Lagrangian Relaxation for the Star-Star Concentrator Location Problem: Approximation Algorithm and Bounds

Andranik Mirzaian
Computer Science Department, York University, Downsview, Ontario, Canada M3J 1P3

The star-star concentrator location problem (SSCLP), which is a network layout problem, is considered. SSCLP is formulated as an integer linear programming problem. The Lagrangian relaxation (LR) method is used to obtain suboptimal solutions (upper bounds) and lower bounds. Three different LRs are used for SSCLP. The resulting Lagrangian dual problems are shown to be equivalent to some linear programming problems. An approximation algorithm is suggested for SSCLP that produces both a feasible solution (upper bound) and a lower bound. It is shown that if $\underline{z}$ and $\bar{z}$ are the lower and upper bounds found, then $\bar{z}/\underline{z} \leqslant k$, where k is the concentrator capacity. Some computational examples with up to 50 terminals and 20 potential concentrator sites are considered. All the network designs obtained are shown to be within 2.8% of optimal.

I. INTRODUCTION

An important computer communication network design problem is how to connect several remote terminal sites T_i, $1 \leqslant i \leqslant n$, to a central (processing) site C_0. The usual design method uses concentrators. We are given C_j, $1 \leqslant j \leqslant m$, a set of potential concentrator sites which is usually a subset of terminal sites, and we must select a subset $Y \subseteq \{C_1, C_2, \ldots, C_m\}$ to be the set of actual concentrator sites. These concentrators will be connected to the central site via high-speed lines. Each terminal must be connected, via a low-speed line, to a unique site in $Y \cup \{C_0\}$. The number of terminals connected to any $C_j \in Y$ must not exceed a positive integer k ($\leqslant n$) called the concentrator capacity.

The cost of installing a concentrator at site C_j and connecting it to the central site through a high-speed line is d_j. The cost of connecting terminal T_i to site C_j is c_{ij}. We assume (c_{ij}) and (d_j) are non-negative integers. For further relations on (c_{ij}) and (d_j) see [18]. The optimization problem of finding a network with minimum cost is called the star-star concentrator location problem (SSCLP).

The SSCLP is shown to be NP-complete in the strong sense even for the special cases

This work was partly supported by grants from the National Science Foundation and the U.S. Army Research Office (Durham, NC).

NETWORKS, Vol. 15 (1985) 1–20

CCC 0028-3045/85/010001-20$04.00

$k = 3$ and $k = n$ (the latter corresponds to the uncapacitated version of SSCLP). The problem is solvable in polynomial time if $k \leqslant 2$ [17, 18]. Therefore it is unlikely that any polynomial-time algorithm will yield an exact solution in general (see [9]).

The SSCLP is closely related to some location problems of operations research. Such problems have been studied by many workers and both exact and heuristic algorithms have been suggested for their solution (see, e.g., [2, 3, 5, 7, 8, 12, 15, 16].

A heuristic algorithm for a hard combinatorial optimization problem is considered effective if it is time efficient and guarantees good performance. This paper is an attempt to study the performance of some heuristics for SSCLP. The technique used is based on linear programming relaxation and Lagrangian relaxation. An alternative approach could be based on the property called supermodularity (submodularity) [4, 17, 19].

II. NOTATION, ASSUMPTIONS, AND PROBLEM FORMULATIONS

We will assume all the problems considered here are feasible and bounded. If a problem is denoted P, then the optimal value will be denoted $v(\mathrm{P})$. If P is an integer linear programming (ILP) problem, then $\bar{\mathrm{P}}$ denotes the linear programming relaxation (LPR) of P. A Lagrangian will be denoted as LR$i(u)$ where i is an index and u is the Lagrangian multiplier. By this notation then the Lagrangian dual problem, i.e., $\max_u v(\mathrm{LR}i(u))$, will be denoted as LD$i$. If a is any real number, then a^+ denotes $\max\{0, a\}$ and a^- denotes $\min\{0, a\}$ and $\lceil a \rceil$, the ceiling of a, denotes the smallest integer not less than a.

Let x_{ij}, $1 \leqslant i \leqslant n$, $0 \leqslant j \leqslant m$, and y_j, $1 \leqslant j \leqslant m$, be 0–1 variables with the following interpretation.

$$y_j = \begin{cases} 1 & \text{if } C_j \text{ is a selected concentrator site,} \\ 0 & \text{otherwise;} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if } T_i \text{ is connected to } C_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then SSCLP, denoted by P from now on, can be written as

$$v(\mathrm{P}) = \min \sum_{i=1}^{n} \sum_{j=0}^{m} c_{ij} x_{ij} + \sum_{j=1}^{n} d_j y_j \qquad \text{(P)}$$

subject to

(i) $\sum_{j=0}^{m} x_{ij} = 1, \qquad 1 \leqslant i \leqslant n;$

(ii) $\sum_{i=1}^{n} x_{ij} \leqslant k y_j, \qquad 1 \leqslant j \leqslant m;$

(iii) $x_{ij}, y_j \in \{0, 1\}.$

III. THE LINEAR PROGRAMMING RELAXATION OF P

Consider the following problem:

$$v(\hat{P}) = \min \sum_{i=1}^{n} \sum_{j=0}^{m} \bar{c}_{ij} x_{ij} + \sum_{j=1}^{m} \bar{d}_j y_j \qquad (\hat{P})$$

subject to

(i) $\sum_{j=0}^{m} x_{ij} = 1, \qquad 1 \leqslant i \leqslant n;$

(ii) $\sum_{i=1}^{n} x_{ij} \leqslant k y_j, \qquad 1 \leqslant j \leqslant m;$

(iii) $y_j \leqslant 1, \qquad 1 \leqslant j \leqslant m;$

(iv) $x_{ij} \geqslant 0, \qquad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$

We see that $\bar{P}$ is the same as $\hat{P}$ if $\bar{c} = c$ and $\bar{d} = d$. Below we give an algorithm that plugs in different costs of $\bar{c}$ and $\bar{d}$ in $\hat{P}$ and solves $\hat{P}$ with those costs. (Note that the constraints $y_j \geqslant 0$ and $x_{ij} \leqslant 1$ are not needed.) If $\lambda = (\lambda_1, \ldots, \lambda_n)$, $\sigma = (\sigma_1, \ldots, \sigma_m)$, and $\gamma = (\gamma_1, \ldots, \gamma_m)$ are the dual vectors of constraints (i), (ii), and (iii), respectively, then the linear programming dual of $\hat{P}$, denoted by $\hat{D}$, is:

$$v(\hat{D}) = \max \sum_{i=1}^{n} \lambda_i - \sum_{j=1}^{m} \gamma_j \qquad (\hat{D})$$

subject to

(i) $\lambda_i \leqslant \bar{c}_{io}, \qquad 1 \leqslant I \leqslant n;$

(ii) $\lambda_i - \sigma_j \leqslant \bar{c}_{ij}, \qquad 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m;$

(iii) $k\sigma_j - \gamma_j = \bar{d}_j, \qquad 1 \leqslant j \leqslant m;$

(iv) $\sigma_j, \gamma_j \geqslant 0, \qquad 1 \leqslant j \leqslant m.$

In this section we address the issue of solving $\hat{P}$ and $\hat{D}$ efficiently. Let us assume

$$\tilde{c}_{ij} = \begin{cases} \bar{c}_{ij} & \text{if } j = 0, \\ \bar{c}_{ij} + (\bar{d}_j)^+/k & \text{otherwise.} \end{cases}$$

Now consider the following problem:

$$v(\tilde{\mathrm{P}}) = \min \sum_{i=1}^{n} \sum_{j=0}^{m} \tilde{c}_{ij} x_{ij} \qquad (\tilde{\mathrm{P}})$$

subject to

(i) $\sum_{j=0}^{m} x_{ij} = 1, \qquad 1 \leqslant i \leqslant n;$

(ii) $\sum_{i=1}^{n} x_{ij} \leqslant k, \qquad 1 \leqslant j \leqslant m;$

(iii) $x_{ij} \geqslant 0, \qquad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$

Lemma 1. If x is an optimal solution of $\tilde{\mathrm{P}}$, then (x, y) is an optimal solution of $\hat{\mathrm{P}}$, where

$$y_j = \begin{cases} \sum_{i=1}^{n} \frac{x_{ij}}{k} & \text{if } \bar{d}_j \geqslant 0, \\ 1 & \text{if } \bar{d}_j < 0, \end{cases} \quad 1 \leqslant j \leqslant m \qquad (1)$$

and

$$v(\hat{\mathrm{P}}) = v(\tilde{\mathrm{P}}) + \sum_{j=1}^{m} (\bar{d}_j)^{-}. \qquad (2)$$

Proof. First let (x, y) be any optimal solution of $\hat{\mathrm{P}}$. By constraints (ii) and (iii) we have $1 \geqslant y_j \geqslant \Sigma_{i=1}^{n} x_{ij}/k$. If $\bar{d}_j < 0$, then we must have $y_j = 1$, because otherwise we can set $y_j = 1$ and this will yield a new feasible solution in $\hat{\mathrm{P}}$ with lower objective value, a contradiction. With a similar reasoning y_j must be $\Sigma_{i=1}^{n} x_{ij}/k$ if $\bar{d}_j > 0$. If $\bar{d}_j = 0$, y_j can be any number between 1 and $\Sigma_{i=1}^{n} x_{ij}/k$. Therefore

$$v(\hat{\mathrm{P}}) = \bar{c}x + \bar{d}y = \bar{c}x + \sum_{j=1}^{m} \sum_{i=1}^{n} (\bar{d}_j)^{+} \frac{x_{ij}}{k} + \sum_{j=1}^{m} (\bar{d}_j)^{-}.$$

By the definition of $\tilde{c}$, we can then write $v(\hat{\mathrm{P}}) = \tilde{c}x + \Sigma_{j=1}^{m} (\bar{d}_j)^{-}$. Furthermore, it is easy to see that x is feasible in $\tilde{\mathrm{P}}$. Therefore $v(\tilde{\mathrm{P}}) \leqslant \tilde{c}x$, which shows

$$v(\hat{\mathrm{P}}) \geqslant v(\tilde{\mathrm{P}}) + \sum_{j=1}^{m} (\bar{d}_j)^{-}. \qquad (3)$$

Now let x be any optimal solution of $\tilde{\text{P}}$. Then consider y defined by (1). (x, y) is feasible in $\hat{\text{P}}$, therefore

$$v(\hat{\text{P}}) \leqslant \bar{c}x + \bar{d}y = \bar{c}x + \sum_{j=1}^{m} \sum_{i=1}^{n} (\bar{d}_j)^+ \frac{x_{ij}}{k} + \sum_{j=1}^{m} (\bar{d}_j)^-$$

$$= \tilde{c}x + \sum_{j=1}^{m} (\bar{d}_j)^- = v(\tilde{\text{P}}) + \sum_{j=1}^{m} (\bar{d}_j)^-.$$

By this and (3) we see that

$$v(\hat{\text{P}}) = \bar{c}x + \bar{d}y = v(\tilde{\text{P}}) + \sum_{j=1}^{m} (\bar{d}_j)^-,$$

and the lemma follows. ■

Let

$$\tilde{c}_{0j} = 0, \qquad 0 \leqslant j \leqslant m;$$

$$a_i = \begin{cases} 1 & \text{if } 1 \leqslant i \leqslant n, \\ km & \text{if } i = 0; \end{cases}$$

$$b_j = \begin{cases} k & \text{if } 1 \leqslant j < m, \\ n & \text{if } j = 0. \end{cases}$$

Then $\tilde{\text{P}}$ may be written as the following transportation problem (note that we have added a slack node $i = 0$):

$$v(\tilde{\text{P}}_1) = \min \sum_{i=0}^{n} \sum_{j=0}^{m} \tilde{c}_{ij} x_{ij} \qquad (\tilde{\text{P}}_1)$$

subject to

(i) $\sum_{j=0}^{m} x_{ij} = a_i, \quad 0 \leqslant i \leqslant n;$

(ii) $\sum_{i=0}^{n} x_{ij} = b_j, \quad 0 \leqslant j \leqslant m;$

(iii) $x_{ij} \geqslant 0, \quad 0 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$

Let $u = (u_0, u_1, \ldots, u_n)$ and $v = (v_0, v_1, \ldots, v_m)$ be the dual vectors of $\tilde{\text{P}}_1$. Then the dual of $\tilde{\text{P}}_1$ is

$$v(\tilde{D}_1) = \max \sum_{i=0}^{n} a_i u_i + \sum_{j=0}^{m} b_j v_j \qquad (\tilde{D}_1)$$

subject to

(i) $u_i + v_j \leqslant \tilde{c}_{ij}, \qquad 0 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$

We obviously have $v(\tilde{P}) = v(\tilde{P}_1) = v(\tilde{D}_1)$. $\tilde{P}_1$ and $\tilde{D}_1$ can be solved by the primal-dual algorithm for the transportation problem [20]. In order to add efficiency to the algorithm we may start from the (infeasible) solution $x_{0j} = k$, $1 \leqslant j \leqslant m$, and $x_{ij} = 0$ otherwise. Then we need n additional augmentations each taking $O(nm)$ steps. Therefore $\tilde{P}_1$ and $\tilde{D}_1$ can be solved in $O(n^2 m)$ steps. The next two theorems are the main results of this section.

Theorem 1. $\hat{P}$ has an optimal solution (x, y) where x is integral and y satisfies Eq. (1). Furthermore, such a solution can be obtained in $O(n^2 m)$ steps.

Proof. $\hat{P}_1$ has integral optimal solution x and it can be found in $O(n^2 m)$ steps by the primal-dual algorithm. The theorem then follows by Lemma 1. ∎

Theorem 2. If (u, v) is an optimal solution of $\tilde{D}_1$, then $(\bar{\lambda}, \bar{\sigma}, \bar{\gamma})$, defined by (i)-(iii) below, is an optimal solution of $\hat{D}$ and can be found in $O(n^2 m)$ steps.

(i) $\bar{\lambda}_i = u_i + v_0, \qquad 1 \leqslant i \leqslant n;$

(ii) $\bar{\sigma}_j = -(v_j + u_0) + (\bar{d}_j)^+/k, \qquad 1 \leqslant j \leqslant m;$

(iii) $\bar{\gamma}_j = -k(v_j + u_0) - (\bar{d}_j)^-, \qquad 1 \leqslant j \leqslant m.$

Proof. By (i) of $\tilde{D}_1$ we have $v_j + u_0 \leqslant 0$, $1 \leqslant j \leqslant m$. So $\bar{\sigma}_j, \bar{\gamma}_j \geqslant 0$, satisfying constraints (iv) of $\hat{D}$. From (i) of $\tilde{D}_1$ and the definition of $\bar{\lambda}_i$ we see that (i) of $\hat{D}$ is satisfied. We also have $\bar{\lambda}_i - \bar{\sigma}_j = (u_i + v_j) + (u_0 + v_0) - (\bar{d}_j)^+/k \leqslant \tilde{c}_{ij} - (\bar{d}_j)^+/k = \bar{c}_{ij}$. Therefore (ii) of $\hat{D}$ is also satisfied. Also we have $k\bar{\sigma}_j - \bar{\gamma}_j = (\bar{d}_j)^+ + (\bar{d}_j)^- = \bar{d}_j$. So (iii) of $\hat{D}$ is also satisfied. Finally, we have

$$\sum_{i=1}^{n} \bar{\lambda}_i - \sum_{j=1}^{m} \bar{\gamma}_j = \sum_{i=1}^{n} (u_i + v_0) + k \sum_{j=1}^{m} (v_j + u_0) + \sum_{j=1}^{m} (\bar{d}_j)^-$$

$$= \sum_{i=0}^{n} a_i u_i + \sum_{j=0}^{m} b_j v_j + \sum_{j=1}^{m} (\bar{d}_j)^- = v(\tilde{D}_1) + \sum_{j=1}^{m} (\bar{d}_j)^-.$$

By Lemma 1 we have $v(\hat{D}) = v(\tilde{D}_1) + \sum_{j=1}^{m} (\bar{d}_j)^-$. So $v(\hat{D}) = \sum_{i=1}^{n} \bar{\lambda}_i - \sum_{j=1}^{m} \bar{\gamma}_j$, which shows the optimality of $(\bar{\lambda}, \bar{\sigma}, \bar{\gamma})$ in $\hat{D}$. Since an optimal solution (u, v) of $\tilde{D}_1$ can be found, by the primal-dual algorithm, in $O(n^2 m)$ steps, then $(\bar{\lambda}, \bar{\sigma}, \bar{\gamma})$ can be found in $O(n^2 m)$ steps, completing the proof of the theorem. ∎

IV. TWO LAGRANGIAN RELAXATIONS OF P

One of the reasons for considering the Lagrangian relaxations (LRs) instead of the linear programming relaxation (LPR) is that any Lagrangian dual (LD) will yield a lower bound at least as good as the lower bound obtained from the LPR. For an excellent introduction to the theory of Lagrangian relaxation see [10, 11].

In this section two different LRs of P based on the two constraint sets (i) and (ii) of P will be considered. It will be shown that both resulting LDs are equivalent to some LP problems, one of which is $\bar{P}$, the LPR of P; the other is $\bar{P}_2$, described below, which is the same as $\bar{P}$ except with an additional set of constraints. Later on we will consider a third LR based on $\bar{P}_2$.

A. The Lagrangian Relaxation LR2 (σ)

We relax the constraints (ii) of P in a Lagrangian fashion. Assuming $\sigma = (\sigma_1, \ldots, \sigma_m)$ is a non-negative real vector, the Lagrangian is

$$v(\text{LR2}(\sigma)) = \min \left[cx + dy + \sum_{j=1}^{m} \sigma_j \left(\sum_{i=1}^{n} x_{ij} - ky_j \right) \right] \qquad (\text{LR2}(\sigma))$$

subject to (i) and (iii) of P.

LR2(σ) satisfies the *integrality property*, since the constraint matrix is *totally unimodular* [note that (iii) of P can be broken up into two constraint sets, namely $\{0 \leqslant x_{ij}, y_j \leqslant 1\}$ and $\{x_{ij}, y_j \text{ integral}\}$.] Hence

$$v(\text{LD2}) = v(\bar{P}) = v(\text{LR2}(\bar{\sigma})) \qquad (4)$$

where $\bar{\sigma}$ is an optimal dual vector corresponding to the constraint set (ii) of $\bar{P}$. Equation (4) shows that we do not obtain a better lower bound by solving LD2 instead of $\bar{P}$. In the next subsection we show an LR for which the corresponding LD indeed yields a better lower bound than $v(\bar{P})$ in general.

B. The Lagrangian Relaxation LR1 (λ)

Now we relax the constraints (i) of P in a Lagrangian fashion. Assuming $\lambda = (\lambda_1, \ldots, \lambda_n)$ is a real vector, the Lagrangian is

$$v(\text{LR1}(\lambda)) = \min \left[cx + dy + \sum_{i=1}^{n} \lambda_i \left(1 - \sum_{j=0}^{m} x_{ij} \right) \right] \qquad (\text{LR1}(\lambda))$$

$$= \min \left(\sum_{i=1}^{n} \sum_{j=0}^{m} (c_{ij} - \lambda_i) x_{ij} + \sum_{j=1}^{m} d_j y_j + \sum_{i=1}^{n} \lambda_i \right)$$

subject to (ii) and (iii) of P.

LR1(λ) does not satisfy the integrality property, and in general we have

$$v(\text{LD1}) \geqslant v(\bar{P}). \qquad (5)$$

Consider the following two questions. First, given a λ how do we solve LR1(λ) efficiently? Second, what choice is best for λ? In other words, what λ_0 satisfies $v(\text{LR1}(\lambda_0)) = v(\text{LD1})$? [Note that $v(\text{LD1}) = \max_\lambda v(\text{LR1}(\lambda))$.]

Let us consider the first question first. For each j, $1 \leqslant j \leqslant m$, consider a nondecreasing ordering of $c_{ij} - \lambda_i$, $1 \leqslant i \leqslant n$. Let the i index sequence of the ordering be $J(j) = (i_1(j), \ldots, i_n(j))$, $1 \leqslant j \leqslant m$. Then consider the following index sets:

$$\hat{J}(j) = \{i_t(j) \mid 1 \leqslant t \leqslant k, c_{i_t(j)j} - \lambda_{i_t(j)} < 0\}, \qquad 1 \leqslant j \leqslant m;$$
$$\hat{J}(0) = \{i \mid c_{i0} - \lambda_i < 0, 1 \leqslant i \leqslant n\}.$$

Each $\hat{J}(j)$, $1 \leqslant j \leqslant m$, can be found in $O(n)$ time, using the well-known linear selection algorithm [1]. (This algorithm finds the kth smallest element from a list.) Therefore $\hat{J}(j)$, for all j, $0 \leqslant j \leqslant m$, can be found in $O(mn)$ time total. An optimal solution to LR1(λ) is then given by the following theorem.

Theorem 3. An optimal solution of LR1(λ) is

$$y_j = \left\{\begin{array}{ll} 1 & \text{if } d_j + \sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i) < 0 \\ 0 & \text{otherwise} \end{array}\right\} \quad 0 \leqslant j \leqslant m;$$

$$x_{ij} = \left\{\begin{array}{ll} y_j & \text{if } i \in \hat{J}(j) \\ 0 & \text{otherwise} \end{array}\right\} \quad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$$

Furthermore, such a solution can be found in $O(mn)$ time. (Note: y_0 is not part of the solution, but it is used conveniently with $d_0 = 0$.)

Proof. The proof is by the fact that if y_j is set to 1, then among $c_{ij} - \lambda_i$, $1 \leqslant i \leqslant n$, the k (or less) most negative ones are selected and the corresponding x_{ij} are set to 1. The proof is complete by the discussion preceding the theorem. ■

By Theorem 3 the first question is answered. In what follows, we attempt to answer the second question. By Theorem 3 we may write

$$v(\text{LR1}(\lambda)) = \sum_{i=1}^{n} \lambda_i + \sum_{j=0}^{m} \left(d_j + \sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i)\right)^- . \tag{6}$$

Equation (6) is the basis of characterizing LD1 with an equivalent LP. The proof of the following theorem is given in Appendix A.

Theorem 4. $v(\text{LD1}) = v(\text{LR1}(\lambda_0))$, where λ_0 is an optimal dual vector of constraints (i) of the following LP:

$$v(\bar{\text{P}}_2) = \min \left(\sum_{i=1}^{n} \sum_{j=0}^{m} c_{ij} x_{ij} + \sum_{j=1}^{m} d_j y_j\right) \tag{$\bar{\text{P}}_2$}$$

subject to (i)-(iv) of $\hat{\mathrm{P}}$ and

$$\text{(v)}\quad x_{ij} \leqslant y_j, \qquad 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m. \qquad \blacksquare$$

If we dualize $\bar{\mathrm{P}}_2$ we get the following LP:

$$v(\bar{\mathrm{D}}_2) = \max\left(\sum_{i=1}^{n} \lambda_i - \sum_{j=1}^{m} \gamma_j\right) \qquad (\bar{\mathrm{D}}_2)$$

subject to

$$\text{(i)}\quad \lambda_i \leqslant c_{io}, \qquad 1 \leqslant i \leqslant n;$$

$$\text{(ii)}\quad \lambda_i - \sigma_j - \beta_{ij} \leqslant c_{ij}, \qquad 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m;$$

$$\text{(iii)}\quad k\sigma_j + \sum_{i=1}^{n} \beta_{ij} - \gamma_j = d_j, \qquad 1 \leqslant j \leqslant m;$$

$$\text{(iv)}\quad \beta_{ij}, \sigma_j, \gamma_j \geqslant 0, \qquad 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m;$$

λ_0 of Theorem 4 is an optimal solution (λ) of $\bar{\mathrm{D}}_2$.

We may notice that $\bar{\mathrm{P}}_2$ is the same as $\hat{\mathrm{P}}$ (i.e., $\hat{\mathrm{P}}$ with $\bar{c} = c$ and $\bar{d} = d$), except that $\bar{\mathrm{P}}_2$ has the additional constraints (v) and this shows once more that $v(\mathrm{LD1}) \geqslant v(\bar{\mathrm{P}})$.

Theorem 4 gives a condition for λ_0 to solve the Lagrangian dual LD1. Whether $\bar{\mathrm{D}}_2$ (or $\bar{\mathrm{P}}_2$ for that matter) can be solved in a number of steps that is polynomial in n and m is not known to the author at the time of this writing. The next section considers a third LP based on $\bar{\mathrm{P}}_2$ that will be the core of the final algorithm.

V. AN APPROXIMATION PROCEDURE FOR P

In this section we will consider an approximation algorithm for P based on $\bar{\mathrm{P}}_2$. We consider an LR of $\bar{\mathrm{P}}_2$ by relaxing the constraints (v) of $\bar{\mathrm{P}}_2$. If $\beta = (\beta_{ij}) \geqslant 0$ is the corresponding Lagrangian multiplier vector, then the Lagrangian is:

$$v(\mathrm{LR3}(\beta)) = \min \quad cx + dy + \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_{ij}(x_{ij} - y_j) \qquad (\mathrm{LR3}(\beta))$$

subject to (i)-(iv) of $\bar{\mathrm{P}}_2$.

We see that LR3(β) is the same as $\hat{\mathrm{P}}$ with $\bar{c}_{ij} = c_{ij}$, for $1 \leqslant i \leqslant n, j = 0$, and $\bar{c}_{ij} = c_{ij} + \beta_{ij}$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$, and $\bar{d}_j = d_j - \sum_{i=1}^{n} \beta_{ij}$, for $1 \leqslant j \leqslant m$. Furthermore $v(\mathrm{LD3}) = v(\mathrm{LR3}(\beta_0))$, where β_0 is any optimal solution of $\bar{\mathrm{D}}_2$. For a given β the procedure PR(β), given below, will yield a lower bound and a feasible solution to P.

Procedure PR(β)

Step 1. Solve LR3(β). Let $(x, \bar{y})$ be the optimal solution found, where x is integral and $\bar{y}$ satisfies Eq. (1). Set $\mathrm{LB}(\beta) = v(\mathrm{LR3}(\beta))$ as a lower bound to v(P).

Step 2. Set (x, y) as a feasible solution to P, where $y_j = \lceil \bar{y}_j \rceil$, $1 \leq j \leq m$. Set UB(β) $cx + dy$ as an upper bound to v(P).

End.

By Theorem 1 the procedure PR(β) takes $O(n^2 m)$ steps. Also note that Step 1 c procedure PR($\beta = 0$) will correspond to solving $\bar{\text{P}}$. The next theorem gives an uppe bound to the worst-case performance of PR(β) with $\beta = 0$, which will suggest that thei might exist a β for which PR(β) has better performance. The latter issue will be coı sidered in the next section.

Theorem 5. The following statements hold true:

(i) $\text{LB}(\beta) \leq v(\text{P}) \leq \text{UB}(\beta)$,
(ii) $\text{UB}(0) \leq k\,\text{LB}(0)$,
(iii) $v(\text{P}) \leq k v(\bar{\text{P}})$.

Proof. (i) The left-hand inequality of (i) is obvious since $v(\text{LR3}(\beta)) \leq v(\text{P})$ for an $\beta \geq 0$. The right-hand inequality is also clear since (x, y) obtained at Step 2 of PR(β) i feasible in P.

(ii) Procedure PR(β) produces integral x. By Eq. (1) we have $\bar{y}_j$ equal to 1 or $\Sigma_{i=}^{n}$ x_{ij}/k, $1 \leq j \leq m$. Therefore $y_j = \lceil \bar{y}_j \rceil \leq k\bar{y}_j$. Hence if $\beta = 0$, we have

$$k\,\text{LB}(0) = k\left(\sum_{i=1}^{n}\sum_{j=0}^{m} \bar{c}_{ij}x_{ij} + \sum_{j=1}^{m} \bar{d}_j\bar{y}_j\right) = k(cx + d\bar{y})$$

$$\geq kcx + dy \geq cx + dy = \text{UB}(0).$$

(iii) By the right-hand inequality of (i) and the fact that $\text{LB}(0) = v(\bar{\text{P}})$ statement (iii is obviously true. ∎

In the next section procedure PR(β) is used in an iterative routine for the subgra dient optimization technique.

VI. APPROXIMATION ALGORITHM FOR P

The paper by Held and Karp [13] on the traveling-salesman problem introduced a surprisingly effective method of iteration that converges to a solution of a Lagrangiaı dual problem. Held et al. [14] generalized the method somewhat and called it the subgradient optimization technique. Since then there has been considerable effort ex pended in improving the method and trying to get a better understanding of the under lying principles. The Lagrangian relaxation coupled with the subgradient optimizatioı technique is one of the most widely used methods in finding tight lower bounds foı hard combinatorial minimization problems. First let us consider the simplified versioı of the final algorithm.

Algorithm A1

1. *Set* $t = 1$, $\beta^t = 0$, and $\bar{t}$ as the iteration limit.
2. *While* $t \leq \bar{t}$ *do*

2.1. call procedure PR(β^t) to find $x, \bar{y}, y$, LB(β^t), and UB(β^t);
2.2. "alter β with the subgradient optimization technique"

set $$w^t = r^t(\mathrm{UB}(\beta^t) - \mathrm{LB}(\beta^t)) \Big/ \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - \bar{y}_j)^2,$$

$$\beta_{ij}^{t+1} = [\beta_{ij}^t + w^t(x_{ij} - \bar{y}_j)]^+;$$

2.3. $t = t + 1$.
End.

In the subgradient optimization technique the coefficients r^t are usually taken as $0 < r^t \leqslant 2$. But in general if $\lim_{t\to\infty} r^t = 0$ and $\lim_{\bar{t}\to\infty} \Sigma_{t=1}^{\bar{t}} r^t = \infty$, it is guaranteed that β^t converges to a solution of the Lagrangian dual (see [14] and references cited therein).

There are a number of points to explain before giving the final version of the algorithm. In SSCLP the potential concentrator sites are usually at a subset of terminal sites. In [18] it is shown that there is an optimal solution in which if C_j is a selected concentrator site, then T_j is connected to C_j. So we may change the inequalities $x_{ij} \leqslant y_j$ for $i = j$ to $x_{jj} - y_j = 0$ in $\bar{\mathrm{P}}_2$. This in turn says that the Lagrangian multipliers β_{jj}, $1 \leqslant j \leqslant m$, need not be restricted to be non-negative. Secondly, at the iteration routine of the algorithm we may use a postoptimization heuristic to improve the upper bound. The idea is from the proof of Facts 1 and 2 in [18]. The heuristic is as follows.

Heuristic POST-OPT1

1. Let (x, y) be a feasible solution to P. Set $(\hat{x}, \hat{y}) = (x, y)$.
2. *For j from* 1 *to m do*
   if $\hat{x}_{jj} = 0$ and $\hat{y}_j = 1$ *then* let j_1, $0 \leqslant j_1 \leqslant m$, be such
   that $\hat{x}_{jj_1} = 1$, and $I_j = \{i \mid x_{ij} = 1\}$, let $i_j \in I_j$
   be the solution to $\min_{i \in I_j} \{c_{jj} + c_{ij_1} - c_{jj_1} - c_{ij}\}$;
   then set $\hat{x}_{jj} = \hat{x}_{i_j j_1} = 1, \hat{x}_{i_j j} = \hat{x}_{jj_1} = 0$.
3. *For j from* 1 *to m do*
   if $\Sigma_{i=1}^{n} \hat{x}_{ij} \geqslant 1$ *then* set $\hat{y}_j = 1$, *otherwise* set $\hat{y}_j = 0$.
4. $(\hat{x}, \hat{y})$ is an enhanced feasible solution (i.e., $c\hat{x} + d\hat{y} \leqslant cx + dy$).
   Output $(\hat{x}, \hat{y})$.

End.

Another postoptimization may be used at the end of the algorithm (or instead of POST-OPT1 at each iteration). If $Y \subseteq \{1, \ldots, m\}$ is the set of selected concentrator sites, then let $Z(Y)$ be the optimal network cost using concentrators at sites denoted by Y. $Z(Y)$ can be found in $O(n^2 m)$ by the primal-dual algorithm for the transportation problem.

In the final algorithm we keep a record of the best lower and upper bounds found. After the last iteration we take Y as the set of selected concentrator sites from the best upper bound, and postoptimize it by finding $Z(Y)$ and the associated solution (x, y) as the final upper bound and feasible solutions, respectively. Since the optimal value

v(P) is integral, then we take the ceiling of the lower bounds as the actual lower bounds. The final version of the algorithm is as follows.

Algorithm NETWORK

1. *Initialization:* Set $t = 1$, $\beta^t = 0$, $\bar{z} = \infty$, $\underline{z} = 0$, where $\bar{z}$ and $\underline{z}$ are the upper and lower bounds, respectively; also initialize $\bar{t}$ and $\bar{\eta}$ as the iteration and performance limit, respectively; set $\eta = \infty$.
2. *While* $t \leqslant \bar{t}$ and $\eta \geqslant \bar{\eta}$ *do*
   call procedure PR(β^t) to obtain $x, \bar{y}, y$, LB(β^t), UB(β^t),
   call procedure POST-OPT1 to obtain $\hat{x}, \hat{y}, z_u^t$ $(= c\hat{x} + d\hat{y})$;
   set $\underline{z} = \max\{\underline{z}, \text{LB}(\beta^t)\}$
   if $\bar{z} > z_u^t$ *then* set $Y = \{j \mid \hat{y}_j = 1\}$, $\bar{z} = z_u^t$
   set $\eta = (\bar{z} - \lceil \underline{z} \rceil)/\lceil \underline{z} \rceil$, $t = t + 1$
   "alter β:"

   set $$w^t = r^t[\bar{z} - \text{LB}(\beta^t)] \Big/ \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - \bar{y}_j)^2,$$

   $$\beta_{ij}^{t+1} = \begin{cases} [\beta_{ij}^t + w^t(x_{ij} - \bar{y}_j)]^+ & \text{if } i \neq j, \\ \beta_{ij}^t + w^t(x_{ij} - \bar{y}_j) & \text{if } i = j. \end{cases}$$

3. "Final postoptimization." Set $y_j = 1$ if $j \in Y$, $y_j = 0$ otherwise. Find $z(Y)$. The lower bound is $\underline{z}$. The upper bound is $z(Y)$. The performance bound is $\eta = [z(Y) - \lceil \underline{z} \rceil]/\lceil \underline{z} \rceil$.

End.

Theorem 6. If η is the performance ratio found by algorithm NETWORK, then $\eta \leqslant k - 1$.

Proof. Since $\beta^1 = 0$, by Theorem 5 we have UB(β^1) $\leqslant k$ LB(β^1). Observing that $z(Y) \leqslant \bar{z} \leqslant \text{UB}(\beta^1)$ and $\lceil \underline{z} \rceil \geqslant \text{LB}(\beta^1)$, the theorem follows. ■

VII. SOME COMPUTATIONAL RESULTS

Recall that n is the number of terminals, m is the maximum number of concentrators, and k is the concentrator capacity.

Three networks were considered with (n, m) equal to (50, 20), (40, 20), and (23, 5). For each network three different values of k were considered, namely 3, 5, and 7. In all these networks the first m terminal sites were considered as the potential concentrator sites. Figures 1, 2, and 3 show the network designs obtained. Table I shows a summary of the computational results. In all cases the postoptimized solutions are seen to be within 2.8% from optimal. (The performance bound mentioned in Table I is the percentage difference between lower and upper bounds.) The details of each network are given in Appendix B. One more detail remains to be explained, and that is the choice of r^t. We decrease r^t linearly from r^1 to r^{t_0} for some t_0, $1 < t_0 < \bar{t}$, then decrease r^t geometrically from r^{t_0} to $r^{\bar{t}}$. That is,

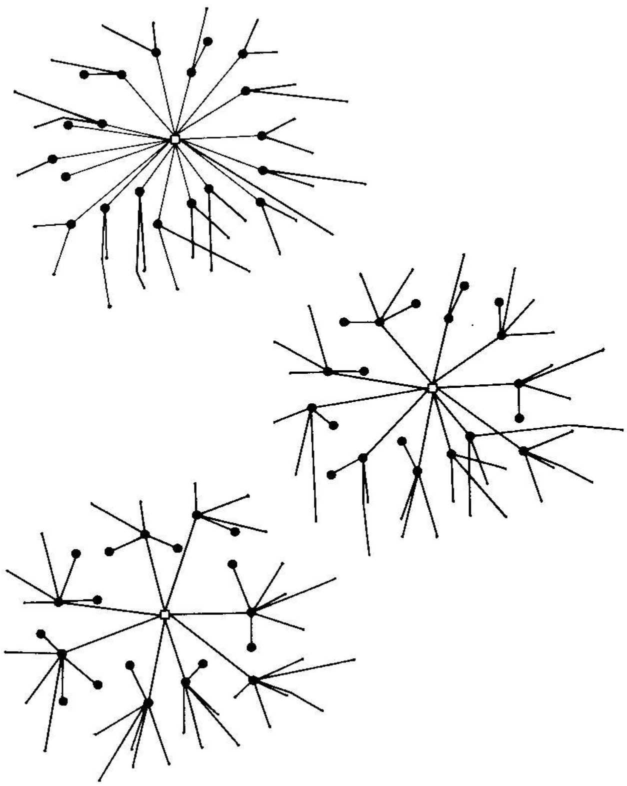

FIG. 1. Network designs with $(n, m) = (50, 20)$. The node □ is the central site, and the nodes ● are the potential concentrator sites.

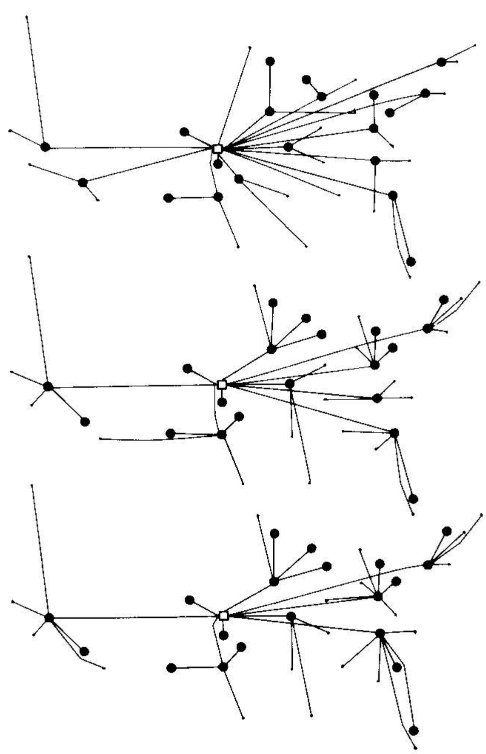

FIG. 2. Network designs with $(n, m) = (40, 20)$. The node □ is the central site, and the nodes ● are the potential concentrator sites.

TABLE I. Summary of computational results; $\hat{t}$ is the number of iterations the algorithm takes.

Number of Terminals n	Maximum Number of Concentrators m	Number of Concentrators Used m'	Concentrator Capacity k	Iterations $\hat{t}$	Lower Bound $\underline{z}$	Lower Bound $\lceil \underline{z} \rceil$	Upper Bound $z(Y)$	Performance Bound η (%)
50	20	16	3	100	368.6665	369	371	0.54
		11	5	100	297.2233	298	305	2.35
		8	7	100	275.3731	276	278	0.72
40	20	12	3	100	321.4991	322	331	2.79
		8	5	100	247.9994	248	254	2.42
		7	7	100	231.4870	232	234	0.86
23	5	5	3	1	264	264	264	0.0
		4	5	25	222.6395	223	223	0.0
		4	7	15	217.1178	218	218	0.0

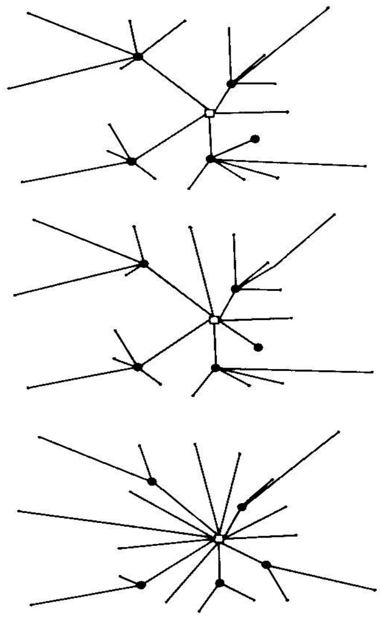

FIG. 3. Network designs with $(n, m) = (23, 5)$. The node □ is the central site, and the nodes ● are the potential concentrator sites.

$$r^t = \begin{cases} r^{t-1} - (r^1 - r^{t_0})/(t_0 - 1) & \text{for } 1 < t \leqslant t_0, \\ r^{t-1}(r^{\bar{t}}/r^{t_0})\ 1/(\bar{t} - t_0) & \text{for } t_0 < t \leqslant \bar{t} \end{cases}$$

For a justification of such a choice see [14]. We have selected $\bar{t} = 100$, $t_0 = 50$, $r^1 = 10$, $r^{t_0} = 2$, $r^{\bar{t}} = 0.08$.

VIII. CONCLUDING REMARKS

We have described an algorithm for the approximation of SSCLP. The algorithm produces both lower and upper bounds. A worst-case performance bound has been established and some computational results have been performed which show the effectiveness of the algorithm. Here we would like to mention the following open questions.

(i) Can we improve the worst-case performance bound?

(ii) We characterized the solution λ_0 to $\max_\lambda v(\text{LR1}(\lambda))$ as being an optimal solution of an LP. Can this LP be solved in polynomial time in n and m?

(iii) The formulation of LR1(λ) resulted in finding the extra linear constraints (v) of $\bar{P}_2$. We know that $\sum_{i=1}^n \sum_{j=0}^m x_{ij} = n$ must be satisfied by any feasible solution of P. So if we add this constraint, even better lower bounds may result. The question is, is the new Lagrangian solvable efficiently? If yes, can we characterize the new LD with an LP that has additional constraints and further characterizes the feasible solutions to P?

(iv) Can we use the dual vectors λ, σ, and γ of $\bar{D}_2$, in addition to β, in an algorithm to obtain more information on the intermediate solutions of the algorithm? One way to do this is the use of the following proposition in a multiplier-adjustment-based algorithm [6].

Proposition 1. An optimal solution σ, β, γ of LR1(λ) (i.e., $\bar{D}_2$ with a fixed λ) is given by (i)-(iii) below:

$$\text{(i)}\quad \sigma_j = \max_{i \notin \hat{J}(j)} (\lambda_i - c_{ij})^+, \qquad 1 \leqslant j \leqslant m;$$

$$\text{(ii)}\quad \beta_{ij} = (\lambda_i - \sigma_j - c_{ij})^+, \qquad 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m;$$

$$\text{(iii)}\quad \gamma_j = \Big(\sum_{i \in \hat{J}(j)} (\lambda_i - c_{ij}) - d_j \Big)^+, \qquad 1 \leqslant j \leqslant m.$$

Proof Outline. It can be shown that (i)-(iii) satisfy (ii)-(iv) of $\bar{D}_2$; the objective value of $\bar{D}_2$ with this solution is equal to $v(\text{LR1}(\lambda))$ as given in Eq. (6). ∎

APPENDIX A (PROOF OF THEOREM 4)

From Eq. (6) noting that $v(\text{LD1}) = \max_\lambda v(\text{LR1}(\lambda))$, we can write

$$v(\text{LD1}) = \max_{\lambda, \gamma} \sum_{i=1}^{n} \lambda_i - \sum_{j=0}^{m} \gamma_j \tag{A1}$$

subject to

$$\gamma_j \geqslant -d_j - \sum_{i \in \hat{J}(i)} (c_{ij} - \lambda_i), \qquad 0 \leqslant j \leqslant m,$$

$$\gamma_j \geqslant 0, \qquad 0 \leqslant j \leqslant m.$$

Assuming $k_j = k$ for $1 \leqslant j \leqslant m$ and $k_0 = n$, we can evaluate $\Sigma_{i \in \hat{J}(j)} (c_{ij} - \lambda_i)$, for $0 \leqslant j \leqslant m$, as follows:

$$-\sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i) = \max_{w} \sum_{i=1}^{n} (\lambda_i - c_{ij}) w_{ij}$$

subject to

$$\text{(i)}\quad \sum_{i=1}^{n} w_{ij} \leqslant k_j;$$

$$\text{(ii)}\quad 0 \leqslant w_{ij} \leqslant 1, \qquad 1 \leqslant i \leqslant n.$$

Dualizing the linear program above, we get

$$-\sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i) = \min_{\sigma, \beta} k_j \sigma_j + \sum_{i=1}^{n} \beta_{ij} \tag{A2}$$

subject to

(i) $-\sigma_j - \beta_{ij} \leqslant c_{ij} - \lambda_i, \quad 1 \leqslant i \leqslant n;$

(ii) $\sigma_j \geqslant 0, \beta_{ij} \geqslant 0, \quad 1 \leqslant i \leqslant n.$

Lemma 2. $v(\mathrm{LD1}) = v(\mathrm{LR1}(\lambda_0))$, where λ_0 is an optimal solution of the following linear program:

$$z = \max \sum_{i=1}^{n} \lambda_i - \sum_{j=0}^{m} \gamma_j \tag{A3}$$

subject to

(i) $\lambda_i - \sigma_j - \beta_{ij} \leqslant c_{ij}, \quad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m;$

(ii) $k_j \sigma_j + \sum_{i=1}^{n} \beta_{ij} - \gamma_j \leqslant d_j, \quad 0 \leqslant j \leqslant m;$

(iii) $\beta_{ij}, \sigma_j, \gamma_j \geqslant 0, \quad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$

Proof. Suppose $(\lambda, \sigma, \beta, \gamma)$ is an optimal solution to (A3). From (i) and (iii) of (A3) we see that (σ, β) is feasible in (A2). Hence

$$k_j \sigma_j + \sum_{i=1}^{n} \beta_{ij} \geqslant \sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i), \quad 0 \leqslant j \leqslant m.$$

From (ii) and (iii) of (A3) we have $\gamma_j \geqslant 0$ and

$$\gamma_j \geqslant -d_j + k_j \sigma_j + \sum_{i=1}^{n} \beta_{ij} \geqslant -d_j - \sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i).$$

Therefore (λ, γ) is feasible in (A1). Hence

$$v(\mathrm{LD1}) \geqslant \sum_{i=1}^{n} \lambda_i - \sum_{j=0}^{m} \gamma_j = z. \tag{A4}$$

Now suppose (λ, γ) is optimal in (A1). From (A2) we know that there exist (σ, β) such that

$$-\sum_{i \in \hat{J}(j)} (c_{ij} - \lambda_i) = k_j \sigma_j + \sum_{i=1}^{n} \beta_{ij}, \quad 0 \leqslant j \leqslant m,$$

and (σ, β) satisfy the constraints of (A2). So $(\lambda, \sigma, \beta, \gamma)$ is feasible in (A3). Hence

$$v(\text{LD1}) = \sum_{i=1}^{n} \lambda_i - \sum_{j=0}^{m} \gamma_j \leqslant z. \tag{A5}$$

From (A4) and (A5) the lemma follows. ∎

Now it is a simple matter to show that ($\bar{\text{P}}_2$), given in Theorem 4, and (A3) are equivalent. To show this we dualize (A3); the following LP results:

$$\min \left(\sum_{i=1}^{n} \sum_{j=0}^{m} c_{ij} x_{ij} + \sum_{j=0}^{m} d_j y_j \right)$$

subject to

(i) $\sum_{j=0}^{m} x_{ij} = 1, \quad 1 \leqslant i \leqslant n;$

(ii) $\sum_{i=1}^{n} x_{ij} \leqslant k_j y_j, \quad 0 \leqslant j \leqslant m;$

(iii) $x_{ij} \leqslant y_j, \quad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m;$

(iv) $0 \leqslant y_j \leqslant 1, \quad 0 \leqslant j \leqslant m;$

(v) $x_{ij} \geqslant 0, \quad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m.$

In the LP above we may remove the constraint $y_j \geqslant 0$ in (iv), since it is implied by (ii) and (v). Secondly, by the fact that $d_0 = 0$ we may set $y_0 = 1$ without affecting the optimality. By doing this then constraints (ii)-(iv) can be removed for $j = 0$. The resulting LP is $\bar{\text{P}}_2$ of Theorem 4.

APPENDIX B

The data are obtained as follows. We consider $n + 1$ points $p_0, p_1, \ldots, p_n$ on the Euclidean plane. The point p_0 is the central site, and $p_1, \ldots, p_m$ are considered as the potential concentrator sites. The cost matrices are calculated as follows:

$$c_{ij} = \lceil |x_i - x_j| + |y_i - y_j| \rceil, \quad 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant m,$$

where (x_i, y_i) is the x-y coordinate of p_i. We can see that (c_{ij}) satisfy the triangle inequality. We set

$$d_j = \bar{m} c_{j0}, \qquad 1 \leqslant j \leqslant m,$$

where $\bar{m} > 1$ is a proportionality constant. We have used $\bar{m} = 2$.

Now it is sufficient to give the coordinates of the terminals and the central sites for networks mentioned in Table I. These are shown in Tables II–IV.

TABLE II. Coordinates of sites of the network with $(n, m) = (50, 20)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x_i	11	9	12	16	15	13	10	6	7	5	7	10	13	16	16
y_i	16	13	12	14	19	22	21	20	17	14	12	11	13	12	16
i	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
x_i	15	12	8	5	4	5	7	9	12	14	15	18	19	19	18
y_i	21	20	20	17	15	11	9	8	9	10	11	11	13	15	17
i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
x_i	18	17	16	13	10	7	4	2	3	2	3	4	7	9	11
y_i	19	21	23	24	23	23	21	19	17	14	11	8	6	7	7
i	45	46	47	48	49	50									
x_i	13	15	17	20	22	21									
y_i	8	8	9	10	13	18									

TABLE III. Coordinates of the sites of the network with $(n, m) = (40, 20)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x_i	15	18	19	18	20	21	24	25	24	24	25	26	27	28	16
y_i	12	14	12	17	16	15	15	14	13	11	9	5	15	17	10
i	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
x_i	15	15	13	12	7	5	30	29	28	26	26	25	24	22	23
y_i	11	9	13	9	10	12	18	17	15	11	4	12	8	9	16
i	30	31	32	33	34	35	36	37	38	39	40				
x_i	23	21	21	20	19	17	16	8	4	4	3				
y_i	14	13	11	6	9	18	6	9	20	11	13				

TABLE IV. Coordinates of the sites of the network with $(n, m) = (23, 5)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x_i	19	16	25	26	19	15	21	26	27	28	28	24	21	16	13
y_i	15	12	9	20	20	18	5	5	10	16	19	22	23	22	22
i	15	16	17	18	19	20	21	22	23						
x_i	12	13	14	17	35	8	36	5	37						
y_i	15	12	9	6	4	4	22	21	12						

References

[1] V. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, MA (1974).

[2] L. R. Bahl and D. T. Tang, "Optimization of concentrator location in teleprocessing networks, in Proceedings, Symposium on Computer Communications and Teletraffic, Polytechnic Institute of Brooklyn, New York, 1972.

[3] R. R. Boorstyn and H. Frank, Large-scale network topological optimization. *IEEE Trans. Comm.* **COM-25** (1977) 29–47.

[4] J. Edmonds, Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications,* R. Guy, Ed. Gordon and Breach, New York (1971), pp. 69–87.

[5] M. A. Efroymson and T. L. Ray, A branch-and-bound algorithm for the plant location. *Operations Res.* **14** (1966) 361–368.

[6] D. Erlenkotter, A dual-based procedure for uncapacitated facility location. *Operations Res.* **26** (1978) 992–1009.

[7] L. R. Esau and K. C. Williams, On teleprocessing system design. *IBM Systems J.* **5** (1966) 142–147.

[8] E. Feldman, F. A. Lehrer, and T. L. Ray, Warehouse location under continuous economies of scale. *Management Sci.* **12** (1966) 670–684.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco (1979).

[10] A. M. Geoffrion, Duality in nonlinear programming: A simplified application-oriented development. *SIAM Rev.* **13** (1971) 1–37.

[11] A. M. Geoffrion, Lagrangian relaxation for integer programming. *Math Programming Study* **2** (1974) 82–114.

[12] G. Y. Handler and P. B. Mirchandani, *Location on Networks: Theory and Algorithms.* MIT, Cambridge, MA (1979).

[13] M. Held and R. M. Karp, The travelling-salesman problem and minimum spanning trees: Part II. *Math. Programming* **1** (1971) 6–25.

[14] M. Held, P. Wolfe, and H. P. Crowder, Validation of subgradient optimization. *Math. Programming* **6** (1974) 62–88.

[15] B. M. Khumawala, Warehouse location problem, efficient branch-and-bound algorithm. *Management Sci.* **18** (1972) B718–B731.

[16] A. A. Kuehn and M. J. Hamburger, A heuristic program for locating warehouses. *Management Sci.* **9** (1963) 643–666.

[17] A. Mirzaian, Topological optimization of computer communication networks: Approximate algorithms and bounds. Ph.D. Thesis, Princeton University (1981).

[18] A. Mirzaian and K. Steiglitz, A note on the complexity of the star-star concentrator problem. *IEEE Trans. Comm.* **COM-29** (1981) 1549–1552.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, An analysis of approximations for maximizing submodular set functions–I. *Math. Programming* **14** (1978) 265–294.

[20] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ (1982).

Received January 25, 1982
Accepted September 12, 1984