Journal of the Operations Research
Society of Japan
Vol. 43, No. 2, June 2000

INTEGER PROGRAMMING MODEL AND EXACT SOLUTION FOR CONCENTRATOR LOCATION PROBLEM

Takayuki Shiina
Central Research Institute of Electric Power Industry

(Received September 24, 1998; Final November 1, 1999)

Abstract Topological design of centralized computer networks is an important problem that has been investigated by many researchers. Such networks typically involve a large number of terminals connected to concentrators, that are then connected to a central computing site. This paper focuses on the concentrator location problem among general topological network design problems. The concentrator location problem is defined as determining the following: (i) the number and locations of concentrators that are to be open, and (ii) the allocation of terminals to concentrator sites without violating the capacities of concentrators. An exact algorithm (fractional cutting plane algorithm/branch-and-bound) is proposed for solving this problem. In this approach an integer programming problem is formulated. Then a class of valid inequalities is derived and a greedy algorithm for a separation problem is shown. A good lower bound is obtained by a lifting procedures. We show how to implement the algorithm using a commercial software for LP and branch-and-bound. Finally, the computational efficiency of our algorithm is demonstrated.

1. Introduction

We consider topological design of centralized computer networks. Such networks typically involve a large number of terminals connected to concentrators, that are then connected to a central computing site. This paper focuses on the concentrator location problem among general topological network design problems. The concentrator location problem is defined as determining the following: (i) the number and locations of concentrators that are to be open, and (ii) the allocation of terminals to concentrator sites without violating the capacities of concentrators.

Even though the concentrator location problem is simpler than the general topological design problem (Ahuja [1], Ahuja et al. [2], Bertsekas and Gallager [5]), it is still a difficult problem to solve. Since the problem belongs to the class of NP-hard, most prior researches have developed heuristic procedures to seek approximate solutions. For the classical concentrator location problem and some of its variations (Mirzaian [9], Pirkul [14], Pirkul et al. [17], Lo and Kershenbaum [8], Pirkul and Nagarajan [16], Narashimhan and Pirkul [10], Pirkul and Gupta [15]) the Lagrangian relaxation method has been used.

In this paper we propose an exact algorithm (fractional cutting plane algorithm/branch-and-bound) for the concentrator location problem. In this approach we formulate an integer programming problem. Then we derive a class of valid inequalities and show a greedy algorithm for a separation problem. Strong valid inequalities are used as cutting planes. A good lower bound is obtained by lifting procedure. Finally, we demonstrate the computational efficiency of our algorithm. The framework of the fractional cutting plane algorithm/branch-and-bound is based on the concept of cutting plane reformulation (Van Roy and Wolsey [20], Nemhauser and Wolsey [11, 12], Wolsey [22]). These researches were motivated in part by the success of large pure 0-1 programs (Crowder et al. [6], Johnson et al. [7]).

In section 2, formulations of the problem are presented. Section 3 describes the fractional cutting plane algorithm/branch-and-bound in detail. Section 4 discusses the computational results, and conclusions are presented in Section 5.

2. Integer Programming Model

2.1. Assumptions for the model

In the local access network remote electronic devices (such as concentrator or multiplexer) enable multiple users to share the same physical line. To meet the demand for different services, traffic processors and transmission facilities have to be set between each terminal and the central computing site. Concentrators perform traffic compression, combining multiple incoming signals into a single outgoing signal and so on. The network is devided into two sections. The feeder network connects the central computing site to concentrators through a high speed line. The distribution networks connect each concentrator to its assigned terminals via low speed lines. The optimization problem is to meet the demand for the service at minimum total cost.

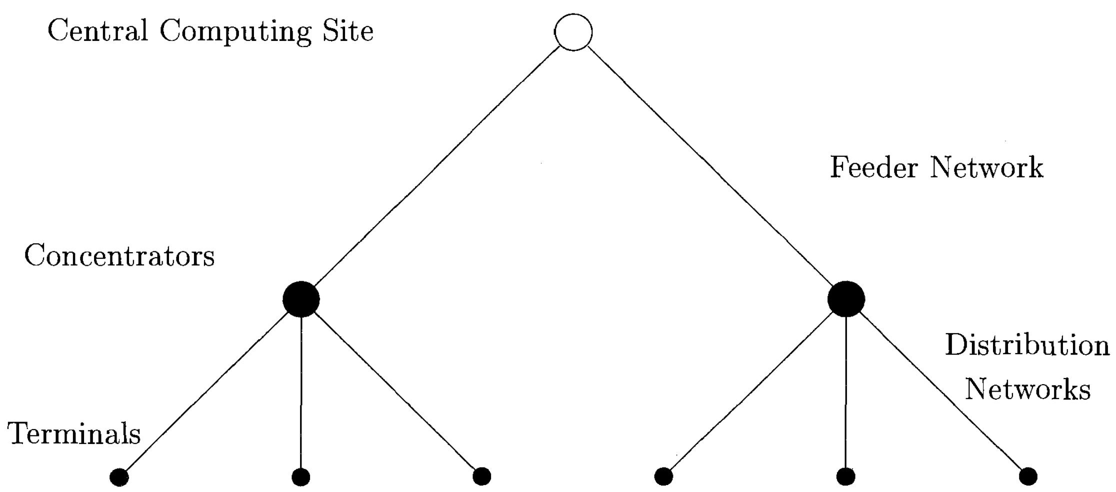

Figure 1: Centralized Computer Network

We make the following assumptions.

Assumptions:

- We consider installing new transmission facilities and traffic processors. The network does not contain any existing facilities.
- All demand can be aggregated into a same service type.
- Each terminal is directly connected to a concentrator. Traffic from each terminal must undergo one level of traffic processing. Each concentrator connects directly to the central computing site.
- The transmission and processor cost functions are separable.

From these assumptions the final network becomes to have a double star topology. The setup cost of a concentrator can include the cost of connecting a concentrator to a central site. Therefore the whole network design problem results in the design of distribution networks. The problem is where to locate new concentrators, and how to connect all terminals to the located concentrators.

2.2. Problem formulation

Table 1 describes the symbols and notations used in the paper. The mathematical formulation of the concentrator location problem is stated as follows. It is assumed that $a_m, b_{lk}, m \in M, k \in K, l \in L$ are positive integers and $c_{ml}, f_{lk} > 0, m \in M, k \in K, l \in L$.

Table 1: Notation

Symbol	Definition
M	index set of terminal locations
L	index set of potential concentrator locations
K	index set of concentrator types
	We let M, K be mutually disjoint finite sets.
c_{ml}	cost connecting terminal m and concentrator l
f_{lk}	fixed setup cost of concentrator k at site l
a_m	traffic at terminal m
b_{lk}	maximum capacity of concentrator k at site l
Decision Variables	
x_{ml}	$\begin{cases} 1 & \text{if link exists between terminal } m \text{ and concentrator } l, m \in M, l \in L \\ 0 & \text{otherwise} \end{cases}$
y_{lk}	$\begin{cases} 1 & \text{if concentrator } k \text{ is open at location } l, k \in K, l \in L \\ 0 & \text{otherwise} \end{cases}$

$$\textbf{Problem (IP}_0\textbf{)}: \min \quad \sum_{m \in M} \sum_{l \in L} c_{ml} x_{ml} + \sum_{l \in L} \sum_{k \in K} f_{lk} y_{lk} \tag{2.1}$$

$$\text{subject to} \quad \sum_{m \in M} a_m x_{ml} \leq \sum_{k \in K} b_{lk} y_{lk}, \quad l \in L \tag{2.2}$$

$$\sum_{l \in L} x_{ml} = 1, \quad m \in M \tag{2.3}$$

$$\sum_{k \in K} y_{lk} \leq 1, \quad l \in L \tag{2.4}$$

$$x_{ml}, y_{lk} \in \{0, 1\}, \ m \in M, l \in L, k \in K \tag{2.5}$$

The objective function (2.1) minimizes the cost of connecting terminals to concentrators and the cost of opening concentrators. Constraint (2.2) represents capacities of concentrators. Constraint (2.3) ensures that each terminal is assigned to a single concentrator. Constraint (2.4) assures that at most one concentrator out of $|K|$ types can be open at each site l.

Now we introduce variables $y_{lk} = 1 - z_{lk}, l \in L, k \in K$. This transformation yields the next formulation **(IP)**.

$$\textbf{Problem (IP)}: \min \quad \sum_{m \in M} \sum_{l \in L} c_{ml} x_{ml} + \sum_{l \in L} \sum_{k \in K} f_{lk} (1 - z_{lk}) \tag{2.6}$$

$$\text{subject to} \quad \sum_{m \in M} a_m x_{ml} + \sum_{k \in K} b_{lk} z_{lk} \leq \sum_{k \in K} b_{lk}, \quad l \in L \tag{2.7}$$

$$\sum_{l \in L} x_{ml} = 1, \quad m \in M \tag{2.8}$$

$$\sum_{k \in K} z_{lk} \geq |K| - 1, \quad l \in L \tag{2.9}$$

$$x_{ml}, z_{lk} \in \{0, 1\}, \ m \in M, l \in L, k \in K \tag{2.10}$$

For the generalized assignment problem Ross and Soland [18] presented a branch-and-bound algorithm. We see that this formulation **(IP)** contains knapsack constraints (2.7). The family of valid inequalities to define a facet of a knapsack polytope was obtained by Balas [3] and Balas and Zemel [4].

3. Fractional Cutting Plane Algorithm/Branch-and-Bound

3.1. Automatic reformulation approach

In this section we briefly outline the automatic reformulation approach (Van Roy and Wolsey [20], Nemhauser and Wolsey [11, 12], Wolsey [22]). The fractional cutting plane algorithm/branch-and-bound (FCPA/B&B) utilized automatic reformulation approach. It is executed in two phases. FCPA/B&B was applied to the maintenance scheduling of power plants by Shiina and Kubo [19]. The first phase, called *cut generation*, is typically a fractional cutting plane algorithm, and is described more formally below.

We start with an integer programming problem **(IP)**. By relaxing the integer constraints of **(IP)**, the relaxed feasible solution set F^0 is described as follows.

$$\begin{aligned} F^0 = \{(\boldsymbol{x},\boldsymbol{z})| \; & \sum_{m\in M} a_m x_{ml} + \sum_{k\in K} b_{lk} z_{lk} \le \sum_{k\in K} b_{lk}, \quad l\in L \\ & \sum_{l\in L} x_{ml} = 1, \quad m \in M \\ & \sum_{k\in K} z_{lk} \ge |K|-1, \quad l\in L \\ & 0 \le x_{ml}, z_{lk} \le 1, \; m\in M, l\in L, k\in K \quad \}, \end{aligned} \tag{3.1}$$

where x_{ml} and z_{lk} denote ml and lk component of $|M||L|$-vector $\boldsymbol{x}$ and $|L||K|$-vector $\boldsymbol{z}$, respectively. At iteration t the basic iterative step of the automatic reformulation procedure deals with F^t, then it attempts to cut off some region from F^t and to obtain $F^{t+1} \subset F^t$, where F^{t+1} contains all feasible solutions of **(IP)**. Given the formulation F^t, the procedure solves the linear programming problem

$$\textbf{Problem (LP)} : \min\{ \sum_{m\in M}\sum_{l\in L} c_{ml}x_{ml} + \sum_{l\in L}\sum_{k\in K} f_{lk}(1-z_{lk}) | (\boldsymbol{x},\boldsymbol{z}) \in F^t\} \tag{3.2}$$

and finds an optimal solution $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ of (3.2). If $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ is integer, $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ solves **(IP)**. If not, the procedure produces a set of valid inequalities $\boldsymbol{\pi}_x^k \boldsymbol{x} + \boldsymbol{\pi}_z^k \boldsymbol{z} \le \pi_0^k$ for $k \in \mathcal{F}^t$ cutting off $(\boldsymbol{x}^t, \boldsymbol{z}^t)$, or fails to find a valid inequality violated by $(\boldsymbol{x}^t, \boldsymbol{z}^t)$. $\mathcal{F}^t$ is an index set of valid inequalities generated in iteration t and $\boldsymbol{\pi}_x^k, \boldsymbol{\pi}_z^k$ and π_0^k are a constant $|M||L|$-vector, a constant $|L||K|$-vector and constant, respectively.

In the first case, $F^{t+1} = F^t \cap \{(\boldsymbol{x},\boldsymbol{z}) | \boldsymbol{\pi}_x^k \boldsymbol{x} + \boldsymbol{\pi}_z^k \boldsymbol{z} \le \pi_0^k, k\in \mathcal{F}^t\}$ is the new formulation, and t is increased by 1. In the second case, the reformulation is terminated with the formulation of F^t, and the procedure solves integer programming problem **(IP')**

$$\textbf{Problem (IP')} : \min\{ \sum_{m\in M}\sum_{l\in L} c_{ml}x_{ml} + \sum_{l\in L}\sum_{k\in K} f_{lk}(1-z_{lk}) | (\boldsymbol{x},\boldsymbol{z}) \in F^t, (\boldsymbol{x},\boldsymbol{z}) \text{is a 0-1 vector}\} \tag{3.3}$$

by using a branch-and-bound algorithm.

- **Phase 1. Fractional Cutting Plane Algorithm (FCPA)**
  - **Step 0. Initialization**
    Set $t = 0$.
  - **Step 1. Solve LP**
    Solve $\min\{\sum_{m\in M}\sum_{l\in L} c_{ml}x_{ml} + \sum_{l\in L}\sum_{k\in K} f_{lk}(1 - z_{lk}) | (\boldsymbol{x}, \boldsymbol{z}) \in F^t\}$ and let $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ be an optimal solution of the linear program.
  - **Step 2. Optimality Test**
    If $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ is integer, $(\boldsymbol{x}^t, \boldsymbol{z}^t)$ solves **(IP)**. Otherwise go to **Step 3**.
  - **Step 3. Find Valid Inequalities**
    Find valid inequalities such as $\boldsymbol{\pi}_x^k \boldsymbol{x} + \boldsymbol{\pi}_z^k \boldsymbol{z} \le \pi_0^k$ for $k \in \mathcal{F}^t$ cutting off $(\boldsymbol{x}^t, \boldsymbol{z}^t)$.
  - **Step 4. Refinement**
    If $\mathcal{F}^t = \phi$, go to **Phase 2**. Otherwise set $F^{t+1} = F^t \cap \{(\boldsymbol{x}, \boldsymbol{z}) | \boldsymbol{\pi}_x^k \boldsymbol{x} + \boldsymbol{\pi}_z^k \boldsymbol{z} \le \pi_0^k, k \in \mathcal{F}^t\}$, and $t = t + 1$, then go to **Step 1**.
- **Phase 2. Branch-and-Bound (B&B)**
  Solve **Problem (IP')** : $\min\{\sum_{m\in M}\sum_{l\in L} c_{ml}x_{ml} + \sum_{l\in L}\sum_{k\in K} f_{lk}(1 - z_{lk}) | (\boldsymbol{x}, \boldsymbol{z}) \in F^t, (\boldsymbol{x}, \boldsymbol{z})$ is a 0-1 vector$\}$ by using a branch-and-bound algorithm.

3.2. Valid inequality and separation problem

In FCPA we derive a class of valid inequalities that is a variant of a knapsack cover inequality (Balas [3], Balas and Zemel [4]). We consider the feasible solution set of **(IP)**.

$$\text{Feasible solution set of } \textbf{(IP)} = \{(\boldsymbol{x}, \boldsymbol{z}) | \begin{array}{l} \sum_{m\in M} a_m x_{ml} + \sum_{k\in K} b_{lk} z_{lk} \le \sum_{k\in K} b_{lk}, \quad l \in L \\ \sum_{l\in L} x_{ml} = 1, \quad m \in M \\ \sum_{k\in K} z_{lk} \ge |K| - 1, \quad l \in L \\ x_{ml}, z_{lk} \in \{0,1\}, \ m \in M, l \in L, k \in K\} \end{array} \quad (3.4)$$

Definition 1 *Let C_M, C_K be the subset of M, K, respectively. If $\sum_{m\in C_M} a_m + \sum_{k\in C_K} b_{lk} > \sum_{k\in K} b_{lk}$, then $C_M \cup C_K$ is called a cover.*

Theorem 1 (Balas [3]) *If $C_M \cup C_K$ is a cover, then $\sum_{m\in C_M} x_{ml} + \sum_{k\in C_K} z_{lk} \le |C_M \cup C_K| - 1$ is a valid inequality for the feasible solution set of* **(IP)**.

In order to be able to apply FCPA/B&B, we formalize the separation problem. Given a point $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{z}})$ that is not an integer solution of (LP), we wish to find C_M and C_K with $\sum_{m\in C_M} \tilde{x}_{ml} + \sum_{k\in C_K} \tilde{z}_{lk} > |C_M \cup C_K| - 1$ and $\sum_{m\in C_M} a_m + \sum_{k\in C_K} b_{lk} > \sum_{k\in K} b_{lk}$. Let variables $v_m, m \in M, w_k, k \in K$ represent the set C_M, C_K, respectively.

$$v_m = \begin{cases} 1, & \text{if } m \in C_M \\ 0, & \text{otherwise} \end{cases} \qquad w_k = \begin{cases} 1, & \text{if } k \in C_K \\ 0, & \text{otherwise} \end{cases}$$

Thus we can formulate the separation problem at site l as follows;

$$\textbf{(Separation Problem at Site } l\textbf{)} \left| \begin{array}{ll} \zeta_l = \min & \sum_{m\in M} (1 - \tilde{x}_{ml}) v_m + \sum_{k\in K} (1 - \tilde{z}_{lk}) w_k \\ \text{subject to} & \sum_{m\in M} a_m v_m + \sum_{k\in K} b_{lk} w_k > \sum_{k\in K} b_{lk} \\ & v_m, w_k \in \{0,1\}, \ m \in M, k \in K \end{array} \right. .$$

In one iteration of FCPA, the separation problem is solved $|L|$ times. If $\zeta_l < 1$ for some $l \in L$, then $\sum_{m\in C_M} x_{ml} + \sum_{k\in C_K} z_{lk} \le |C_M \cup C_K| - 1$ is one of the most violated valid inequality. Since the separation problem is a difficult problem, we adopt a heuristic approach.

algorithm GREEDY:
  begin
    $\zeta_l := 0$;
    $capacity := 0$;
    $C_M, C_K := \phi$;
    while $capacity \le \sum_{k\in K} b_{lk}$ **and** $(M \setminus C_M) \cup (K \setminus C_K) \neq \phi$ **do**
      begin
        if $M \setminus C_M = \phi$ **then** $val_M := \infty$;
        else $m^* := \underset{m\in M\setminus C_M}{\text{argmin}} (\frac{1-\tilde{x}_{ml}}{a_m}), val_M := \frac{1-\tilde{x}_{m^*l}}{a_{m^*}}$;
        if $K \setminus C_K = \phi$ **then** $val_K := \infty$;
        else $k^* := \underset{k\in K\setminus C_K}{\text{argmin}} (\frac{1-\tilde{z}_{lk}}{b_{lk}}), val_K := \frac{1-\tilde{z}_{lk^*}}{b_{lk^*}}$;
          if $val_M \le val_K$ **then**
            begin
            $C_M := C_M \cup \{m^*\}$;
            $capacity := capacity + a_{m^*}$;
            $\zeta_l := \zeta_l + (1 - \tilde{x}_{m^*l})$;
            end
          else
            begin
            $C_K := C_K \cup \{k^*\}$;
            $capacity := capacity + b_{lk^*}$;
            $\zeta_l := \zeta_l + (1 - \tilde{z}_{lk^*})$;
            end
      end
  end

As the algorithm GREEDY requires sorting of $|M \cup K|$ elements, it runs in $O(|M \cup K| \log |M \cup K|)$ time.

3.3. Lifting procedure

To strengthen the valid inequality (Theorem 1) of the site $l \in L$, we adopt a lifting procedure. Let $\bar{C} = (M \setminus C_M) \cup (K \setminus C_K)$, $\bar{M} = M \setminus C_M$, $\bar{K} = K \setminus C_K$, $r = |C_M \cup C_K|$. Given a cover $C_M \cup C_K$ such that $\sum_{m\in C_M} a_m + \sum_{k\in C_K} b_{lk} > \sum_{k\in K} b_{lk}$, it is defined that $\{c_{j_1}, c_{j_2}, \dots, c_{j_r}\} = \{a_m, m \in C_M\} \cup \{b_{lk}, k \in C_K\}$ and assumed that coefficients are monotonically ordered so that $c_{j_1} \ge c_{j_2} \dots \ge c_{j_r}$. Let $\mu_0 = 0$, $\mu_h = \sum_{i=1}^{h} c_{j_i}$ for $h = 1, \dots, r$, $\mu_h = \mu_r$ for $h > r$, and $\lambda = \sum_{m\in C_M} a_m + \sum_{k\in C_K} b_{lk} - \sum_{k\in K} b_{lk} > 0$. For $j \in \bar{C}$, let $\beta_j = h$ if $(\mu_h \le a_j \le \mu_{h+1} - 1)$ or $(\mu_h \le b_{lj} \le \mu_{h+1} - 1)$. For $Q \subseteq \bar{C}$ define $\beta(Q) = \sum_{j\in Q}(\beta_j + 1)$.

A lifted valid inequality is the inequality of the form (3.5).

$$\sum_{m\in C_M} x_{ml} + \sum_{k\in C_K} z_{lk} + \sum_{j\in \bar{M}} \alpha_j x_{jl} + \sum_{j\in \bar{K}} \alpha_j z_{lj} \le |C_M \cup C_K| - 1 \tag{3.5}$$

The lifting coefficients α_j are nonnegative integers. Balas and Zemel[4] showed that $\alpha_j \le \beta_j + 1$ for all $j \in \bar{C}$ is a necessary condition for (3.5) to be valid, and showed that for at least one such j, it is valid to set $\alpha_j = \beta_j + 1$.

Nemhauser and Vance [13] extended the result of Balas and Zemel by giving an inequality that specifies exactly which subsets of the undetermined lifting coefficients can be set to higher values simultaneously. Nemhauser and Vance called a set $S \subseteq \bar{C}$ an *independent* set if for all nonempty $Q \subseteq S$,

$$\sum_{j\in Q\cap\bar{M}} a_j + \sum_{j\in Q\cap\bar{K}} b_{lj} > \mu_{\beta(Q)} - \lambda \tag{3.6}$$

holds.

Next, it will be shown that all the members of S may have $\alpha_j = \beta_j + 1$. Define $G(v)$ as follows, and let $b = \sum_{k\in K} b_{lk}$.

$$\begin{aligned} G(v) = \max \quad & \sum_{m\in C_M} x_{ml} + \sum_{k\in C_K} z_{lk} && (3.7)\\ \text{subject to} \quad & \sum_{m\in C_M} a_m x_{ml} + \sum_{l\in C_K} b_{lk} z_{lk} \le v && (3.8)\\ & x_{ml} \in \{0,1\}, m \in C_M && (3.9)\\ & z_{lk} \in \{0,1\}, k \in C_K && (3.10) \end{aligned}$$

From the relation $b - \mu_{\beta(Q)} + \lambda = \mu_r - \mu_{\beta(Q)}$, we have $G(b - \mu_{\beta(Q)} + \lambda) = r - \beta(Q)$. This yields $G(b - \mu_{\beta(Q)} + \lambda - 1) \le r - \beta(Q) - 1$.

For the lifted valid inequality (3.5) to be valid, the inequality

$$\sum_{j\in Q\cap\bar{M}} \alpha_j + \sum_{j\in Q\cap\bar{K}} \alpha_j + G(b - \sum_{j\in Q\cap\bar{M}} a_i - \sum_{j\in Q\cap\bar{K}} b_{lj}) \le r - 1 \tag{3.11}$$

must hold.

Suppose $S \subseteq \bar{C}$ is independent, then from the inequality (3.6)

$$b - \sum_{j\in Q\cap\bar{M}} a_j - \sum_{j\in Q\cap\bar{K}} b_{lj} < b - \mu_{\beta(Q)} + \lambda.$$

Therefore

$$\begin{aligned} G(b - \sum_{j\in Q\cap\bar{M}} a_j - \sum_{j\in Q\cap\bar{K}} b_{lj}) &\le G(b - \mu_{\beta(Q)} + \lambda - 1)\\ &\le r - \beta(Q) - 1. \end{aligned}$$

Now if we set $\sum_{j\in Q\cap\bar{M}} \alpha_j + \sum_{j\in Q\cap\bar{K}} \alpha_j = \beta(Q)$, then

$$\begin{aligned} \sum_{j\in Q\cap\bar{M}} \alpha_j + \sum_{j\in Q\cap\bar{K}} \alpha_j + G(b - \sum_{j\in Q\cap\bar{M}} a_j - \sum_{j\in Q\cap\bar{K}} b_{lj}) &\le \beta(Q) + G(b - \mu_{\beta(Q)} + \lambda - 1)\\ &\le r - 1. \end{aligned}$$

This inequality assures that if we set $\alpha_j = \beta_j + 1, j \in S$, the lifted inequality (3.5) remains a valid inequality. Identifying the set S with maximal violation of (3.5) by $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{z}})$ is a difficult problem. In the heuristic procedure LIFTING, we check whether the element of $\bar{C}$ can be contained in S in decreasing order of the element of $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{z}})$.

```
procedure LIFTING:
  begin
    S := φ;
    while{M̄ ∪ K̄} \ S ≠ φ; do
      begin
        if M \ C_M = φ then val_x := −∞;
        else m* := argmax_{m∈M\C_M} (x̃_{ml}), val_x := x̃_{m*l};
        if K \ C_K = φ then val_z := −∞;
        else k* := argmax_{k∈K\C_K} (z̃_{lk}), val_z := z̃_{lk*};
        if val_x ≥ val_z then n* := m*;
        else n* := k*;
        if Σ_{j∈Q} a_j + Σ_{j∈Q} b_{lj} > μ_{β(Q)} − λ, ∀Q ⊆ S ∪ {n*};
          begin
            S := S ∪ {n*};
            α_{n*} := β_{n*} + 1;
          end
        else
          begin
            α_{n*} := β_{n*};
            {M̄ ∪ K̄} := {M̄ ∪ K̄} \ {n*};
          end
      end
  end
```

Example 1

We consider the case for $M = \{6, 7, 9, 16, 23, 26, 29, 31, 34, 35, 36, 42, 45, 47, 51, 53, 55, 57\}$, $l = 1$, $K = \{1\}$. Let $b_{1,1} = 2580$, and $a_m, m \in M$ as follows.

$$(a_6, a_7, a_9, a_{16}, a_{23}, a_{26}, a_{29}, a_{31}, a_{34}, a_{35}, a_{36}, a_{42}, a_{45}, a_{47}, a_{51}, a_{53}, a_{55}, a_{57}) = (203,294,271,175,209,82,57,313,261,312,319,139,321,71,131,194,56,286)$$

The valid inequality is obtained with $C_M = \{6, 9, 23, 26, 29, 31, 34, 35, 36, 42, 47, 53, 57\}$, $C_K = \phi$. The **procedure** LIFTING yields the next inequality with $\bar{C} = \{1, 7, 16, 45, 51, 55\}$.

$$\sum_{m \in C_M} x_{m,1} + 11 z_{1,1} + x_{7,1} + x_{45,1} \leq 13 - 1$$

We have $(\mu_1, \ldots, \mu_{13})$ =(319, 632, 944, 1230, 1501, 1762, 1971, 2174, 2368, 2507, 2589, 2660, 2717) and $S = \{1, 7\}$.

4. Numerical Experiments

We utilized FCPA/B&B to solve the above problem. The whole framework of FCPA/B&B (Figure 2) including the algorithm GREEDY and the procedure LIFTING was coded in Perl (Wall and Schwartz [21]) and a commercial software XPRESS-MP [23] was used as a linear programming/branch-and-bound solver. Computational experiments were carried out on a SPARC Station 2.

XPRESS-MP consists of a model builder and an optimizer. The model builder interprets the symbolic model specification statements and generates the input data in XMPS format.

The optimizer seeks an optimal solution by the revised simplex method. In searching a tree of branch-and-bound each active node has an LP relaxation value and an estimated degradation to a feasible integer solution. We adopt the two descendant nodes as the candidate set for the node selection. If both bave been fathomed, then all active nodes form the set and the node with the best estimate is chosen. In a variable selection for branching we take the one with the highest estimated degradation.

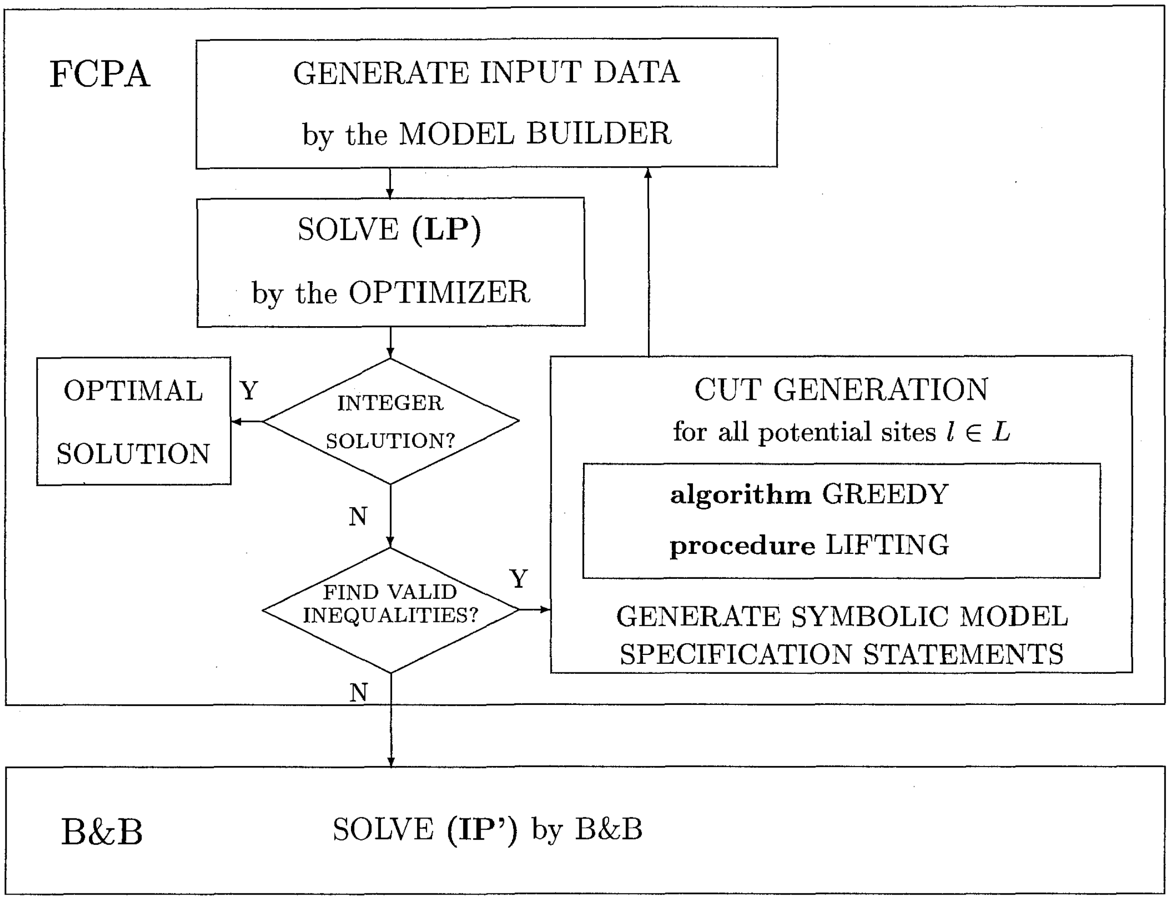

Figure 2: FCPA/B&B using Commercial Software for LP and B&B

Example 2

In the model specification statements of XPRESS-MP the valid inequality of Example 1 is described as follows. This statement is added to the model file of XPRESS-MP. Then the model builder can interpret the model file and generate the input data in XMPS format.

The Lifted Valid Inequality:

```
x(6,1)+x(9,1)+x(23,1)+x(26,1)+x(29,1)+x(31,1)+x(34,1)+x(35,1)+x(36,1)
+x(42,1)+x(47,1)+x(53,1)+x(57,1)+11*z(1,1)+1*x(7,1)+1*x(45,1) < 13-1
```

The coordinates for the sites of terminals and concentrator locations were generated from a uniform distribution over a rectangle of $[0,100] \times [0,100]$, $[10,90] \times [10,90]$, respectively. In these experiments the number of concentrators at every site was set as $|K| = 1$, and the number of concentrators that each terminal could connect to was restricted to 5. We let $l_i(m)$ be the i-th nearest potential concentrator location to terminal $m \in M$. For $i = 1, \ldots, 5$, the Euclidean distance between terminal m and potential concentrator location $l_i(m)$ was used to define the cost coefficient $c_{ml_i(m)}$ as:

$$c_{ml_i(m)} = \lfloor(\text{distance between } m \text{ and } l_i(m)) \times 0.2\rfloor + 1, \tag{4.1}$$

otherwise we set $c_{ml_i(m)} = \infty$. The traffic data from every terminal were defined as:

$$a_m = \lfloor U[0,300]\rfloor + 50, \tag{4.2}$$

where $U[0,300]$ was a number drawn from a uniform distribution between 0 and 300. The capacity b_{lk} and the fixed cost f_{lk} of the concentrator were defined as:

$$b_{lk} = \lfloor |M| \times U[0,30]\rfloor + 20, f_{lk} = \lfloor\sqrt{b_{lk}}\rfloor, \tag{4.3}$$

where $U[0,30]$ was a number drawn from a uniform distribution between 0 and 30.

The problems considered in this paper consist of 60–100 terminal sites and 30–40 potential concentrator locations. Since the problem (IP) and (IP') have special ordered sets (SOS) constraints of the form $\sum_{l\in L} x_{ml} = 1, m \in M$, we adopt the branching scheme that takes SOS into account.

The results show that FCPA/B&B performs reasonably well on relatively large problems. The computing time includes the time for problem generation in XPRESS-MP. The number of branchings and the computing time tend to rise as the size of the problem increases. It is observed that in all cases the number of branchings and the computing time of FCPA/B&B are less than those of the usual B&B. Especially in the case with 40 potential concentrator locations and 100 terminals the computing time of FCPA/B&B is nearly $17.9 \approx \frac{50343}{2811}$ times faster than that of B&B.

Table 2: Computational Results

$\|M\|$	$\|L\|$	Lower Bound		Optimal Value	Number of Branchings		Computing Time (Sec)	
		LP	FCPA	IP	B&B	FCPA/B&B	B&B	FCPA/B&B
60	30	423.6	494.9	533	21184	579	488.3	287.8
80	30	521.3	635.1	664	45264	799	1685.1	841.2
100	30	620.5	686.7	749	103728	2267	4574.7	1207.3
60	40	429.5	515.3	534	76922	1335	2096.1	518.1
80	40	552.7	650.1	708	156163	3438	6021.4	824.7
100	40	666.9	765.3	870	993365	5643	50343	2811.6

Though the computing time for our results may seem large, it should be noted that our algorithm yields an exact solution. Further, these networks would represent a large computer network in which each one of these terminals is likely to be a cluster of connected smaller computer systems. It is thus obvious that we can treat larger network problems as well. Figure 3 illustrates the optimal concentrator location with 30 potential concentrator sites and 80 terminals. In Figure 3 (□) represents terminals and (∗) potential concentrator locations. This shows the number of concentrators to be open is 9, and the other potential 21 sites concentrators are not open.

In Figure 4 the number of concentrators to be open is 10. The lower bound for the optimal objective value and the number of added valid inequalities are shown in Table 3 and 4.

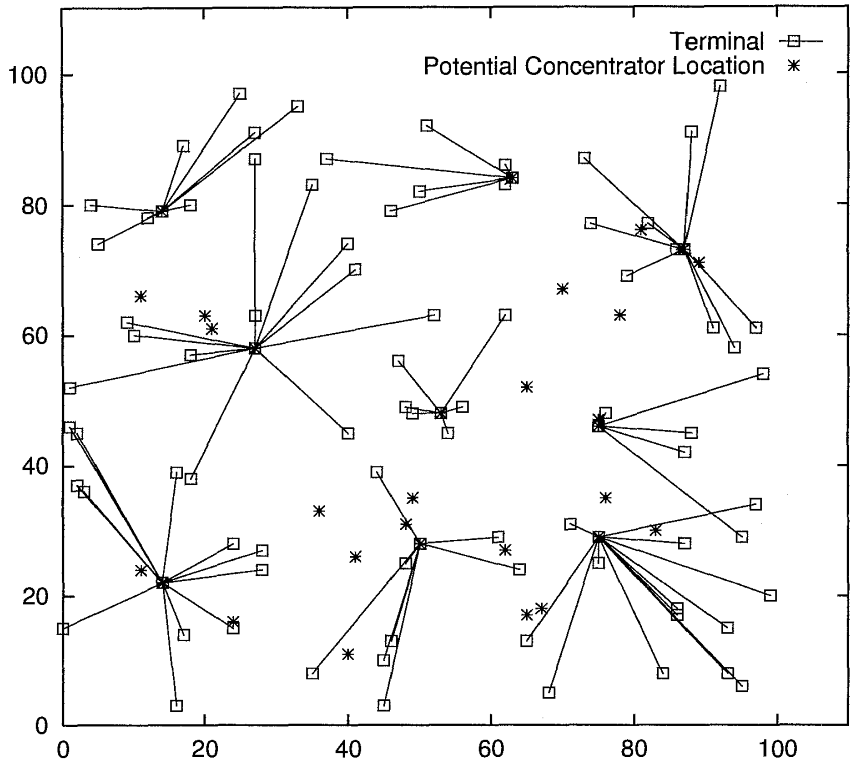

Figure 3: Optimal Network (30 potential concentrator sites, 80 terminals)

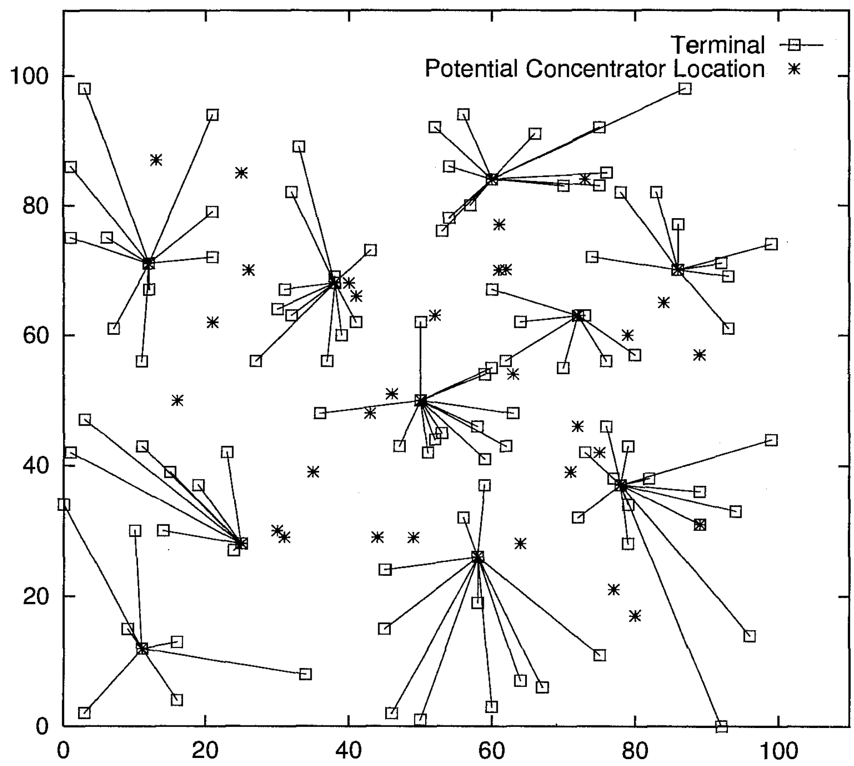

Figure 4: Optimal Network (40 potential concentrator sites, 100 terminals)

Table 3: Results of FCPA($|L| = 30$)

iteration	$\|M\|$ =60		$\|M\|$ =80		$\|M\|$ =100	
	Lower Bound	Added Cuts	Lower Bound	Added Cuts	Lower Bound	Added Cuts
1	423.6	19	521.3	23	620.5	24
2	429.8	18	527.5	22	629.8	20
3	435.4	16	531.4	21	632.4	19
4	441.8	17	535.7	20	634.8	17
5	444.2	15	536.7	19	637.2	16
6	450.3	13	539.1	21	639.1	16
7	456.1	14	541.6	16	641.9	13
8	457.7	10	542.4	16	643.2	12
9	460.2	8	544.0	20	646.0	15
10	462.0	10	545.8	18	651.7	15
11	462.7	8	548.9	17	657.9	15
12	468.3	9	550.9	18	660.3	11
13	470.9	8	556.8	18	663.7	12
14	481.4	7	561.0	19	666.2	13
15	483.8	6	567.6	17	667.3	9
16	485.1	6	579.6	18	669.6	11
17	491.5	6	585.4	14	670.5	14
18	493.2	4	591.4	14	671.5	7
19	494.6	2	593.8	13	672.8	7
20	494.8	1	596.1	8	674.2	8
21	494.8	1	599.8	8	674.9	7
22	494.9	1	601.1	6	675.6	8
23	494.9	0	602.5	7	676.7	10
24			608.5	6	679.4	12
25			611.4	11	680.5	7
26			614.5	11	681.1	6
27			619.1	6	681.8	4
28			620.5	6	682.0	3
29			620.9	5	682.1	3
30			622.3	5	682.1	2
31			623.9	7	682.1	2
32			625.7	4	682.1	1
33			628.3	4	682.1	1
34			628.9	2	682.1	1
35			629.5	2	684.0	2
36			630.0	3	684.6	2
37			630.2	2	684.9	2
38			630.7	3	685.4	4
39			631.1	2	685.8	2
40			631.3	3	686.7	2
41			632.2	4	686.7	1
42			633.1	3	686.7	0
43			633.3	2		
44			633.4	2		
45			633.4	2		
46			633.8	2		
47			634.5	2		
48			634.6	2		
49			635.1	0		

Table 4: Results of FCPA($|L| = 40$)

iteration	$\|M\|$ =60		$\|M\|$ =80		$\|M\|$ =100	
	Lower Bound	Added Cuts	Lower Bound	Added Cuts	Lower Bound	Added Cuts
1	429.5	24	552.7	25	666.9	29
2	437.7	23	563.0	21	675.8	28
3	443.7	22	566.8	19	680.4	28
4	449.3	19	573.7	18	686.0	22
5	454.7	18	579.6	16	688.3	23
6	463.1	16	583.9	12	690.2	22
7	464.7	15	585.6	13	695.2	23
8	476.7	12	588.6	12	698.4	22
9	490.1	10	590.3	13	701.2	18
10	492.3	9	603.0	12	702.2	15
11	494.0	8	604.5	8	708.1	19
12	500.3	9	605.6	8	710.8	18
13	502.2	8	608.3	9	712.9	18
14	502.6	7	611.4	8	714.5	16
15	503.3	9	614.9	6	716.5	17
16	504.5	6	620.8	11	718.1	16
17	505.6	7	625.5	6	720.0	12
18	506.1	6	628.0	10	721.0	13
19	506.4	5	631.9	6	721.3	10
20	506.5	4	635.9	9	722.3	11
21	507.4	2	637.0	4	724.2	8
22	507.5	3	638.3	3	724.8	6
23	507.8	2	639.0	6	725.2	6
24	508.5	5	642.0	5	725.8	7
25	509.8	2	643.3	6	726.4	9
26	510.4	4	645.6	8	728.1	10
27	511.8	3	647.2	5	729.5	11
28	512.6	2	647.6	3	731.7	8
29	514.1	4	648.3	4	733.8	9
30	514.7	2	648.7	2	736.8	8
31	515.1	1	649.5	7	737.9	8
32	515.3	1	649.6	1	739.1	5
33	515.3	0	649.7	1	739.4	3
34			649.7	2	743.6	8
35			649.9	2	746.8	6
36			650.0	3	750.1	8
37			650.0	4	756.1	8
38			650.1	1	758.0	7
39			650.1	0	759.8	6
40					761.4	3
41					762.0	2
42					762.1	3
43					762.5	1
44					762.8	2
45					762.8	1
46					763.2	2
47					763.5	1
48					763.9	1
49					764.3	5
50					764.6	3
51					764.9	2
52					765.3	1
53					765.3	0

5. Concluding Remarks

We have proposed an exact algorithm (fractional cutting plane algorithm/branch-and-bound (FCPA/B&B)) for the concentrator location problem. In this approach an integer programming problem is formulated. Then by deriving a class of valid inequalities a greedy algorithm for a separation problem is utilized. A good lower bound is obtained by the lifting procedure. The computational results show that FCPA/B&B performs reasonably well on relatively large problems up to 40 potential sites and 100 terminals.

The following points are left as future problems. For many actual problems, the assumption that the traffic demands at each terminal are deterministic known data is often unjustified. These data contain uncertainty and are thus represented as random variables since the data represent information about the future. Locating too few concentrators may result in shortage of capacity for the future demand. On the other hand excessive investments will cause excess of capacity. Our problem becomes thus a strategic decision problem under uncertainty and can be viewed as a stochastic programming problem. This will be useful in considering these problems.

References

[1] V. Ahuja: *Design and Analysis of Computer Communication Networks* (McGraw-Hill, 1982).

[2] R. K. Ahuja, T. L. Magnanti and J. B. Orlin: *Network Flows* (Prentice-Hall, 1993).

[3] E. Balas: Facet of the knapsack inequality. *Mathematical Programming*, **8**(1975) 146–164.

[4] E. Balas and E. Zemel: Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, **34**(1978) 119–148.

[5] D. Bertsekas and R. Gallager: *Data Networks* (Prentice-Hall, 1987).

[6] H. Crowder, E. L. Johnson and M. Padberg: Solving large scale zero-one linear programming problems. *Operations Research*, **31**(1983) 803–834.

[7] E. L. Johnson, M. M. Kostreva and H. H. Suhl: Solving 0-1 integer programming problems arising from large scale planning models. *Operations Research*, **33**(1985) 803–819.

[8] C. Lo and A. Kershenbaum: A two-phased algorithm and performance bounds for the concentrator location problem. *IEEE Transactions on Communications*, **37**(1989) 1151–1163.

[9] A. Mirzaian: Lagrangian relaxation for star-star concentrator location problem: approximation algorithm and bounds. *Networks*, **15**(1985) 1–20.

[10] S. Narashimhan and H. Pirkul: Hierachical concentrator location problem. *Computer Communications*, **15**(1992) 185–191.

[11] G. L. Nemhauser and L. A. Wolsey: *Integer and Combinatorial Optimization* (John Wiley & Sons, 1988).

[12] G. L. Nemhauser and L. A. Wolsey: Integer Programming. in G. L. Nemhauser, A. H. G. Rinnooy Kan and M. J. Todd (eds.): *Optimization* (Elsevier Science Publishers B. V., 1989).

[13] G. L. Nemhauser and P. H. Vance: Lifted cover facets of the 0-1 knapsack polytope with GUB constraints. *Operations Research Letters*, **16**(1994) 255–263.

[14] H. Pirkul: Efficient algorithms for the capacitated concentrator location problem. *Comput. Operations Research*, **14**(1987) 197–208.

[15] H. Pirkul and R. Gupta: Topological design of centralized computer networks. *International Transactions in Operational Research*, **4**(1997) 75–83.

[16] H. Pirkul and V. Nagarajan: Locating concentrators in centralized computer networks. *Annals of Operations Research*, **36**(1992) 247–261.

[17] H. Pirkul, S. Narashimhan and P. De: Locating concentrators for primary and secondary coverage in a computer communications network. *IEEE Transactions on Communications*, **36**(1988) 450–458.

[18] G. T. Ross and R. M. Soland: A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, **8**(1975) 91–103.

[19] T. Shiina and M. Kubo: Fractional cutting plane algorithm/branch-and-bound for maintenance scheduling of power plants (in Japanese). *Transactions of the Japan Society for Industrial and Applied Mathematics*, **8**(1998) 157–168.

[20] T. J. Van Roy and L. A. Wolsey: Solving mixed integer programming problems using automatic reformulation. *Operations Research*, **35**(1987) 45–57.

[21] L. Wall and R. L. Schwartz: *Programming Perl* (O'Reilly and Associates, Inc., 1990).

[22] L. A. Wolsey: Strong formulations for mixed integer programming: A survey. *Mathematical Programming*, **45**(1989) 173–191.

[23] *XPRESS-MP REFERENCE MANUAL Release 8* (Dash Associates, 1984).

Takayuki Shiina
Communication & Information Research Laboratory
Central Research Institute of Electric Power Industry
2-11-1, Iwado-Kita, Komae-Shi, Tokyo 201-8511, Japan
E-mail: `shiina@criepi.denken.or.jp`