq 1$ 의 가중치

$$z^p = z^p + \lambda \cdot W$$

(8)

3. 2 분지변수의 선택기준

사용자가 정수변수간의 우선순위(priorities)를 제공할 수 있는 경우, 이러한 우선순위에 의한 변수선택기준은 매우 효율적이다. 예를 들어 어떤 프로젝트를 선택할 것인가, 선택하지 않을 것인가와 관련된 0-1변수가 그 프로젝트내의 어떤 세부적 의사결정과 관련된 정수변

수 보다 높은 우선순위를 갖는다. 그러나 대부분의 경우 문제의 구조로부터 변수간의 우선순위를 정의하기가 쉽지 않다(이에 대해서는 5에서 상세히 설명하기로 한다). 따라서 본 연구에서는 다음과 같이 4개의 변수선택기준을 고려하였고, IGN 시스템에서는 변수간의 혹은 변수집합간의 우선순위를 제공할 수 있는 경우에는 이를 이용할 수 있도록 하였다.

- ① $\max \min\{f_j, 1-f_j\}$: 이하에서는 maxmin기준이라 한다.
- ② $\min\{f'_j\}$, $f'_j = |0.3-f_j|$, $f_j \leq 0.5$ 인 경우,
 $f'_j = |0.7-f_j|$, $f_j > 0.5$ 인 경우 : 이하에서는 min기준이라 한다.
- ③ $\max \min\{D_j f_j, U_j(1-f_j)\}$: 이하에서는 pseudo-cost(PC)기준이라 한다.
- ④ SOS에서 변수선택기준 : 이하에서는 SOS기준이라 한다.

①에서는 0.5 부근의 값을 갖는 변수를 선택하는 기준으로 보수적 관점에서 가능한 정수실행불가능성이 큰 변수를 빨리 정수조건을 충족시키게 하므로써 실행가능해를 산출하고자 한다.

②에서는 0.3 혹은 0.7 부근의 값을 갖는 변수를 선택하는 기준으로 0.3부근의 값을 갖는 변수는 0으로 0.7부근의 값을 갖는 변수는 1로 고정시키므로써 정수실행불가능성을 빨리 감소시킬 수 있다는 점에 착안하고 있다.

③은 pseudo-cost를 고려한 변수선택기준이다.

④의 SOS의 개념을 적용한 경우의 변수 선택기준은 아래와 같다.

조건 1) : $N_{sk} = 1$ 을 만족하는 변수집합.

조건 2) : 조건 1)을 만족하는 변수집합중에서 S_{sk} 가 최대인 변수집합

조건 3) : 조건 2)를 만족하는 변수집합내에서 분지변수는 다음 기준에 의하여

선택된다
 선택된 변수집합내의 개개변수를 $x_j(1 \leq j \leq n')$ 라 할 때,
 x_j 의 가중평균 we 를 구하여
 $j_0 = [we]$ 에 상응하는 변수
 x_{j_0} 를 분지변수로 선택한다.
 $we = \sum_j \cdot x_j, j \in sk$

주어진 부문제에서 정수조건을 충족하지 않는 분지변수가 주어지면 두 개의 새로운 부문제가 생성되는데, 이때 어느 부문제를 먼저 선택하느냐에 따라 최적해에 도달하기까지 생성된 총부문제의 수에 영향을 미친다. 본 연구에서는 $x_k=0$ 에 의한 부문제(left branch)를 선택하고, 다음으로 $x_k=1$ 에 의한 부문제(right branch)를 선택하는 기준을 사용하였다.

3. 3 분지한계법의 개선전략

정수 일반네트워크문제의 LIFO규칙에 의한 분지한계법을 개선하기 위하여 본 연구에서는 4

개의 마디선택기준과 4개의 분지변수선택기준에 따라 <표 1>과 같이 9개의 전략을 사용하였다.

<표 1> 분지한계전략

마디선택기준 분지변수 선택기준*	BP기준	PC기준		PC기준의 변형	SOS에서 BP기준	
① maxmin기준	BP-MAXMIN전략					
② min기준	BP-MIN기준	PC-MIN 전략		VPC-MIN전략		
③ PC기준		PC전략	PCP전략	VPC전략		
④ SOS 기준					SOS전략	SOSP전략

* ① $\max \min\{f, 1-f\}$

② $\min\{f, f'\}$, $f' = |0.3-f|$, $f_i \leq 0.5$ 인 경우,
 $f' = |0.7-f|$, $f_i > 0.5$ 인 경우

③ $\max \min\{Df, U(1-f)\}$

④ SOS에서 변수선택기준

<표 1>에서 BP-MAXMIN(Best Projection for node selection and MAXMIN for branching variable selection) 전략과 BP-MIN (Best Projection for node selection and MIN for branching variables) 전략은 둘다 마디선택으로 BP기준을 사용하나 분지변수선택에 있어서는 BP-MAXMIN 전략에서는 0.5부근의 값을 갖는 변수를 BP-MIN 전략에서는 0.3 혹은 0.7부근의 값을 갖는 변수를 선택하는 기준을 사용하고 있다. 마디선택기준으로 BP 기준을 사용하는 경우, <표 1>에서 분지변수선택 기준으로 ③과 ④, 즉 개개 변수에 대한 pseudo-cost를 사용하는 기준이나 SOS 기준은 의미가 없다.

PC-MIN(Pseudo-Cost for node selection

and MIN for branching variable) 전략, PC (Pseudo-Cost for both selection of node and branching variable) 전략, PCP(Pseudo-Cost for both selection of node and branching variable with Priority) 전략은 마디선택기준으로 모두 PC기준을 사용하나, PC-MIN 전략에서는 0.3 혹은 0.7부근의 값을 갖는 변수를, PC전략에서는 pseudo-cost에 의한 분지변수선택기준을 사용하고 있다. 한편 PCP전략은 PC 전략에서 이진변수간의 우선순위를 사용할 수 있도록 한 것이다. 그러나 마디선택기준으로 PC 기준을 선택하는 경우, 0.5부근의 값을 갖는 변수를 분지변수로 선택하는 기준은 적절하지 못하다. 왜냐하면 pseudo-cost가 계산되는 경우, pseudo-cost기준이 그 보다 우월하기 때문이다.

VPC-MIN(Variant of PC MIN) 전략과 V-PC(Variant of PC) 전략에서는 둘다 초기 실행가능해가 산출되기까지 마디선택을 제한하는 PC기준의 변형된 마디선택기준을 사용하나 분지변수선택으로 전자에서는 0.3 혹은 0.7기준을 후자에서는 pseudo-cost 기준을 사용하고 있다.

끝으로 SOS로 나타나는 정수 일반네트워크문제를 해결하기 위한 SOS전략(Special Ordered Set strategy)과 SOSP(SOS with Priority)전략간의 차이점은 SOSP전략이 SOS전략과 같은 기준으로 사용되나 단지변수집합에 대한 우선순위를 사용하고 있다는 점이다.

9개의 분지전략에 따라 정수 일반네트워크문제의 해를 산출하기 위한 컴퓨터 코드(이하에서는 IGN 시스템이라 함)는 FORTRAN으로 작성되었다.

4. 실험설계

본 연구에서는 IGN 시스템의 전략간의 효율성 비교 및 기존의 0-1 정수계획문제를 해결하는 데 많이 이용되는 LINDO 시스템과의 비교에서 실험의 객관성과 신뢰성을 높이기 위하여 다음과 같은 세가지 기준을 설정하였다.

첫째, 그들간의 효율성을 계산시간(CPU시간) 측면에서 비교하데, 평균계산시간 및 최악의 경우(worst case)에서의 계산시간 두 측면을 고려한다. 둘째, 분지한계법에서 서로 다른 마디선택 기준과 분지변수선택기준에 따라 최적해를 산출하기까지의 해결해야 할 부문제의 수가 다를 수 있다. 비록 IGN시스템의 모든 전략에서 네트워크 이완에 의하여 부문제의 해를 구하지만 마디와 분지변수의 선택기준이 계산시간에 영향을 미치므로 최적해를 산출하기까지의 부문제를 측정하여 간접적으로 전략간의 효율성을 비교한다. 셋째, 서론에서 설명한 바와 같이 다양한 현실 문제가 IGN으로 표현되지만 일반적으로 잘 알려진 문제를 실험대상으로 설정한다. 그리고 각 문제에서 임의성을 배제하기 위하여 문제의 계

수는 난수를 발생하여 그 값을 설정한다.

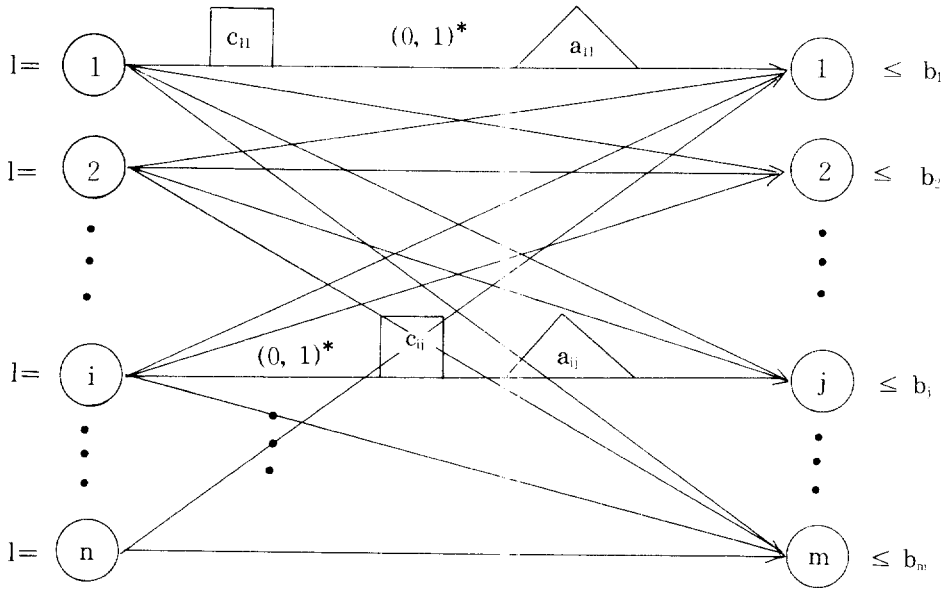
4. 1 실험 대상문제

분지전략간의 효율성을 검증하기 위하여 일반 할당문제(Generalized Assignment Problem : 이하에서는 GAP라 함)와 프로젝트 선택 및 자본예산문제(Project Selection and Capital Budgeting Problem : 이하에서는 CBP라 함)를 실험대상문제로 선정하였다.

(1) 일반 할당문제

일반 할당문제란 n 개의 과업을 m 개의 작업장에 할당하는 경우, 각 작업장의 능력을 초과하지 않고 과업할당에 따른 비용이 최소가 되도록 각 과업을 정확하게 하나의 작업장에 할당하는 문제이다. 일반 할당문제는 각 과업의 할당에서 소요되는 자원이 1로 제한되지 않고 0-1 이진변수만으로 구성된 문제라는 점에서 정수일반네트워크문제의 특수한 형태이다.

[그림 1]에서는 일반 할당문제의 정수 일반네트워크를 표현하고 있다.



[그림 1] 일반 할당문제의 네트워크

[그림 1]에서 사용된 기호는 아래와 같다.

c_{ij} : 마디 i 에서 마디 j 로의 단위흐름비용

a_{ij} : 가지승수(마디 i 에서 공급되는 한 단위 흐름이 마디 j 에 도달하면 a_{ij} 가 된다. 즉, $a_{ij} > 1$ 인 경우에는 gain, $a_{ij} < 1$ 인 경우에는 loss)

b_j : 작업장 j 의 능력

$(0, 1)^*$: 마디 i 와 마디 j 를 잇는 가지에서의 흐름하한은 0이고, 흐름상한은 1이다. 또한 이 가지상의 흐름은 0 혹은 1로 제약된다.

n : 과업의 수

m : 작업장의 수

[그림 1]에서는 각 과업에 대하여 하나의 공급 마디 i 가 정의되고, 각 작업장에 대하여 하나의 수요마디 j 가 정의된다. 마디 i 에서 마디 j 로 단위흐름을 보내는 경우 그 비용은 c_{ij} 이고, 마디 i 에서 단위흐름을 보내면 가지 (i, j) 에서 가지승수(a_{ij})만큼 흐름이 변화하여 마디 j 에 도착한다. 따라서 [그림 1]에서는 각 공급마디의 공급량 1을 충족시키고 각 수요마디의 능력을 초과하지 않고, 각 가지의 흐름이 0 혹은 1

이 되는 총 비용이 최소인 가지상의 흐름을 결정하는 문제이다.

(2) 자본예산문제

자금의 공급곡선이 증가하는 불완전자본시장에서 자본예산에 관한 Weingartner의 계획기간모형(Weingartner's horizon model)은 [그림 2]와 같은 정수 일반네트워크로 표현된다[8].

이 모형은 일정한 계획대상기간 동안 매기간 기업운영으로부터 기대되는 자금의 한계내에서 일정기간후 차입과 대출 및 투자에 의한 가치

가 최대가 되도록 다수의 투자안에 자원을 할당하는 문제이다.

[그림 2]에서 사용된 기호는 다음과 같다.

j : 프로젝트에 해당하는 인덱스, $j = 1, \dots, n$ (단, n 은 투자 가능한 프로젝트의 수)

t : 계획대상기간에 해당하는 인덱스, $t = 1, \dots, T$ (단, T 는 계획대상기간의 수)

a_{jt} : 프로젝트 j 를 수락하는 경우, 기간 t 의 자금흐름($a_{jt} < 0$: 자금유출)

c_j : j 번째 프로젝트의 현금흐름을 계획기간 T 를 기준으로 하여 특정의 이자율로 할인한 모든 흐름의 가치

D_t : 기간 t 의 기업운영으로부터 발생할 것으로 기대되는 자금

r : 대출이자율

r_i : $r_{i-1} < r_i < r_{i+1}$ 를 만족하는 한계자금공급곡선의 i 번째 단계에 적용하는 차입이자율

B_{it} : 기간 t 에 이자율 r_i 로 차입가능한 자금상한

m_j : 프로젝트 j 를 수락하는 경우, 현금흐름이 발생하는 기간의 수

(0, 1)*: 가지의 흐름하한은 0이고 흐름상한은 1이며, *는 가지흐름이 0 혹은 1로 제약됨을 의미한다.

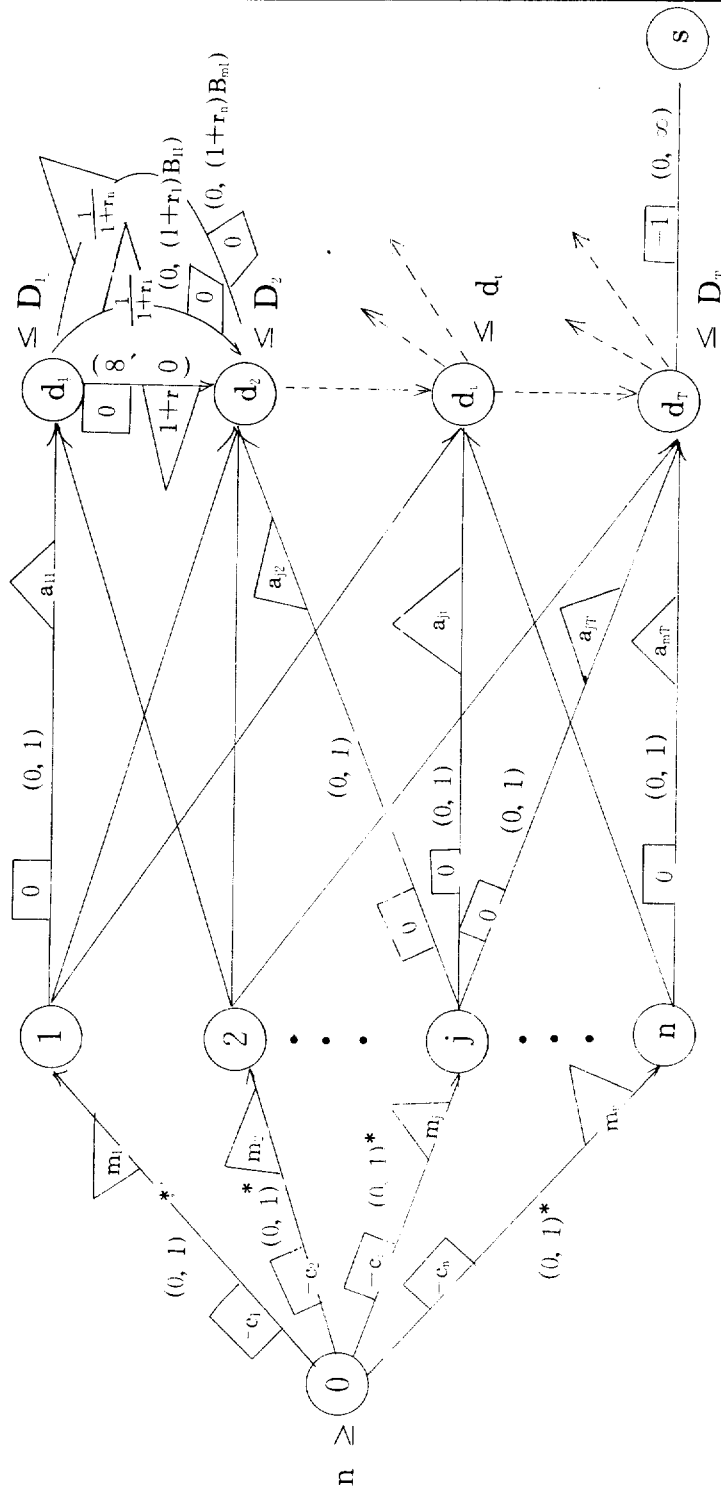
[그림 2]에서는 프로젝트의 수(n)에 해당하는 공급량을 갖는 하나의 마디 0이 정의되고 각 프로젝트에 대하여 하나의 중간마디 j 가 정의된다. 그리고 각 기간에 따라 하나의 마디 d_t 가 정의되며 각 마디 d_t 에서의 자금의 능력은 D_t 로 제한된다. j 번째 프로젝트에 대한 가지 (0, j)에서 c_j 는 프로젝트 j 의 순현재가치이고 가지 승수 m_j 는 이 프로젝트로 현금흐름이 발생하는 기간의 수이다. 또한 j 번째 프로젝트를 수락할 것인가 아닌가를 문제이므로 가지 (0, j)에서의 흐름은 0 혹은 1로 제한된다. 가지 (j , t)에서 가지승수 a_{jt} 는 j 번째 프로젝트에 의한 기간 t 에 발생하는 현금흐름이다. 가지 (d_t , d_{t+1})에서 가지승수 $1+r_t$ 은 기간 1에서 기간 2까지 자금을 이자율 r_t 로 대출하는 경우의 증가계수이며, 대출한계에는 제한이 없으므로 이 가지의 흐름상한은 ∞ 이다. 끝으로 가지 (d_t , d_{t+1})에서 가지

승수는 기간 2에서 기간 1로 차입한 자금을 차입이자율 r_t 로 할인한 것이며, 가지상한은 각 기간의 차입가능한 자금에 의하여 산출된다.

4. 2 실험대상문제의 생성

본 연구에서는 GAP와 CBP를 생성하기 위한 문제발생프로그램을 FORTRAN언어로 작성하였다. 이 프로그램에서는 IGN system과 LINDO 각각의 입력양식에 맞는 GAP와 CBP 문제를 산출하게 된다.

본 연구에서는 <표 2>에서와 같이 일반 할당 문제를 공급마디와 수요마디의 수 그리고 가지 승수의 범위에 따라 네 유형의 GAP를 문제생성기로 산출하였다. <표 2>에서 사용한 기호는 [그림 1]에서 설명한 기호의 내용과 같다. 각



[그림 2] 자본예산문제의 정수 일반네트워크

유형의 문제 각각에 대하여 난수발생 seed에 따라 10개의 문제, 즉 총 40문제가 산출되었다.

〈표 2〉에서 GAP 1과 GAP 1-1은 과업의 수가 10개이고 작업장의 수가 3개인 문제이나 가지승수와 작업장의 능력을 산출하는 범위가 다

르다. 또한 GAP 2와 GAP 2-1은 GAP 1이나 GAP 1-1보다 문제의 크기를 증가시킨 것으로 난수 발생범위를 서로 달리 주었다는 점에서 GAP 2와 GAP 2-1의 차이가 있다.

본 연구에서 계산실험용으로 산출된 프로젝트

〈표 2〉 GAP의 유형과 문제산출자료

	GAP 1	GAP 1-1	GAP 2	GAP 2-1
	n=10 m=3	n=10 m=3	n=10 m=4	n=10 m=4
문제크기*	BV=30 NODE=13	BV=30 NODE=13	BV=40 NODE=14	BV=40 NODE=14
난수 발생 영역	$c_{ij} : 5.0 \sim 25.0$ $a_{ij} : 5 \sim 25$ $b_j = 0.8 \sum_i a_{ij} / m$	$c_{ij} : 5.0 \sim 25.0$ $a_{ij} : 0.5 \sim 2.5$ $b_j = 0.8 \sum_i a_{ij} / m$	$c_{ij} : 5.0 \sim 25.0$ $a_{ij} : 5 \sim 25$ $b_j = 0.8 \sum_i a_{ij} / m$	$c_{ij} : 5.0 \sim 25.0$ $a_{ij} : 0.5 \sim 2.5$ $b_j = 0.8 \sum_i a_{ij} / m$

* BV : 이진변수의 수, 즉, 네트워크에서 이진가지의 수
NODE : 제약식의 수, 즉, 네트워크에서 마디의 수

트선택 및 자본예산에 관한 문제는 세 유형의 CBP로 구성되어 있다. 각 유형의 CBP에서 각각 10문제씩, 즉 30문제가 문제생성기에서 산출되었다. 각 유형의 CBP의 산출에 사용된 자료는 〈표 3〉과 같다. 〈표 3〉에서 사용한 기호는 [그림 2]에서 설명한 내용과 같고, CBP 1에서는 계획대상기간을 8로 설정하고 이 기간 동안의 프로젝트의 수를 5개, 각 프로젝트에서 발생하는 현금흐름 기간의 수를 4기간으로 설정하였다. 현금흐름이 발생하는 특정 기간은 무작위로 선정한다. CBP 1을 네트워크로 표현하면 이진가지가 5개, 연속가지가 47개, 마디의 수가 14개인 문제가 된다. 또한 CBP 2나 CBP 3는 CBP 1에서 문제의 크기를 증가시킨 것이다.

계산실험에 이용한 컴퓨터는 IBM 16bit AT

기종의 대우 코로나로 Coprocessor가 내장되어 있지 않고, 램메모리(RAM memory)는 640KB이다. 그리고 포트란 프로그램을 컴파일하는 데는 MS FORTRAN Compiler Ver. 4. 0을 이용하였으며, 계산시간 측정을 위해서는 컴파일러에 내장되어 있는 함수 GETTIM()을 이용하였다. 또한 LINDO에서의 계산시간을 측정하기 위해서는 초시계(stopwatch)를 이용하였다.

5. 실험결과

40개의 일반 할당문제(GAP 1, GAP 1-1, GAP 2, GAP2-1)와 30개의 자본예산문제(CBP 1, CBP 2, CBP 3), 즉 총 70문제에 대한 IGN

〈표 3〉 CBP의 유형 및 문제산출 자료

	CBP 1	CBP 2	CBP 3
	n=5 T=8 m _j =4	n=10 T=8 m _j =3	n=10 T=10 m _j =2
문제의 크기*	BV=5 CV=47 NODE=14	BV=10 CV=62 NODE=19	BV=10 CV=58 NODE=21
난	c _j : 10.0~40.0	c _j : 10.0~40.0	c _j : 10.0~40.0
수	a _{ij} : 1.5~3.0	a _{ij} : 1.0~2.5	a _{ij} : 1.5~3.0
발	-a _{ij} : -9.0~-5.5	-a _{ij} : -9.0~-5.5	-a _{ij} : -9.0~-5.5
생	D _i =sum _i /3.25	D _i =sum _i /4.00	D _i =sum _i /4.00
영	sum _i =∑ _j a _{ij}	sum _i =∑ _j a _{ij}	sum _i =∑ _j a _{ij}
역	1+r=1.1	1+r=1.1	1+r=1.1
	1/(1+r ₁)=0.9	1/(1+r ₁)=0.9	1/(1+r ₁)=0.9
	1/(1+r ₂)=0.8	1/(1+r ₂)=0.8	1/(1+r ₂)=0.8
	(1+r ₁)B _{1i} =2.5	(1+r ₁)B _{1i} =2.5	(1+r ₁)B _{1i} =2.5
	(1+r ₂)B _{2i} =2.5	(1+r ₂)B _{2i} =1000.0	(1+r ₂)B _{2i} =1000.0

* BV : 이진변수의 수, 즉 네트워크에서 이진가지의 수

CV : 여유변수를 제외한 연속변수의 수, 즉 네트워크에서 연속가지의 수

NODE : 제약식의 수, 즉 네트워크에서 마디의 수

시스템과 LINDO의 계산시간 및 IGN 시스템에서 각 분지전략간의 비교결과는 〈표 4〉 - 〈표 8〉과 같다. 이들 표에서 ①은 10문제의 평균에 의한 CPU seconds(branch의 수)이고, ②는 10 문제 중에서 최대 CPU seconds(branch의 수)를 나타내고 있다.

〈표 4〉와 〈표 5〉에서 SOSP전략의 변수집합에 대한 우선순위를 제공하는 방법과 같다. [그림 1]의 일반 할당문제에서 네트워크의 모든 가지 중에서 최대비용을 25.0이라 하고 가지 (i, j)의 가지비용을 c_{ij} 그리고 가지승수를 a_{ij}라 하면, 네트워크의 최대비용과 가지 (i, j)의 비용과의 차이를 작업장 i에서의 능력소요량으로 나눈, uc_{ij} = (25.0 - c_{ij})

/a_{ij}를 산출한다. 그리고 k번째 변수집합에서 최대의 uc_{ij}와 그 다음의 uc_{ij}와 차이가 큰 변수집합에 높은 우선순위가 제공된다.

〈표 6〉 - 〈표 8〉의 PCP전략에서 각 변수에 대하여 우선순위를 제공하는 방법은 계획기간 초에 가까운 기간에 현금흐름을 발생하는 프로젝트에 높은 우선순위를 부여하였다.

9개의 전략을 사용하여 일반 할당문제를 대상으로 실험한 결과 분지변수선택기준으로 0.3 혹은 0.7을 사용한 BP-MIN, PC-MIN, VPC-MIN 전략이 BP-MAXMIN, PC, VPC 전략보다 계산시간관점에서 효율적이었다. 따라서 〈표 4〉와 〈표 5〉에서는 SOS전략을 포함한 5개의 전략에 의한 결과를 제시하고 있다.

일반 할당문제는 다중제약식을 포함하고 있으므로 변수집합을 대상으로 분지하는 SOS전략과 SOSP전략이 다른 전략보다 효율적이다. 또한 우선순위를 제공한 SOSP전략이 SOS전략보다 효율적이었다. 문제의 크기가 증대됨에 따라 IGN 시스템이 LINDO보다 평균계산시간이 더욱 단축되었다. 특히 문제크기가 증가된 GAP 2와 GAP 2-1에서는 SOSP가 LINDO보다 평균계산시간이 3.9배 단축되었고, <표 5>에서 GAP 2-1의 10문제 중에서 가장 오래 소요된 LINDO의 계산시간(460초)보다 SOSP전략을 사용한 IGN 시스템의 계산시간(82초)은 5.

6배 단축되었다. 일반 할당문제에서 초기 정수해를 산출하기까지 선택마디의 수를 제한한 VPC MIN전략은 초기 정수실행가능해를 빨리 산출하였으나 최적해에 대한 오차율이 평균 5% - 7.2%였로 큰 편이었다. 한편 SOSP전략에 의한 초기 정수해는 40개의 일반 할당문제에서 2개 문제(평균 오차율 0.02%)를 제외하고 최적해와 동일하였다. 따라서 대규모 문제에서 SOSP전략은 초기 정수해로 문제의 근사해를 산출하는 방법으로 이용될 수 있음을 암시하고 있다.

<표 4> GAP 1과 GAP 1-1(이진변수 30개, 제약식 13개)에 대한 실험결과

문 제		SOS	BP-MIN	PC-MIN	VPC-MIN	SOSP	LINDO
GAP1	① CPU Seconds (branch의 수)	41.26 (91)	41.95 (93)	44.42 (98)	47.42 (106)	38.87 (86)	61.7
	② 최대 CPU Seconds (branch의 수)	60.42 (137)	60.38 (139)	60.91 (147)	65.42 (142)	53.66 (119)	89.0
GAP 1-1	① CPU Seconds (branch의 수)	40.81 (89)	38.20 (88)	40.57 (90)	43.80 (101)	38.21 (84)	67.10
	② CPU Seconds (branch의 수)	78.43 (180)	76.29 (192)	79.10 (188)	85.47 (212)	77.56 (193)	133.0

<표 5> GAP 2와 GAP 2-1(이진변수 4)개, 제약식 14개)에 대한 실험결과

문 제		SOS	BP-MIN	PC-MIN	VPC-MIN	SOSP	LINDO
GAP2	① CPU Seconds (branch의 수)	57.20 (85)	60.31 (90)	62.84 (89)	75.50 (110)	53.83 (79)	141.9
	② 최대 CPU Seconds (branch의 수)	94.69 (149)	100.30 (153)	113.43 (177)	120.78 (188)	88.75 (139)	192.0
GAP 2-1	① CPU Seconds (branch의 수)	70.21 (110)	70.72 (114)	72.04 (109)	80.90 (127)	54.41 (73)	212.5
	② CPU Seconds (branch의 수)	133.52 (224)	150.95 (237)	128.52 (213)	133.52 (224)	82.22 (135)	460.0

자본예산문제에 대한 계산실험에서는 BP-MAXMIN, PC, VPC전략이 BP-MIN, PC-MIN, VPC-MIN전략보다 효율적이었다. 따라서 <표 6> - <표 8>에서는 9개의 전략 중에서 PC전략을 비롯한 4개의 전략에 의한 실험결과만을 제시하고 있다. 일반할당문제에서는 마디선택기준으로 BP기준이 PC관련기준보다 효율적이었다고, 자본예산문제에서는 PC관련기준이 BP기준보다 효율적이었다. <표 6> - <표 8>에서 4개의 전략 모두가 LINDO보다 효율적이

고, 4개 전략간의 비교에서는 PCP전략이 가장 효율적이었다. PCP전략은 LINDO보다 최대 3.5배 계산시간이 단축되었다. 특히 CBP1에서 PCP전략이 LINDO보다 계산시간이 크게 단축되었는데, 이는 CBP 1은 CBP 2나 CBP 3보다 상대적으로 연속변수의 수가 많아 부문제의 해를 산출하는 데 있어서 네트워크 구조를 이용하는 IGN 시스템의 효과가 크게 나타난 결과라 볼 수 있다.

<표 6> CBP 1(이진변수 5개, 연속변수 47개, 제약식 14개)에 대한 실험결과

	BP-MAXMIN	PC	VPC	PCP	LINDO
① CPU Seconds (branch의 수)	21.74 (30)	21.33 (31)	21.90 (31)	20.05 (29)	70.4
② 최대 CPU Seconds (branch의 수)	28.67 (39)	27.30 (39)	27.35 (38)	27.13 (37)	86.0

<표 7> CBP 2(이진변수 10개, 연속변수 62개, 제약식 19개)에 대한 실험결과

	BP-MAXMIN	PC	VPC	PCP	LINDO
① CPU Seconds (branch의 수)	104.38 (104)	94.63 (93)	96.63 (94)	86.63 (83)	252.6
② 최대 CPU Seconds (branch의 수)	136.32 (133)	132.04 (122)	124.57 (118)	118.1 (118)	410.0

<표 8> CBP 3(이진변수 10개, 연속변수 58개, 제약식 21개)에 대한 실험결과

	BP-MAXMIN	PC	VPC	PCP	LINDO
① CPU Seconds (branch의 수)	106.82 (149)	98.67 (150)	115.88 (166)	94.79 (131)	287.3
② 최대 CPU Seconds (branch의 수)	193.23 (271)	176.75 (263)	186.59 (270)	162.14 (245)	536.0

일반 할당문제는 다중제약식을 포함하고 있으므로 변수집합을 대상으로 분지하는 SOS전략과 SOSP전략이 다른 전략보다 효율적이다. 또한 우선순위를 제공한 SOSP전략이 SOS전략보다 효율적이었다. 문제의 크기가 증대됨에 따라 IGN 시스템이 LINDO보다 평균계산시간이 더욱 단축되었다. 특히 문제크기가 증가된 GAP 2와 GAP 2-1에서는 SOSP가 LINDO보다 평균계산시간이 3.9배 단축되었고, <표 5>에서 GAP 2-1의 10문제 중에서 가장 오래 소요된 LINDO의 계산시간(460초)보다 SOSP전략을 사용한 IGN 시스템의 계산시간(82초)은 5.

6배 단축되었다. 일반 할당문제에서 초기 정수해를 산출하기까지 선택마디의 수를 제한한 VPC MIN전략은 초기 정수실행가능해를 빨리 산출하였으나 최적해에 대한 오차율이 평균 5% - 7.2%였로 큰 편이었다. 한편 SOSP전략에 의한 초기 정수해는 40개의 일반 할당문제에서 2개 문제(평균 오차율 0.02%)를 제외하고 최적해와 동일하였다. 따라서 대규모 문제에서 SOSP전략은 초기 정수해로 문제의 근사해를 산출하는 방법으로 이용될 수 있음을 암시하고 있다.

<표 4> GAP 1과 GAP 1-1(이진변수 30개, 제약식 13개)에 대한 실험결과

문 제		SOS	BP-MIN	PC-MIN	VPC-MIN	SOSP	LINDO
GAP1	① CPU Seconds (branch의 수)	41.26 (91)	41.95 (93)	44.42 (98)	47.42 (106)	38.87 (86)	61.7
	② 최대 CPU Seconds (branch의 수)	60.42 (137)	60.38 (139)	60.91 (147)	65.42 (142)	53.66 (119)	89.0
GAP 1-1	① CPU Seconds (branch의 수)	40.81 (89)	38.20 (88)	40.57 (90)	43.80 (101)	38.21 (84)	67.10
	② CPU Seconds (branch의 수)	78.43 (180)	76.29 (192)	79.10 (188)	85.47 (212)	77.56 (193)	133.0

<표 5> GAP 2와 GAP 2-1(이진변수 4)개, 제약식 14개)에 대한 실험결과

문 제		SOS	BP-MIN	PC-MIN	VPC-MIN	SOSP	LINDO
GAP2	① CPU Seconds (branch의 수)	57.20 (85)	60.31 (90)	62.84 (89)	75.50 (110)	53.83 (79)	141.9
	② 최대 CPU Seconds (branch의 수)	94.69 (149)	100.30 (153)	113.43 (177)	120.78 (188)	88.75 (139)	192.0
GAP 2-1	① CPU Seconds (branch의 수)	70.21 (110)	70.72 (114)	72.04 (109)	80.90 (127)	54.41 (73)	212.5
	② CPU Seconds (branch의 수)	133.52 (224)	150.95 (237)	128.52 (213)	133.52 (224)	82.22 (135)	460.0

- [5] Barzilai, J., W. D. Cook and M. Kress, "A Generalized Network Formulation of the Pairwise Comparison Consensus Ranking Model," *Management Science*, Vol. 32, No. 8(1986), pp. 1007-1014.
- [6] Beale, E. M. L. and J. J. H. Forrest, "Global Optimization Using Special Ordered Sets," *Mathematical Programming*, Vol 10(1976), pp. 52-69.
- [7] Brown, G. G. and R. D. McBride, "Solving Generalized Networks," *Management Science*, Vol. 30, No. 12(1984), pp. 1497-1523.
- [8] Crum, R. L., D. Klingman, and L. A. Tavis, "Implementation of Large-Scale Financial Planning Models : Solution Efficient Transformations," *Journal of Financial and Quantitative Analysis*, Vol. XIV, No. 1(1979), pp. 137-152.
- [9] Elam, J., F. Glover and D. Klingman, "A Strongly Convergent Primal Simplex Algorithm for Generalized Networks," *Mathematics of Operations Research*, Vol. 4, No. 1(1979), pp. 39-59.
- [10] Engquist, M. and M. D. Chang, "New Labeling Procedures for the Basis Graph in Generalized Networks," *Operations Research Letters*, Vol. 4, No. 4(1985), pp. 151-155.
- [11] Escudero, L. F., "S3 Sets, an Extension of the Beale-Tomlin Special Ordered Sets," *Mathematical Programming*, Vol. 42(1988), pp. 113-123.
- [12] Forrest, J. J., J. P. H. Hirst and J. A. Tomlin, "Practical Solution of Large Mixed Integer Programming Problems with UMPIRE," *Management Science*, Vol. 20, No. 5(1974), pp. 736-773.
- [13] Glover, F., J. Hultz, D. Klingman and J. Stutz, "Generalized Networks: A Fundamental Computer-Based Planning Tool," *Management Science*, Vol. 24, No. 12(1978), pp. 1209-1220.
- [14] Glover, F., and D. Klingman, "Layering Strategies for Creating Exploitable Structure in Linear and Integer Programs," *Mathematical Programming*, Vol 30(1988), pp. 165-181.
- [15] Glover, F., D. Klingamn, and N. V. Phillips, "A Network-Related Nuclear Power with an Intelligent Branch-and-Bound Solution Approach," *Annals of Operations Research*, Vol. 21(1989), pp. 317-332.
- [16] Glover, F., D. Klingman, and N. V. Phillips, "Netform Modeling and Applications," *Interfaces*, Vol. 20, No. 4 (1990), pp. 7-27.
- [17] Glover, F. and J. M. Mulvey, "Equivalence of the 0-1 Integer Programming Problem to Discrete Generalized and Pure Networks," *Operations Research*, Vol. 28, No. 3(1980), pp. 829-839.
- [18] Ignizio, J. P., "An Approach to the Modeling and Analysis of Multiobjective Generalized Networks," *European Journal of Operational Research*, Vol 12(1983), pp. 357-361.
- [19] Joh, I. H., "A Computational Study of the Generalized Network Algorithms,"

Ph. D. Dissertation, University of Washington, 1982.

- [20] Nemhauser, G. L. and L. A. Wolsey, *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [21] Steinberg, E., and H. A. Napier, "Optimal Multi-Level Sizing for Requirements Planning Systems," *Management Science*, Vol. 26, No. 12(1980), pp. 1258-1271.