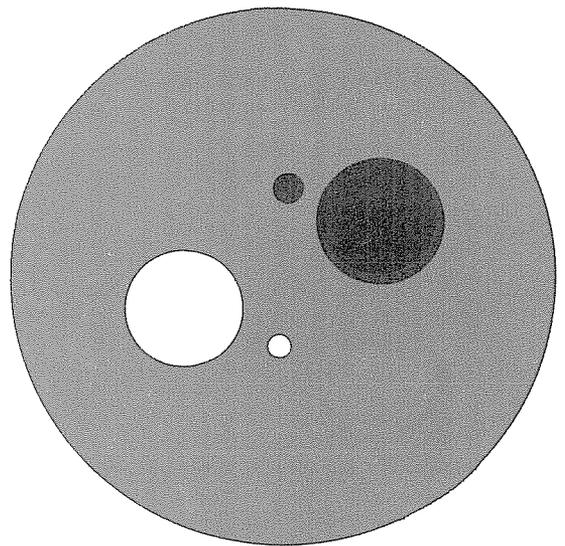


COMPUTER SCIENCES DEPARTMENT

University of Wisconsin-
Madison



Construction of Plant Location Test
Problems with Known Optimal Solutions

by

Jay M. Fleisher

Computer Sciences Technical Report #263

December 1975

Computer Sciences Department
1210 West Dayton Street
Madison, Wisconsin 53706

Received November 4, 1975

Construction of Plant Location Problems
With Known Optimal Solutions*

J. M. Fleisher

* This research was supported by the National Science Foundation
(Grant DCR 74-20584)

Construction of Plant Location Test
Problems with Known Optimal Solutions

Jay M. Fleisher

November 1975

ABSTRACT

This report presents a procedure for the construction of plant location problems with known optimal solutions. Plant location problems can be formulated as mixed integer programming problems and thus provide a source of problems for testing integer programming codes. The data comprising these problems will be selected in such a manner that sufficient optimality conditions will be satisfied.

Computational results are given to show that this procedure can efficiently generate difficult test problems of this type.



1. INTRODUCTION

1.1 Statement of the Problem

The type of plant location problem considered in this report has the following form:

$$(PLN) \text{ Minimize } \sum_{k=1}^p c_k x_k + \sum_{i=1}^m g_i y_i$$

$$\text{subject to } - \sum_{s_k=1} x_k + a_i y_i \geq 0, \quad i = 1, 2, \dots, m,$$

$$\sum_{d_k=j} x_k = b_j, \quad j = 1, 2, \dots, n,$$

$$x_k \geq 0, \quad k = 1, 2, \dots, p, \quad y_i = 0 \text{ or } 1, \quad i = 1, 2, \dots, m,$$

where

m = number of supply points (SPs),

n = number of demand points (DPs),

p = number of shipping routes,

s_k = index of the supply point corresponding to the kth route,

d_k = index of the demand point corresponding to the kth route,

c_k = cost of shipping one unit along the kth route from SP(s_k) to DP(d_k);

g_i = fixed cost incurred if SP(i) is opened,

a_i = supply at SP(i),

b_j = demand at DP(j),

x_k = amount shipped along the kth route from SP(s_k) to DP(d_k),
 y_i = 0 if SP(i) is closed; 1 if SP(i) is open.

Thus, the objective is to ship material from the supply points to the demand points, satisfying the demands at minimum cost.

Note that the total amount which can be shipped out of SP(i) is limited to a_i if y_i = 1 (SP(i) is open) and 0 if y_i = 0 (SP(i) is closed).

Problems of the form (PLN) are discussed in references [1] and [9] with p = m·n: i.e., shipment from any SP(i) to any DP(j) is possible. This report considers the more general case where the network of shipping routes is not complete due to many routes being nonexistent or prohibitively expensive in practice. For problems of the form (PLN) the condition $\sum_{i=1}^m a_i \geq \sum_{j=1}^n b_j$ by itself does not imply the existence of a feasible solution: for example if m = n and only shipment from SP(i) to DP(i) is allowed, i = 1, 2, ..., m, then any single a_i < b_i will make the problem infeasible.

It is assumed throughout this report that the data for (PLN) consists of nonnegative integers. If (PLN) has an optimal solution (x*, y*), it has one with x* integer valued, as long as the data is integer valued. This is because fixing y* to any 0-1 vector and solving (PLN) for x* results in a problem whose constraint matrix is totally unimodular [3].

In matrix notation, problem (PLN) becomes

$$\begin{aligned} & \text{Minimize} && cx + \varepsilon y \\ & \text{subject to} && Jx + Ay \geq 0 \\ & && Kx + Oy = b \\ & && x \geq 0, 0 \leq y \leq 1, y \text{ integer} \end{aligned}$$

where

$c \in R_+^p$ is the vector of shipping costs,

$\varepsilon \in R_+^m$ is the vector of fixed costs,

$b \in R_+^n$ is the vector of demands,

J is an $m \times p$ matrix with $J_{ik} = \begin{cases} -1 & \text{if } i=s_k, \\ 0 & \text{otherwise,} \end{cases}$

K is an $m \times p$ matrix with $K_{ik} = \begin{cases} +1 & \text{if } i=d_k, \\ 0 & \text{otherwise,} \end{cases}$

A is an $m \times n$ matrix with $A_{ij} = \begin{cases} a_i & \text{if } j=i, \\ 0 & \text{otherwise,} \end{cases}$

and 0 and 1 refer to vectors of appropriate dimension whose entries are all 0 and 1, respectively.

Upper bounds on x are not given explicitly but an upper bound on x_k in any feasible solution is given by $\text{Min.}(a_i, b_k, d_k)$ as can be seen from the nonnegativity of the data and the constraints

$$\sum_{s_k=i} x_k + a_i y_i \geq 0 \text{ and } \sum_{d_k=j} x_k = b_j.$$

1.2 Basic Idea

The sufficient optimality conditions derived in [2] form the basis for construction of integer programs with known optimal solutions.

The data comprising (PLN) will be generated in such a manner that the optimality conditions will be satisfied at a solution (x^*, y^*) , which is also generated. Alternate optima are possible, however, the optimal value of (PLN) is $cx^* + \varepsilon y^*$, so it is possible to determine how close to optimality a feasible solution obtained from a code actually is.

The sufficient optimality criterion, which is derived in reference [2], is stated below for problems of the form (PLN).

(SOC) Sufficient Optimality Criteria:

Let $x^* \in R_+^p, y^* \in R_+^n$ be a 0-1 vector, $t^* \in R_+^m$,

$c^{(\lambda)}, \varepsilon^{(\lambda)}$ be integer valued vectors, $\lambda = 1, 2, \dots, q$,

$u^{(\lambda)}, v^{(\lambda)}, w^{(\lambda)} \in R_+^m, \lambda = 1, 2, \dots, q$,

$\hat{u}^{(\lambda)} \in R_+^n, \lambda = 1, 2, \dots, q$,

$\hat{v}^{(\lambda)} \in R_+^p, \lambda = 1, 2, \dots, q$,

and

$\lambda_\ell \geq 0, \lambda = 1, 2, \dots, q$.

If

$$(1) \quad c_k^{(\ell)} = \hat{u}_k^{(\ell)} - u_k^{(\ell)} + \hat{v}_k^{(\ell)}, \quad k = 1, 2, \dots, p, \quad \ell = 1, 2, \dots, q,$$

$$(2) \quad g_i^{(\ell)} = a_i u_i^{(\ell)} + v_i^{(\ell)} - w_i^{(\ell)}, \quad i = 1, 2, \dots, m, \quad \ell = 1, 2, \dots, q,$$

$$(3) \quad c_k = \sum_{\ell=1}^q \lambda_{\ell} c_k^{(\ell)}, \quad k = 1, 2, \dots, p,$$

$$(4) \quad g_i = \sum_{\ell=1}^q \lambda_{\ell} g_i^{(\ell)}, \quad i = 1, 2, \dots, m,$$

$$(5) \quad - \sum_{s_k=i} x_k^* + a_i v_i^* - t_i^* = 0, \quad i = 1, 2, \dots, m,$$

$$(6) \quad \sum_{d_k=j} x_k^* = b_j, \quad j = 1, 2, \dots, n,$$

$$(7) \quad c^{(\ell)} = 0 \Rightarrow \delta_{\ell} = \sum_{i=1}^m t_i u_i^{(\ell)} + \sum_{k=1}^p x_k^* \hat{v}_k^{(\ell)}$$

$$+ \sum_{i=1}^m y_i v_i^{(\ell)} + \sum_{i=1}^m (1 - y_i) w_i^{(\ell)} < \gamma_{\ell},$$

where $\gamma_{\ell} = \text{gcd}(g_1^{(\ell)}, g_2^{(\ell)}, \dots, g_m^{(\ell)})$,

$$(8) \quad c^{(\ell)} \neq 0 \Rightarrow \sum_{i=1}^m t_i u_i^{(\ell)} + \sum_{k=1}^p x_k^* \hat{v}_k^{(\ell)} + \sum_{i=1}^m y_i v_i^{(\ell)}$$

$$+ \sum_{i=1}^m (1 - y_i) w_i^{(\ell)} = 0,$$

then $(x, y)^*$ solves (PLN).

[†] Greatest common divisor. A generalized greatest common divisor, applicable when the arguments are rational numbers, is described in reference [2] which also lists properties of the generalized greatest common divisor.

Relations (1) and (2) correspond to dual feasibility, (3) and (4) correspond to composition, (5) and (6) correspond to primal feasibility, (7) corresponds to quasicomplementarity, and (8) corresponds to complementarity. If $c^{(\ell)} = 0$, the ℓ^{th} component

will be referred to as an integer component, otherwise, it will be referred to as a continuous component. The quantity δ_{ℓ} in (7) will be referred to as the index of quasicomplementarity for the ℓ^{th} component, and for integer components, γ_{ℓ} in (7) will be referred

to the critical index for the ℓ^{th} component. In addition the

$$\text{quantities } \sum_{i=1}^m t_i u_i^* \text{ and } \sum_{k=1}^p x_k^* \hat{v}_k^{(\ell)} + \sum_{i=1}^m y_i v_i^* \text{ and } \sum_{i=1}^m (1 - y_i) w_i^* \text{ will be referred to respectively as the } \underline{\text{slack}}$$

quasicomplementary index and the solution quasicomplementary index for the ℓ^{th} component. Finally, the λ_{ℓ} 's will be referred to as component weights, $u^{(\ell)}$ and $\hat{v}^{(\ell)}$ will be referred to as u-multipliers for supply and demand rows, $v^{(\ell)}$ and $\hat{w}^{(\ell)}$ will be referred to as w-multipliers for integer and continuous columns, and $\hat{w}^{(\ell)}$ will be referred to as w-multipliers for integer columns. There are no w-multipliers

for continuous columns because there are no explicit upper bounds on x^* , although they could be included using the implicit upper

bounds of $x_k \leq \text{Min.}(a_s, b_{d_k})$.

The existence of test problems of the form (PLN) such that (SOC) holds at a solution $(x, y)^*$ and $(x, y)^*$ does not solve the

continuous relaxation of (PLN) is established in the following

theorem:

Theorem 1:

Given the data m, n, p, y^* and s and d describing the network for a problem of the form (PLN), if there exists an index r such that $y_r^* = 1$ and $s_k = r$ for some $1 \leq k \leq p$ then x, a, b, c , and g can be generated for (PLN) such that (SOC) holds at (x, y) and (x, y) does not solve the continuous relaxation of (PLN).

Proof:

Set $x_k^* = 2y_r^* s_k$, $k = 1, 2, \dots, p$,

$$a_i = \begin{cases} \sum_{k=i}^k x_k^* + 1 & \text{if } i=r \\ \sum_{k=i}^k x_k^* & \text{if } i \neq r, \end{cases}$$

$$t_i = \begin{cases} 1 & \text{if } i=r \\ 0 & \text{if } i \neq r, \end{cases}$$

$$b_j = \sum_{k=j}^n x_k^*, \quad j = 1, 2, \dots, n,$$

$$c_k = 1, \quad k = 1, 2, \dots, p,$$

$$g_i = \begin{cases} a_i - 1 & \text{if } i=r \\ a_i & \text{if } i \neq r \end{cases}$$

With $q = 2$ (2 components), set $u^{(\ell)}, \hat{u}^{(\ell)}, v^{(\ell)}, \hat{v}^{(\ell)}$, $w^{(\ell)}, c^{(\ell)}, g^{(\ell)}$, and λ_ℓ as follows:

ℓ	1	2
$u^{(\ell)}$	1 (vector of ones)	0
$\hat{u}^{(\ell)}$	1	1
$v^{(\ell)}$	0	0
$\hat{v}^{(\ell)}$	0	0
$w^{(\ell)}$	$w_i^{(\ell)} = \begin{cases} 1 & \text{if } i=r \\ 0 & \text{if } i \neq r \end{cases}$	0
$c^{(\ell)}$	0	1
$g^{(\ell)}$	g	0
λ_ℓ	1	1

With component 1 as an integer component and component 2 as a continuous component, we have $a_i \bmod 2 = \begin{cases} 1 & \text{if } i=r \\ 0 & \text{if } i \neq r \end{cases}$ and hence,

$$g_i \equiv 0 \pmod{2}, \quad i = 1, 2, \dots, m, \text{ so that } \gamma_1 = \gcd(g_1^{(1)}, g_2^{(1)}, \dots, g_m^{(1)})$$

≥ 2 . The index of quasicomplementarity in the first component

is $\delta_1 = t_r^* + 1 - y_r^* = 1 < 2$ so that (7) of (SOC) holds. Verification that the remaining conditions of (SOC) hold is straightforward.

To show (x, y) is not optimal for the continuous relaxation of (PLN), consider the solution (x^0, y^0) where $x^0 = x$

$$\text{and } y_i^0 = \begin{cases} \frac{a_r - 1}{a_r} & \text{if } i=r \\ y_i^* & \text{if } i \neq r \end{cases}. \text{ The assumption that } y_r^* = 1 \text{ and}$$

$s_k = r$ for some k guarantees $a_r \geq 3$ and $g_r \geq 2$.

The solution (x^0, y^0) is feasible for the continuous relaxation of (PLN) since

$$- \sum_{s_k=1} x_k^0 + a_i y_i^0 = \sum_{s_k=i} x_k^* + a_i y_i^* \geq 0, \quad i = 1, 2, \dots, m \quad (i \neq r)$$

$$\sum_{d_k=j} x_k^0 = \sum_{d_k=j} x_k^* = b_j, \quad j = 1, 2, \dots, n,$$

$$\text{and } - \sum_{s_k=r} x_k^0 + a_r y_r^0 = \sum_{s_k=r} x_k^* + (a_r - 1) = 0.$$

The solution (x^0, y^0) yields a lower objective value than (x^*, y^*) since $cx^* + \epsilon y^* - cx^0 - cy^0 = g_r \left(\frac{1}{a_r}\right) > 0$. ■

The test problem generator to be described in Section 2 is a procedure for generating data and other quantities appearing in (SOC) for problems of the form (PLN) such that (SOC) holds at a solution vector (x^*, y^*) . The quantities y^* and $s = (s_1, s_2, \dots, s_p)$ and $d = (d_1, d_2, \dots, d_p)$ describing the network of routes are specified at the beginning. Two components will be generated: the first will be an integer component ($c^{(1)} = 0$) and the second will be a continuous component. For the first component, the quantities $t^*, v^{(1)}, w^{(1)}, a, g^{(1)}, x$, and b are generated in such a manner that (5), (6), (7) and (1) and (2) for $\ell = 1$ will hold. For the second component, the quantities $u^{(2)}, \hat{u}^{(2)}, g^{(2)}$,

$c^{(2)}, \lambda_1$, and c will be generated in such a manner that (3), (4), (8) and (1) and (2) for $\ell = 2$ will hold. The generator will set all $u^{(1)}$ and $\hat{u}^{(1)}$ values to u_0 (a specified parameter), $\hat{v}^{(1)}, \hat{v}^{(2)}, v^{(2)}, w^{(2)}$ to 0, and $\lambda_2 = 1$. With (SOC) satisfied, it is guaranteed that (x^*, y^*) is optimal for (PLN).

2. GENERATION OF TEST PROBLEMS

2.1. Properties of Test Problem Generators

The generator to be described guarantees that the conditions (SOC) are satisfied. In addition, the following properties have also been considered:

(9) The data for (PLN) should be integer valued. This will eliminate rounding errors in the data which could possibly result in the generation of a test problem with the conditions in (SOC) not satisfied and (x^*, y^*) being non-optimal. Furthermore, source data for test problems generally requires less space when the data values are all integer valued.

(10) The test problem generator should be able to produce problems which are fairly difficult to solve. In particular, the solution to the continuous relaxation of (PLN) should not solve (PLN).

(11) There should be at least one integer component and one continuous component in the generation of test problems of the form (PLN). The procedure to be described generates one integer component and one continuous component. If there are no integer components, (SOC) becomes the Kuhn-Tucker conditions for linear programming problems [4], and the solution to the continuous relaxation of (PLN) solves (PLN). If there are no continuous components, $c = 0$ must hold: i.e., no shipping costs along the routes.

2.2 Procedure for Generating Test Problems

The following procedure is a systematic way of generating data and other quantities appearing in (SOC) for problems of the form (PLN) such that (SOC) holds at a solution vector (x^*, y^*) . All data generated for (PLN) will be integer valued.

In what follows, Latin letters will refer to quantities which assume integer values and Greek letters will refer to quantities which may assume any real values. The expression $e(\alpha)$ will denote $\lfloor \alpha + 0.5 \rfloor$ where $\lfloor x \rfloor$ denotes the largest integer which does not exceed x : i.e., $e(\alpha)$ is the value of α "rounded" to the nearest integer.

Input Parameters

- m = number of supply points; $m \geq 1$.
- n = number of demand points; $n \geq 1$.
- p = number of shipping routes; $\max(m,n) \leq p \leq m \cdot n$.

⁺s, d = two vectors of length p used to specify the shipping routes. The kth route is from supply point s_k to demand point d_k.

⁺y* = binary vector which will be part of the solution vector, (x*, y*).

a₀ = parameter specifying approximate expected value for each supply, a₁, generated; $a_0 \geq 1$.

c₀ = parameter specifying approximate expected value for each shipping cost, c_k, generated; $c_0 \geq 2$.

k₀ = parameter specifying approximate percent of total cost to be borne by the fixed cost in the solution (x*, y*): i.e., $\sum y_i^* (cx_i^* + \sum y_j^*) / 100$; $1 \leq k_0 \leq 99$.

t₀ = parameter specifying the maximum number of slack supply rows permitted for the solution (x*, y*). The actual number of supply rows cannot exceed $\sum_{i=1}^m y_i$; $0 \leq t_0 \leq m$.

α = ratio of the index of quasicomplementarity to its maximum allowed value for the integer (first) component generated; $0 \leq \alpha \leq 1$ (larger values of α tend to generate more difficult test problems).

⁺s, d, and y* may be input explicitly or generated according to prespecified parameters. See the Appendix. They should be specified or generated in such a manner that a positive demand can be satisfied at each demand point with the given network of routes and open supply points.

β = maximum ratio of the slack quasicomplementarity index to the total index; $(0 \leq \beta \leq 1)$ (for a fixed α , larger values of β tend to result in more slack in the supply rows and less variety in the values of the supplies generated. If $t_0 = 0$, this parameter has no effect).

u₀ = values of the u-multipliers used for the integer (first) component, all u⁽¹⁾ multipliers will be u₀; (u₀ \geq 1) (larger values tend to result in a greater variety of supply values to be generated at the expense of smaller gaps between the optimal objective values of (PLN) and its continuous relaxation. It is recommended that u₀ \leq 5.).

Procedure P2: (see Figure 1)

Two components will be generated. Steps 1-7 generate the first (integer) component as well as t^* , $v^{(1)}$, $w^{(1)}$, a , $g^{(1)}$, x^* , and b , and steps 8-12 generate the second (continuous) component as well as $u^{(2)}$, $\hat{u}^{(2)}$, $g^{(2)}$, $c^{(2)}$, λ_1 , and c . Quantities not explicitly generated are $u^{(1)}$ and $\hat{u}^{(1)}$ (all values set to u_0), $\hat{v}^{(1)}$, $v^{(2)}$, $\hat{v}^{(2)}$, $w^{(2)}$, (set to 0) and λ_2 (set to 1).

1. [Determine critical index for the first component.]

$$\text{Set } \hat{G} = e\left(\frac{2 \cdot u \cdot a}{u_0 + 1}\right) \text{ and } G = u_0 \left[\frac{G}{u}\right] + 1 \text{ (G will be the}$$

critical index for the first component. It is selected so that the expected value of a_i generated in step 5 will be approximately a_0 and $\text{gcd}(u_0, G) = 1$, which is necessary for the generation of a and $g^{(1)}$ in step 5).

2. [Determine amounts of quasicomplementarity.]

Set $Q = e(\alpha \cdot (G-1))$ and $Q_1 = [\beta \cdot Q]$ (Q will be the index of quasicomplementarity for the first component and Q_1 will be an upper bound on the slack index of quasicomplementarity).

3. [Generate slack vector.]

Generate $t^* \geq 0$ such that $Q_2 = u_0 \sum_{i=1}^m t_i^* \leq Q_1$ and at most t_0 of the components in t are positive (See the

Appendix). Set $Q_3 = Q - Q_2$ (Q_3 will be the solution index of quasicomplementarity).

4. [Generate v and w multipliers for first component.]

Generate $v^{(1)}$ and $w^{(1)}$ such that

$$\sum_{i=1}^m v_i^* v_i^{(1)} + \sum_{i=1}^m (1 - v_i^*) w_i^{(1)} = Q_3 \text{ and}$$

$$0 \leq v_i^{(1)}, w_i^{(1)} \leq G - 1, i = 1, 2, \dots, m. \text{ See the Appendix.}$$

(This will make the index of quasicomplementarity for the first component equal to Q).

5. [Generate a and $g^{(1)}$]

Solve the system of equations $u_0 \cdot a_i + v_i^{(1)} - w_i^{(1)} = G \cdot \hat{g}_i^{(1)}$,

$$1 \leq \hat{g}_i^{(1)} = u_0, i = 1, 2, \dots, m, \text{ in integers. Since}$$

$\text{gcd}(u_0, G) = 1$, the system has a unique solution by the Chinese remainder theorem [6] (See the Appendix). Since $0 \leq v_i^{(1)}$,

$w_i^{(1)} \leq G - 1$, $G \cdot \hat{g}_i^{(1)} - v_i^{(1)} + w_i^{(1)} > 0$ for $\hat{g}_i^{(1)} \geq 1$ which guarantees that a_i will be positive. Set $g_i^{(1)} = G \cdot \hat{g}_i^{(1)}$,

$i = 1, 2, \dots, m$. Then $g_i^{(1)}$ is an integral multiple of G

so that $\text{gcd}(g_i^{(1)}, g_i^{(2)}, \dots, g_i^{(m)}) \geq G$. (Note that a value

of u_0 is being used for all $u^{(1)}$ and $\hat{u}^{(1)}$ multipliers.

This in conjunction with $\hat{v}^{(1)} = 0$ will make $c^{(1)} = 0$).

6. [Generate x^*]
 Generate values of $x \geq 0$ such that the supply constraints in (PLN) are satisfied: i.e.,

$$\sum_{s_k=i} x_k^* = s_i y_i^* - t_i^*, \quad i = 1, 2, \dots, m$$
 (See the Appendix).
7. [Generate b]
 Generate values of b such that the demand constraints in (PLN) are satisfied: i.e.,

$$\text{set } b_j = \sum_{d_k=j} x_k^*, \quad j = 1, 2, \dots, m.$$
8. [Generate u and \hat{u} multipliers for the second component]
 Generate values of $u^{(2)}$ and $\hat{u}^{(2)}$ such that $u_t^{(2)*} = 0$, and $\hat{u}_k^{(2)} \geq u_{s_k}^{(2)} \geq 0$, $k = 1, 2, \dots, p$. In addition, the expected value of $a \hat{u}_j^{(2)} - u_i^{(2)}$ combination should be approximately c_0 , to make the expected value of c , generated in step 10, approximately c_0 (See the Appendix).
9. [Generate $g^{(2)}$]
 Set $g_i^{(2)} = a_i u_i^{(2)}$, $i = 1, 2, \dots, m$.
 (Note that $v^{(2)} = w^{(2)} = 0$ is being used).

10. [Generate c]
 Set $c_k = \hat{u}_{d_k}^{(2)} - u_{s_k}^{(2)}$, $k = 1, 2, \dots, p$. Since $\hat{u}_{d_k}^{(2)} \geq u_{s_k}^{(2)}$, $c(k) \geq 0$.
 (Note $\hat{v}^{(2)} = 0$ and $\lambda_2 = 1$, $c^{(1)} = 0$ imply $c = c^{(2)}$).
11. [Generate g]
 Set $L_1 = \text{Max.}(1, e^{(\frac{.01 k_0 (cx + g^{(2)*} - g^{(2)*} y)}{(1 - .01 k_0) g^{(1)*} y})})$,

$$\lambda_1 = \frac{L_1}{G}$$
, and $g_i = \lambda_1 g_i^{(1)} + g_i^{(2)}$, $i = 1, 2, \dots, m$
 (equivalently, $g_i = L_1 \hat{g}_i^{(1)} + g_i^{(2)}$). This will make g integer valued, since $g_i^{(2)} \equiv 0 \pmod{G}$, and gY^* (fixed costs) approximately equal to k_0 percent of $cx + gY^*$ (total costs) if $L_1 > 1$.
12. [End of Generation]
 Set $z^* = cx^* + gY^*$, the optimal objective value. ■

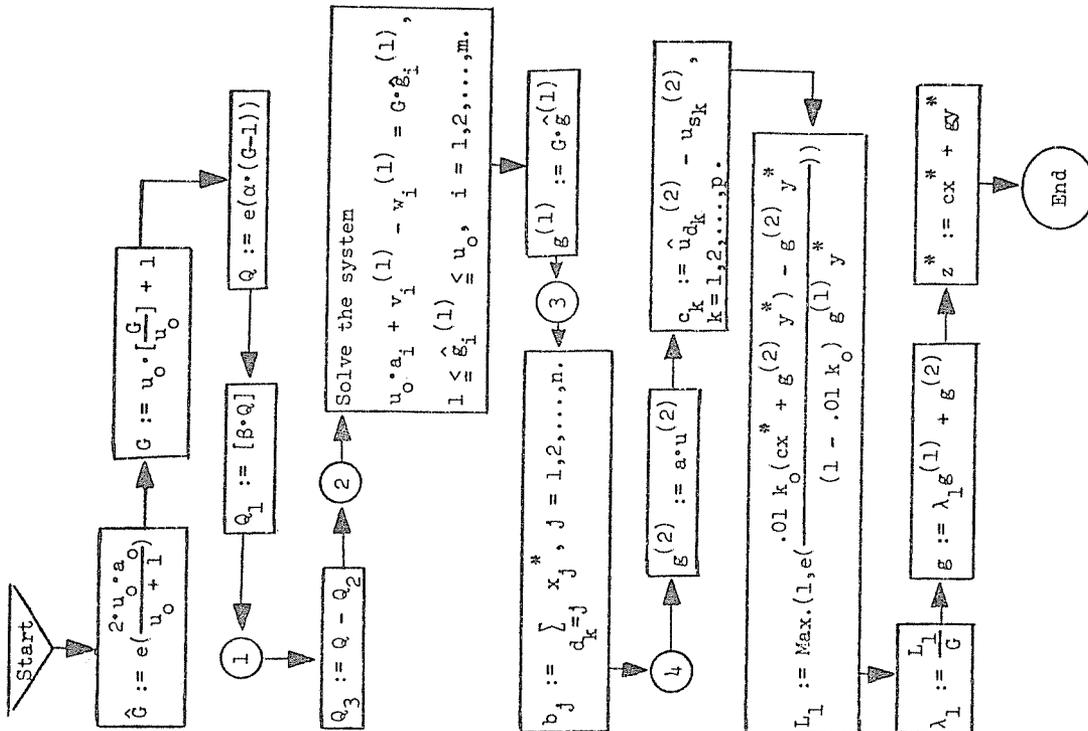


Figure 1: Flow Chart for the Test Problem Generator

Key to Procedures in Flowchart:

1. Generate $t^* \geq 0$ such that $Q_2 = u_0 \sum_{i=1}^m t_i^* \leq Q_1$ and at most t_0 of the components in t are positive.
2. Generate $v^{(1)}$ and $w^{(1)}$ such that $\sum_{i=1}^m y_i^* v_i^{(1)} + \sum_{i=1}^m (1 - y_i^*) w_i^{(1)} = Q_3$ and $0 \leq v_i^{(1)}, w_i^{(1)} \leq G - 1, i = 1, 2, \dots, m$.
3. Generate values of $x^* \geq 0$ such that $\sum_{s_k=i} x_k^* = a_i y_i^* - t_i^*, i = 1, 2, \dots, m$.
4. Generate $u^{(2)}$ and $\hat{u}^{(2)}$ such that $u^{(2)}_t^* = 0$ and $\hat{u}^{(2)}_d \geq u^{(2)}_s \geq 0, k = 1, 2, \dots, p$. In addition, the expected value of a $\hat{u}_j^{(2)} - u_{i_1}^{(2)}$ combination should be approximately c_0 .

2.3 Validity of Procedure

The procedure generates two components and implicitly sets all values of $u^{(1)}$ and $\hat{u}^{(1)}$ to $u_0, \hat{v}^{(1)}, v^{(2)}, \hat{v}^{(2)}, w^{(2)}$ to 0, and λ_2 to 1. Component 1 is an integer component and component 2 is a continuous component.

In steps 3 and 4, we generate $t, v^{(1)}$, and $w^{(1)}$ such that

$$(12) \quad u_0 \sum_{i=1}^m t_i + \sum_{i=1}^m v_i + \sum_{i=1}^m (1 - v_i) w_i = Q$$

where $Q < G$ (This can be seen by summing the quasicomplementarity relations in steps 3 and 4). In step 5, we generate a as well as $g^{(1)}$ such that

$$(13) \quad g_i^{(1)} = u_{o,i} + v_i^{(1)} - w_i^{(1)} = r_j G$$

where r_j is a nonnegative integer. Since all values of $u^{(1)}$ and $\hat{u}^{(1)}$ are u_0 and $v^{(1)} = 0$, (12) and (13) become conditions (7) and (2) of (SOC) for $l = 1$. Note $c^{(1)} = 0$ is consistent with (1) of (SOC). Thus, after step 5 of the procedure has been executed, conditions (1), (2), and (7) of (SOC) are satisfied for the first component.

In step 6, we generate x^* such that $\sum_{k=i}^m x_k^* = a_i x_i^* - t_i$, $i = 1, 2, \dots, m$, which is condition (5). In step 7, we generate b such that $b_j = \sum_{d=k}^m x_d^*$, which is condition (6). In step 8,

we generate $u^{(2)}$ and $\hat{u}^{(2)}$ multipliers such that $\sum_{i=1}^m t_i u_i^{(2)} = 0$.

and $u_i^{(2)} \geq 0$ which is condition (8) since $v^{(2)}, \hat{v}^{(2)}, w^{(2)}$ are 0. In step (9) we generate $g^{(2)}$ such that

$$(14) \quad g_i^{(2)} = a_i \cdot u_i^{(2)}, \quad i = 1, 2, \dots, m$$

and in step (10) we generate $c^{(2)}$ such that

$$(15) \quad c_k^{(2)} = \hat{u}_{d,k}^{(2)}, \quad k = 1, 2, \dots, p.$$

Since $v^{(2)} = w^{(2)} = 0$, (14) and (15) become (2) and (1) of (SOC) for $l = 2$. Since $\lambda_2 = 1$ and $c^{(1)} = 0$, $c = c^{(2)}$ implies (3) of (SOC). In step 11, we generate λ_2 and set $g_i^{(2)} = \lambda_1 g_i^{(1)} + g_i^{(2)}$, $i = 1, 2, \dots, m$, which is condition (4) since $\lambda_2 = 1$.

Thus, at the conclusion of the procedure, all of the conditions in (SOC) are satisfied.

From (SOC), an upper bound on the differential between the optimal objective values of (PLN) and its continuous relaxation is given by $\lambda_1 \delta_1$ since the gap between the optimal objective values of (PLN) (problem (PLN)) with $g = g^{(1)}$ and $c = c^{(1)} = 0$ is bounded above by δ_1 [2]. Since $\delta_k = e(\alpha \cdot (G-1))$ in Procedure P2, it is suggested that the parameter α should be set at or near

1 for constructing difficult test problems although a large gap needn't imply a difficult problem or vice-versa.

For a variety of test problems constructed using Procedure P2, the actual gap was usually greater than 50% of this upper bound (see Section 3). A sufficient condition for the gap to be posi-

tive and hence, for the solution to the continuous relaxation of (PLN) to not solve (PLN), is the existence of an index r such that $y_r^* = 1, t_r^* > 0$, and $g_r > 0$. In such a case, the solution

$$(x^0, y^0) \text{ where } x_i^0 = x^* \text{ and } y_i^0 = \begin{cases} (a_i - t_i^*)/a_i & \text{if } i=r \\ y_i^* & \text{if } i \neq r \end{cases}$$

is feasible for the continuous relaxation of (PLN) since

$$\sum_{s_k=i} x_k^0 + a_i y_i^0 = \sum_{s_k=i} x_k^* + a_i y_i^* \geq 0, \quad i = 1, 2, \dots, m \quad (i \neq r),$$

$$\sum_{d_k=j} x_k^0 = \sum_{d_k=j} x_k^* = b_j, \quad j = 1, 2, \dots, n,$$

$$\text{and } - \sum_{s_k=r} x_k^0 + a_r y_r^0 = \sum_{s_k=r} x_k^* + (a_r - t_r^*) = 0.$$

The solution (x^0, y^0) yields a lower objective value than (x^*, y^*) since

$$cx^* + gy^* - cx^0 - cy^0 = g_r \left(\frac{t_r^*}{a_r} \right) > 0.$$

From a physical standpoint, this corresponds to the case where costs could be reduced if plant r could be open "a fraction of the way with a commensurate operating cost" so as to just allow for the desired quantity of material to be shipped out of it.

3. COMPUTATIONAL EXPERIENCE

3.1 Generation of Test Problems

A variety of mixed integer programming problems of the form (PLN) were generated using Procedure P2 described in Section 2.2 and the Appendix. The parameters used in generating the problems are as described in Section 2.2 except that z_0 and y_0 , used to specify the generation of the network (s,d) and y^* , respectively, are described in the Appendix.

Table I

Parameters for Test Problems Generated

Problem	m	n	p	z_0	y_0	a_0	c	k_0	t_0	α	β	u_0
1	10	10	30	1	5	35	10	45	2	1.000	.200	1
2	15	10	30	1	6	70	15	35	0	1.000	.000	4
3	15	20	30	1	10	40	20	50	2	1.000	.300	3
4	20	20	40	1	10	50	15	33	1	1.000	.100	3
5	25	20	60	1	10	40	15	35	0	1.000	.000	1
6	25	30	60	1	16	65	20	45	1	1.000	.200	1
7	20	20	100	1	7	50	15	40	1	1.000	.300	4
8	20	20	100	1	10	60	20	50	3	1.000	.400	1
9	30	30	100	1	10	45	10	35	0	1.000	.000	2
10	30	30	100	1	15	55	15	45	2	1.000	.200	3

The test problems generated along with solutions are listed in the Appendix. Brief characteristics of the test problems generated are listed in Table II where m, n, and p are the same as in Table I,

Nodes = upper bound on the number of admissible assignments of values to the integer variables $y^* = 2^m$,
 z^* = the optimal objective value of the problem,
 z^0 = the optimal objective value of the corresponding continuous relaxation,
 Gap = $z^* - z^0$,
 Bound = a priori upper bound on $z^* - z^0$ based on results from the generator,
 % Gap = 100 times $(z^* - z^0)/z^*$,
 % Bound = 100 times $(z^* - z^0)/\text{Bound}$.

Table II
 Characteristics of Test Problems Generated

Problem	m	n	p	Nodes	z^*	z^0	Gap	Bound	% Gap	% Bound
1	10	10	30	1.02×10^3	3137	2930.22	206.78	233.33	6.59	88.6
2	15	10	30	3.28×10^4	13002	12883.20	118.80	185.35	0.91	64.1
3	15	20	30	3.28×10^4	17755	17610.33	144.67	337.38	0.81	42.9
4	20	20	40	1.05×10^6	12671	12557.99	113.01	142.11	0.89	79.5
5	25	20	60	3.36×10^7	9354	9178.79	175.21	237.07	1.87	73.9
6	25	30	60	3.36×10^7	37458	36930.36	527.64	861.74	1.41	61.2
7	20	20	100	1.05×10^6	11153	11028.92	124.08	178.77	1.11	69.4
8	20	20	100	1.05×10^6	22777	21983.85	793.15	989.51	3.48	80.2
9	30	30	100	1.07×10^9	6128	6098.11	29.89	42.30	0.49	70.7
10	30	30	100	1.07×10^9	20602	20394.83	207.17	252.88	1.01	81.9

3.2 Testing the Problems

Problems 1-3 were run on the mixed integer programming code IPMIXD available at the Madison Academic Computing Center using a Univac 1110 computer. IPMIXD is a linear programming based branch and bound algorithm based on the method of Land and Doig [5] for solving pure and mixed integer programming problems.

Problems 4-10 were run on the integer programming code IPDNUM, also available at the Madison Academic Computing center. IPDNUM is a pure integer programming code developed from ENUMER8 [7,8] which uses an implicit enumeration algorithm with an assortment of fathoming tests. It can be used to solve mixed integer programming problems of the form (PLN) by branching only on the integer variables, y, as long as the data is integer valued. The solution values for x will be integer valued because fixing y to any integer vector and solving (PLN) for x results in an integer solution.

Problems 4-10 were not run using IPMIXD because problem 4 (and, presumably, 5-10) require a prohibitive amount of core to store the intermediate tableaus. IPDNUM on the other hand, has the advantage of requiring a fixed amount of storage space to solve a problem.

Table III lists computational results where

Code = 1 for IPMIXD, 2 for IPDNUM,

Terminated = explanation on how run terminated,

Best Objective = objective value of best feasible solution obtained,

% Error = 100 (Best Objective - z*)/z*, where z* is the optimal objective value (see Table II),

Iterations = number of nodes explicitly analyzed,

Time = solution time in seconds,

Best Node # = node number at which best solution was found,

Incumbents = number of successive feasible solutions found with strictly decreasing objective value.

Table III
Computational Results

Problem	Code	Terminated	Best Objective	% Error	Iterations	Time	Best Node #	Incumbents
1	1	Optimal	3137	.00	242	12.53	142	4
2	1	Optimal	13002	.00	351	21.09	307	8
3	1	Optimal	17755	.00	62	5.06	45	2
4	2	Optimal	12671	.00	224	35.91	160	5
5	2	Time Limit	9640	3.06	1051	180.00+	591	8
6	2	Time Limit*	37458	.00	551	180.00+	270	7
7	2	Time Limit	11559	3.64	576	240.00+	552	4
8	2	Time Limit	22802	.11	326	240.00+	260	5
9	2	Time Limit	6226	1.60	401	240.00+	290	5
10	2	Time Limit	21623	4.96	301	240.00+	118	2

These results indicate that Procedure P2 can be used for constructing difficult test problems of the form (PLN). Even the moderate sized problems 5-10 might require examination of several thousand nodes to verify optimality although the IPMIXD (storage space permitting) and IPDNUM codes usually arrive at a reasonably good solution early in the search.

* Best incumbent is optimal, but optimality not verified within time limit.

Appendix

Procedures for Generating u-, v-, and w-Multipliers, x, and t

Listed here are some procedures required by Procedure P2 of Section 2.2 for generating u-, v-, and w-multipliers, x, and t. The same conventions used in Section 2.2 will apply here and urn(p,q) will denote a random integer uniformly distributed on {p, p+1, p+2, ..., q}.

A1. Generation of $t \geq 0$ such that $Q_2 = u_0 \sum_{i=1}^m t_i^* \leq Q_1$ and at most t_0 of the components in t are positive.

This procedure is used in step 3 of Procedure 2.

1. [Initializations]

Set $r = \text{Min.}(t_0, y_0)$ where $y_0 = \sum_{i=1}^m v_i^*$ (r is the number of supply rows which may have positive slack).

Set $t^* = 0, Q_2 = 0, j = 0$, and $\theta = \frac{2Q_1}{v_0 \cdot r \cdot (r+1)}$. For $i = 1, 2, \dots, m$ do steps 2 and 3.

2. [Set up j^{th} slack row if $v_i^* = 1.$]

If $v_i^* = 0$, skip the remainder of this step as well as step 3.

3. Otherwise, set $j = j+1, t_i^* = [\theta \cdot j]$, and $Q_2 = Q_2 + u_0 t_i^*$.

3. [All slack rows generated?]

If $j = r$, stop.

This procedure generates r slack values in the approximate ratios of $1:2:\dots:r$ and at the conclusion of the procedure,

$$Q_2 = u_0 \sum_{j=1}^r [0j] \leq u_0 \sum_{j=1}^r \theta j = u_0 \theta \frac{r(r+1)}{2} = Q_1.$$

A2. Generation of $v^{(1)}$ and $w^{(1)}$ such that

$$\sum_{i=1}^m v_i^*(1) + \sum_{i=1}^m (1-v_i^*)w_i^*(1) = Q_3 \text{ and}$$

$$0 \leq v_i^{(1)}, w_i^{(1)} \leq G-1, i = 1, 2, \dots, m.$$

This procedure is used in step 4 of Procedure P2.

1. [Initializations]

Set $Q = 0, M_0 = 0$, and $M_1 = 0$ (The pointers $M_0(M_1)$ will be 1 for the 1st, 3rd, 5th, ..., (2k+1)th open (closed) plant and 0 otherwise. One of $v_i^{(1)}$ and $w_i^{(1)}$ may be nonzero and contribution to quasicomplementarity will be made on the 1st, 3rd, 5th, ..., (2k+1)th open (closed plant).

2. [Compute trial multipliers.]

For $i = 1, 2, \dots, m$ do steps 2.1-2.4.

2.1 [Open or closed plant?]

If $v_i^* = 0$ go to step 2.2; if $v_i^* = 1$ go to step 2.3.

2.2 [Kth closed plant: K odd or even?]

Set $M_0 = 1 - M_0$. If $M_0 = 0$, set $\hat{w}_i^{(1)} = 0$,

$\hat{v}_i^{(1)} = \text{urn}(1, Q_3)$, and go to step 2.4; if $M_0 = 1$, set

$\hat{v}_i^{(1)} = 0, \hat{w}_i^{(1)} = \text{urn}(1, Q_3), Q = Q + \hat{w}_i^{(1)}$, and go to step 2.4.

2.3 [Kth open plant: K odd or even?] Set $M_1 = 1 - M_1$. If

$M_1 = 0$, set $\hat{v}_i^{(1)} = 0, \hat{w}_i^{(1)} = \text{urn}(1, Q_3)$, and go to step 2.4;

if $M_1 = 1$, set $\hat{v}_i^{(1)} = 0$, $\hat{v}_i^{(1)} = \text{urn}(1, Q_3)$, $Q = Q + \hat{v}_i^{(1)}$, and go to step 2.4.

2.4 [End of Loop]

3. [Scale the trial multipliers]

Set $v_j^{(1)} = \text{Max.}(\frac{Q_3}{Q} \hat{v}_j^{(1)}], G-1), j = 1, 2, \dots, n$,

and $w_j^{(1)} = \text{Max.}(\frac{Q_3}{Q} \hat{w}_j^{(1)}], G-1), j = 1, 2, \dots, n$.

(This guarantees $y v^{*(1)} + \dagger(1-y)w^{*(1)} \leq Q_3$.)

4. [Increase quasicomplementarity to Q_3]

Set $Q = y v^{*(1)} + (1-y)w^{*(1)}$, $M_0 = 0$, and $M_1 = 0$.

For $i = 1, 2, \dots, m$ do steps 4.1-4.5.

4.1 [Does quasicomplementarity equal Q_3 ?]

If $Q = Q_3$ stop, otherwise, go to step 4.2.

4.2 [Open or closed plant?]

If $y_i^* = 0$ go to step 4.3; if $y_i^* = 1$ go to step 4.4.

4.3 [K^{th} closed plant: K odd or even?]

Set $M_0 = 1 - M_0$. If $M_0 = 0$, set $v_i^{(1)} = \text{Max.}(v_i^{(1)} + 1, G-1)$

and go to step 4.5; if $M_0 = 1$, set $w_i^{(1)} = w_i^{(1)} + 1$, $Q =$

$Q + 1$, and go to step 4.5.

4.4 [K^{th} open plant: K odd or even?]

Set $M_1 = 1 - M_1$. If $M_1 = 0$, set $w_i^{(1)} = \text{Max.}(w_i^{(1)} + 1, G-1)$

and go to step 4.5; if $M_1 = 1$, set $v_i^{(1)} = v_i^{(1)} + 1$,

[†] 1 denotes a vector of all ones.

$Q = Q + 1$, and go to step 4.5.

4.5 [End of Loop]

In step 2 we generate trial multipliers $\hat{u}_i^{(1)}$ and $\hat{v}_i^{(1)}$ such that one of $\hat{u}_i^{(1)}$, $\hat{v}_i^{(1)}$ is zero and the other is positive. The multiplier $\hat{v}_i^{(1)}$ ($\hat{w}_i^{(1)}$) is positive for the 1st, 3rd, 5th, ..., (2K+1)th open (closed) plant: i.e., $y_i^* = 1$ ($y_i^* = 0$). Thus, the terms in

$$Q = \sum_{i=1}^m y_i^* \hat{v}_i^{(1)} + \sum_{i=1}^m (1-y_i^*) \hat{w}_i^{(1)}$$

are positive for the

1st, 3rd, 5th, ..., (2K+1)th open (closed) plant.

In step 3, we scale the multipliers so that

$$(A1) \quad Q_3 - f < \sum_{i=1}^m y_i^* v_i^{*(1)} + \sum_{i=1}^m (1-y_i^*) w_i^{*(1)} \leq Q_3$$

where f is the number of positive terms in the sum in (A1).

A slack of less than f units may occur due to the truncation of $v^{(1)}$ and $w^{(1)}$ to integers in step 3. Note that since

$Q_3 \leq G-1$, the multipliers corresponding to positive terms

in the sum of (A1) must be less than G without explicitly limiting the multiplier values to $G-1$.

In step 4, we add one to the multipliers (designated as positive by the rule in step 2) until

$$\sum_{i=1}^m y_i^* v_i^{*(1)} + \sum_{i=1}^m (1-y_i^*) w_i^{*(1)} = Q_3.$$

This is always possible because of (A1) and increasing

$v_i^{(1)}$ or $w_i^{(1)}$ by one unit increases the quasicomplementarity by zero or one units.

A3. Generation of values of $x_k^* \geq 0$ such that $\sum_{s_k=i} x_k^* = a_i y_i^* - t_i$, $i = 1, 2, \dots, m$.

This procedure is used in step 6 of Procedure P2.

1. [Initializations]

Set $z_i, \hat{z}_i = 0$, $i = 1, 2, \dots, m$
(used to keep track of sums).

2. [Compute trial values]

For $k = 1, 2, \dots, p$ do steps 2.1-2.2.

2.1 [Determine supply index]

Set $i = s_k$.

2.2 [Generate trial values]

If $v_i^* = 0$, set $\hat{x}_k^* = 0$;

if $v_i^* = 1$, set $\hat{x}_k^* = \text{urn}(100, 300)$ and

$z_i = z_i + \hat{x}_k^*$.

3. [Scale the trial values]

3.1 [Compute scale factors]

For $i = 1, 2, \dots, m$ set

$z_i = 0$ if $z_i = 0$

$\theta_i = \begin{cases} a_i - t_i \\ z_i \end{cases}$ if $z_i \neq 0$

For $k = 1, 2, \dots, p$ do steps 3.2-3.3.

3.2 [Determine supply index]

Set $i = s_k$

3.3 [Scale]

If $v_i^* = 1$, set $x_k^* = [\theta_i \cdot x_k]$ and $\hat{z}_i = \hat{z}_i + x_k^*$.

(At the end of step 3, $\sum_{s_k=i} x_k^* < a_i y_i^* - t_i$, $i = 1, 2, \dots, m$.)

4. [Add 1 to components until supply constraints check]

For $k = 1, 2, \dots, p$ do steps 4.1-4.2

4.1 [Determine supply index]

Set $i = s_k$.

4.2 [Increase x_k^* by 1 if supply row doesn't check.]

If $v_i^* = 1$ and $\hat{z}_i < a_i - t_i$, set $x_k^* = x_k^* + 1$

and $\hat{z}_i = \hat{z}_i + 1$.

In step 2, we generate trial values \hat{x}_k^* which are 0 if

supply point s_k is closed and in the range 100-300 if supply

point s_k is open. It is desirable to have $x_k^* > 0$ if supply

point s_k is open to avoid the generation of zero demands

in step 7 of Procedure P2, as long as at least one route

exists from some open supply point into each demand point.

In step 3, we scale the values so that

$$(A2) \quad a_i y_i^* - t_i - f_i < \sum_{s_k=i} x_k^* \leq a_i y_i^* - t_i, \quad i = 1, 2, \dots, m$$

where f_i = the number of terms in the sum of (A2) (all quantities are zero if $V_i^* = 0$). If $V_i^* = 1$, a slack of less than f_i units may occur due to the truncation of x_k^* to integer values in step 3.

In step 4, we add one to values of x_k^* such that $y_{s_k}^* = 1$ until $\sum_{s_k=i} x_k^* = a_i V_i^* - t_i$, $i = 1, 2, \dots, m$.

This is always possible because of (A2) and increasing an x_k^* by one unit increases some $\sum_{s_k=i} x_k^*$ by one unit.

A4. Generation of $u^{(2)}$ and $\hat{u}^{(2)}$ such that $u^{(2)*} = 0$ and $\hat{u}^{(2)} \geq u^{(2)} \geq 0$, $k = 1, 2, \dots, p$. In addition the expected value of $-u_i^{(2)} + \hat{u}_j^{(2)}$ is approximately c_0 .

This procedure is used in step 8 of Procedure P2.

1. [Initializations]

Set $r_1 = 2c_0$, $r_2 = \lfloor \sqrt{r_1} \rfloor$, and $r_3 = r_2 - 1$ so that $r_1 > r_2 > r_3 > 0$, since $c_0 \geq 2$.

2. [Generate $u^{(2)}$]

For $i = 1, 2, \dots, m$, if $t_i^* > 0$, set $u_i^{(2)} = 0$, if $t_i^* = 0$, set $u_i^{(2)} = \text{urn}(1, r_3)$.

3. [Generate $\hat{u}^{(2)}$]

For $j = 1, 2, \dots, n$, set $u_j^{(2)} = \text{urn}(r_1, r_2)$. ■

Since $\hat{u}_j^{(2)} > u_i^{(2)} > 0$ for all values of i and j , $\hat{u}_d^{(2)} > u_{s_k}^{(2)} \geq 0$, and the expected value of $-u_i^{(2)} + \hat{u}_j^{(2)}$

is approximately c_0 , depending upon the number of slack rows where $u_i^{(2)} = 0$.

A5. Solution of the equation $qx + r = dy$ in integers with $1 < y < q$.

Step 5 of Procedure P2 requires the solution of

$$(A3) \quad \begin{cases} qx + r = dy \\ 1 < y < q \\ x, y \text{ integer} \end{cases}$$

for x and y in terms of q, r , and d (all quantities are scalars) in order to determine a and $g^{(1)}$ in (PLN). For $i = 1, 2, \dots, m$, q corresponds to u_0 , x to a_i , r to $v_i^{(1)} - w_i^{(1)}$, d to G , and y to $\hat{g}_i^{(1)}$. If $\text{gcd}(q, d) = 1$, then (A1) has a unique solution (\hat{x}, \hat{y}) by the Chinese remainder theorem (see reference [6]).

The following procedure can be used to solve (A3) assuming $\text{gcd}(q, d) = 1$. It is efficient as long as q is not very large.

Procedure for solving (A3) where $\text{gcd}(q, d) = 1$.

1. [Initialize]

Set $z = -r$ and $y = 0$

2. [Next value of y]

Set $y = y + 1$, $z = z + d$, and $x = \lfloor z/q \rfloor$.

3. [Is system satisfied?]

If $qx = z$, terminate, otherwise go to step 2. ■

Steps 2 and 3 are executed at most q times as long as $gcd(q,d) = 1$.

Extensions to the Test Problem Generator

When using Procedure P2 to generate test problems of the form (PLN), it is convenient to be able to specify a solution vector for the integer variables y^* , and the network of routes determined by s and d from among a standard set via single parameters, y_0 and z_0 . For the test problem generator used to generate the problems in Section 3, the parameters y_0 and z_0 were used as follows:

- $y_0 = 0$ - input the y^* vector explicitly.
- $1 \leq y_0 \leq m$ - set y_0 of the components of y^* to 1. Specifically, $y_i^* = 1$ for $i = e(\frac{dm}{m})$, $j = 1, 2, \dots, y_0$, and $y_i^* = 0$ otherwise.
- $z_0 = 0$ - input s and d explicitly
- $z_0 = 1$ - generate s and d according to the following procedure.

The following procedure generates s and d specifying a network with p routes with the following properties:

- 1) The routes are distributed among the supply points as equally as possible: i.e., the number of routes emanating from a supply point is $\lfloor \frac{D_j}{m} \rfloor$ or $\lfloor \frac{D_j}{m} \rfloor + 1$.

- 2) The demand point index assumes the values $1, 2, 3, \dots, n$; $1, 2, 3, \dots, n$;... among successive routes generated emanating from open supply points ($y_i^* = 1$).
 - 3) The demand point index assumes the values $1, n, n-1, \dots, 2, 1$; $n, n-1, \dots, 2, \dots$ among successive routes generated emanating from closed supply points ($y_i^* = 0$).
 - 4) The routes are effectively rearranged so that the pairs of indices $(s_1, d_1), (s_2, d_2), \dots, (s_p, d_p)$ are in lexicographic order.
- Then, a sufficient condition to have at least one route emanating from some open supply point to each of the demand points is $\sum_{m=1}^p y_0 \geq n$ where y_0 is the number of open supply points. If $x_k^* > 0$ when supply point s_k is open (see Procedure A3), this condition will ensure that no zero demands are generated in step 7 of Procedure (P2).

1. [Initialize]

- Set $M_j = 0$, $j = 1, 2, \dots, n$,
 - $q = \lfloor \frac{D}{m} \rfloor$,
 - $r = p - m \cdot q$,
 - $k = 0$,
 - $j_0 = 2$,
 - $j_1 = n$.
- For $i = 1, 2, \dots, m$, do steps 2-4.

2. [Determine number of routes out of supply points]

Set $f = \begin{cases} q & \text{if } i \geq r \\ q+1 & \text{if } i < r \end{cases}$

3. [Determine the demand points]

For $\ell = 1, 2, \dots, f$ do steps 3.1-3.4.

3.1 [Plant open or closed?]

If $y_i^* = 0$ go to step 3.2; if $y_i^* = 1$

go to step 3.3

3.2 [Next demand point for closed plants]

if $j_o = 1,$

Set $j_o = \begin{cases} n & \text{if } j_o = 1, \\ j_o - 1 & \text{if } j_o \neq 1. \end{cases}$

Set $M_{j_o} = i$ and go to step 3.4.

3.3 [Next demand point for open plants]

Set $j_1 = \begin{cases} 1 & \text{if } j_1 = n, \\ j_1 + 1 & \text{if } j_1 \neq n. \end{cases}$

Set $M_{j_1} = i$ and go to step 3.4

3.4 [End of loop]

4. [Set up s and d for routes emanating from supply point]

For $j = 1, 2, \dots, n,$

if $M_j = i,$ set $k = k+1, s_k = i,$ and $d_k = j.$

Data for Test Problems

The data generated for the problems described in Section 3.1 along with solutions appear in the following pages. The correspondence between headings and symbols used throughout the body of this report is as follows:

TOTAL COST - $cx + \epsilon y^*$

VARIABLE COST - cx^*

FIXED COST - ϵy^*

SUPPLY POINTS

S.P. - i (index of supply point)

CAPACITY - a_i

COST - ϵ_i

YSTAR - y_i^*

DEMAND POINTS

D.P. - j (index of demand point)

DEMAND - b_j

SHIPPING ROUTES

ROUTE - k (index of route)

FROM - s_k (source supply point)

TO - d_k (destination demand point)

COST - c_k

XSTAR - x_k^*

ROUTE	FROM	TO	COST	XSTAR
1	1	1	17	0
2	2	2	14	0
3	3	3	6	0
4	4	4	4	0
5	5	5	7	0
6	6	6	6	0
7	7	7	12	0
8	8	8	10	0
9	9	9	7	0
10	10	10	21	50
11	11	11	12	0
12	12	12	21	0
13	13	13	14	0
14	14	14	26	0
15	15	15	26	0
16	16	16	23	0
17	17	17	10	0
18	18	18	7	0
19	19	19	9	0
20	20	20	10	0
21	21	21	12	0
22	22	22	13	0
23	23	23	15	0
24	24	24	17	0
25	25	25	15	0
26	26	26	4	0
27	27	27	26	0
28	28	28	4	0

SHIPPING ROUTES				
ROUTE	FROM	TO	COST	XSTAR
1	1	1	17	0
2	2	2	14	0
3	3	3	6	0
4	4	4	4	0
5	5	5	7	0
6	6	6	6	0
7	7	7	12	0
8	8	8	10	0
9	9	9	7	0
10	10	10	21	50
11	11	11	12	0
12	12	12	21	0
13	13	13	14	0
14	14	14	26	0
15	15	15	26	0
16	16	16	23	0
17	17	17	10	0
18	18	18	7	0
19	19	19	9	0
20	20	20	10	0
21	21	21	12	0
22	22	22	13	0
23	23	23	15	0
24	24	24	17	0
25	25	25	15	0
26	26	26	4	0
27	27	27	26	0
28	28	28	4	0

DEMAND POINTS				
D.P.	DEMAND	D.P.	DEMAND	D.P.
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28

SUPPLY POINTS				
S.P.	CAPACITY	S.P.	CAPACITY	S.P.
1	15	1	3	1
2	119	2	3	1
3	52	3	3	1
4	90	4	3	1
5	119	5	3	1
6	52	6	3	1
7	90	7	3	1
8	119	8	3	1
9	52	9	3	1
10	90	10	3	1
11	119	11	3	1
12	52	12	3	1
13	90	13	3	1
14	119	14	3	1
15	52	15	3	1
16	90	16	3	1
17	119	17	3	1
18	52	18	3	1
19	90	19	3	1
20	119	20	3	1
21	52	21	3	1
22	90	22	3	1
23	119	23	3	1
24	52	24	3	1
25	90	25	3	1
26	119	26	3	1
27	52	27	3	1
28	90	28	3	1

SHIPPING ROUTES				
ROUTE	FROM	TO	COST	XSTAR
1	1	1	17	0
2	2	2	14	0
3	3	3	6	0
4	4	4	4	0
5	5	5	7	0
6	6	6	6	0
7	7	7	12	0
8	8	8	10	0
9	9	9	7	0
10	10	10	21	50
11	11	11	12	0
12	12	12	21	0
13	13	13	14	0
14	14	14	26	0
15	15	15	26	0
16	16	16	23	0
17	17	17	10	0
18	18	18	7	0
19	19	19	9	0
20	20	20	10	0
21	21	21	12	0
22	22	22	13	0
23	23	23	15	0
24	24	24	17	0
25	25	25	15	0
26	26	26	4	0
27	27	27	26	0
28	28	28	4	0

DEMAND POINTS				
D.P.	DEMAND	D.P.	DEMAND	D.P.
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28

SUPPLY POINTS				
S.P.	CAPACITY	S.P.	CAPACITY	S.P.
1	15	1	3	1
2	119	2	3	1
3	52	3	3	1
4	90	4	3	1
5	119	5	3	1
6	52	6	3	1
7	90	7	3	1
8	119	8	3	1
9	52	9	3	1
10	90	10	3	1
11	119	11	3	1
12	52	12	3	1
13	90	13	3	1
14	119	14	3	1
15	52	15	3	1
16	90	16	3	1
17	119	17	3	1
18	52	18	3	1
19	90	19	3	1
20	119	20	3	1
21	52	21	3	1
22	90	22	3	1
23	119	23	3	1
24	52	24	3	1
25	90	25	3	1
26	119	26	3	1
27	52	27	3	1
28	90	28	3	1

PROBLEM NO. 1
TOTAL COST = 3137
VARIABLE COST = 1725
FIXED COST = 1412

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

PROBLEM NO. 2
TOTAL COST = 13002
VARIABLE COST = 8448
FIXED COST = 4554

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

SUPPLY POINTS									
S.P.	CAPACITY	COST	XSTAR	D.P.	DEMAND	S.P.	CAPACITY	COST	XSTAR
1	27	225	0	2	169	1	27	225	0
2	27	225	0	3	590	2	27	225	0
3	27	225	0	4	657	3	27	225	0
4	27	225	0	5	79	4	27	225	0
5	27	225	0	6	75	5	27	225	0
6	27	225	0	7	47	6	27	225	0
7	27	225	0	8	53	7	27	225	0
8	27	225	0	9	76	8	27	225	0
9	27	225	0	10	25	9	27	225	0
10	27	225	0	11	16	10	27	225	0
11	27	225	0	12	17	11	27	225	0
12	27	225	0	13	16	12	27	225	0
13	27	225	0	14	15	13	27	225	0
14	27	225	0	15	14	14	27	225	0
15	27	225	0	16	13	15	27	225	0
16	27	225	0	17	12	16	27	225	0
17	27	225	0	18	11	17	27	225	0
18	27	225	0	19	10	18	27	225	0
19	27	225	0	20	9	19	27	225	0
20	27	225	0	21	8	20	27	225	0
21	27	225	0	22	7	21	27	225	0
22	27	225	0	23	6	22	27	225	0
23	27	225	0	24	5	23	27	225	0
24	27	225	0	25	4	24	27	225	0
25	27	225	0	26	3	25	27	225	0
26	27	225	0	27	2	26	27	225	0
27	27	225	0	28	1	27	27	225	0
28	27	225	0	29	0	28	27	225	0
29	27	225	0	30	0	29	27	225	0
30	27	225	0	31	0	30	27	225	0
31	27	225	0	32	0	31	27	225	0
32	27	225	0	33	0	32	27	225	0
33	27	225	0	34	0	33	27	225	0
34	27	225	0	35	0	34	27	225	0
35	27	225	0	36	0	35	27	225	0
36	27	225	0	37	0	36	27	225	0
37	27	225	0	38	0	37	27	225	0
38	27	225	0	39	0	38	27	225	0
39	27	225	0	40	0	39	27	225	0

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

TOTAL COST =									
PROBLEM NO.	4	12671	8485	4186					
TOTAL COST	4186	12671	8485	4186					
VARIABLE COST	4186	12671	8485	4186					
FIXED COST	0	0	0	0					

SUPPLY POINTS									
S.P.	CAPACITY	COST	XSTAR	D.P.	DEMAND	S.P.	CAPACITY	COST	XSTAR
1	1157	1157	0	2	1349	1	1157	1157	0
2	1157	1157	0	3	369	2	1157	1157	0
3	1157	1157	0	4	1157	3	1157	1157	0
4	1157	1157	0	5	66	4	1157	1157	0
5	1157	1157	0	6	66	5	1157	1157	0
6	1157	1157	0	7	66	6	1157	1157	0
7	1157	1157	0	8	66	7	1157	1157	0
8	1157	1157	0	9	66	8	1157	1157	0
9	1157	1157	0	10	66	9	1157	1157	0
10	1157	1157	0	11	66	10	1157	1157	0
11	1157	1157	0	12	66	11	1157	1157	0
12	1157	1157	0	13	66	12	1157	1157	0
13	1157	1157	0	14	66	13	1157	1157	0
14	1157	1157	0	15	66	14	1157	1157	0
15	1157	1157	0	16	66	15	1157	1157	0
16	1157	1157	0	17	66	16	1157	1157	0
17	1157	1157	0	18	66	17	1157	1157	0
18	1157	1157	0	19	66	18	1157	1157	0
19	1157	1157	0	20	66	19	1157	1157	0
20	1157	1157	0	21	66	20	1157	1157	0
21	1157	1157	0	22	66	21	1157	1157	0
22	1157	1157	0	23	66	22	1157	1157	0
23	1157	1157	0	24	66	23	1157	1157	0
24	1157	1157	0	25	66	24	1157	1157	0
25	1157	1157	0	26	66	25	1157	1157	0
26	1157	1157	0	27	66	26	1157	1157	0
27	1157	1157	0	28	66	27	1157	1157	0
28	1157	1157	0	29	66	28	1157	1157	0
29	1157	1157	0	30	66	29	1157	1157	0
30	1157	1157	0	31	66	30	1157	1157	0
31	1157	1157	0	32	66	31	1157	1157	0
32	1157	1157	0	33	66	32	1157	1157	0
33	1157	1157	0	34	66	33	1157	1157	0
34	1157	1157	0	35	66	34	1157	1157	0
35	1157	1157	0	36	66	35	1157	1157	0
36	1157	1157	0	37	66	36	1157	1157	0
37	1157	1157	0	38	66	37	1157	1157	0

PROBLEM NO. 7 TOTAL COST = 11153 VARIABLE COST = 6697 FIXED COST = 4456

SUPPLY POINTS		DEMAND POINTS	
S.P.	CAPACITY	D.P.	DEMAND
1	42	1	1
4	82	2	12
7	62	2	12
10	63	3	3
13	61	8	16
16	83	14	14
19	61	2	2
		20	20
		D.P.	D.P.
		2	6
		12	18
		23	19
		31	
		DEMAND	DEMAND

SHIPPING ROUTES		DEMAND POINTS		SUPPLY POINTS	
ROUTE	COST	D.P.	DEMAND	S.P.	CAPACITY
1	1	1	1	1	42
19	19	2	12	4	82
27	27	2	12	7	62
10	10	3	3	10	63
3	3	8	16	13	61
7	7	14	14	16	83
19	19	2	2	19	61
20	20	20	20		
		D.P.	D.P.		
		2	6		
		12	18		
		23	19		
		31			
		DEMAND	DEMAND		

PROBLEM NO. 8 TOTAL COST = 22777 VARIABLE COST = 11389 FIXED COST = 11388

SUPPLY POINTS		DEMAND POINTS	
S.P. CAPACITY	55	D.P. DEMAND	44
COST	1006	D.P. DEMAND	33
YSTAR	1	D.P. DEMAND	37
S.P. CAPACITY	55	D.P. DEMAND	4
COST	1276	D.P. DEMAND	10
YSTAR	1	D.P. DEMAND	16
S.P. CAPACITY	55	D.P. DEMAND	9
COST	1059	D.P. DEMAND	3
YSTAR	0	D.P. DEMAND	36
S.P. CAPACITY	68	D.P. DEMAND	19
COST	1246	D.P. DEMAND	8
YSTAR	0	D.P. DEMAND	19
S.P. CAPACITY	66	D.P. DEMAND	14
COST	1276	D.P. DEMAND	20
YSTAR	1	D.P. DEMAND	14
S.P. CAPACITY	54	D.P. DEMAND	6
COST	1250	D.P. DEMAND	18
YSTAR	0	D.P. DEMAND	12
S.P. CAPACITY	64	D.P. DEMAND	44
COST	1336	D.P. DEMAND	18
YSTAR	1	D.P. DEMAND	33
S.P. CAPACITY	54	D.P. DEMAND	6
COST	1276	D.P. DEMAND	18
YSTAR	1	D.P. DEMAND	12
S.P. CAPACITY	53	D.P. DEMAND	44
COST	1059	D.P. DEMAND	18
YSTAR	0	D.P. DEMAND	33
S.P. CAPACITY	54	D.P. DEMAND	6
COST	1276	D.P. DEMAND	18
YSTAR	1	D.P. DEMAND	12
S.P. CAPACITY	66	D.P. DEMAND	44
COST	1336	D.P. DEMAND	18
YSTAR	1	D.P. DEMAND	33

SHIPPING ROUTES		DEMAND POINTS		SUPPLY POINTS	
ROUTE	FROM	D.P. DEMAND	D.P. DEMAND	S.P. CAPACITY	COST
100	1	1	1	55	1006
97	1	1	1	55	1276
94	1	1	1	55	1059
91	1	1	1	68	1246
88	1	1	1	66	1276
85	1	1	1	54	1250
82	1	1	1	64	1336
79	1	1	1	54	1276
76	1	1	1	53	1059
73	1	1	1	54	1276
70	1	1	1	66	1336
67	1	1	1	66	1336
64	1	1	1	54	1276
61	1	1	1	53	1059
58	1	1	1	54	1276
55	1	1	1	68	1246
52	1	1	1	66	1276
49	1	1	1	54	1250
46	1	1	1	64	1336
43	1	1	1	54	1276
40	1	1	1	53	1059
37	1	1	1	54	1276
34	1	1	1	66	1336
31	1	1	1	66	1336
28	1	1	1	54	1276
25	1	1	1	53	1059
22	1	1	1	54	1276
19	1	1	1	68	1246
16	1	1	1	66	1276
13	1	1	1	54	1250
10	1	1	1	64	1336
7	1	1	1	54	1276
4	1	1	1	53	1059
1	1	1	1	54	1276
27	2	2	2	55	1006
24	2	2	2	55	1276
21	2	2	2	55	1059
18	2	2	2	68	1246
15	2	2	2	66	1276
12	2	2	2	54	1250
9	2	2	2	64	1336
6	2	2	2	54	1276
3	2	2	2	53	1059
20	3	3	3	54	1276
17	3	3	3	66	1336
14	3	3	3	66	1336
11	3	3	3	54	1276
8	3	3	3	53	1059
5	3	3	3	54	1276
2	3	3	3	68	1246
98	4	4	4	66	1276
95	4	4	4	54	1250
92	4	4	4	64	1336
89	4	4	4	64	1336
86	4	4	4	54	1276
83	4	4	4	53	1059
80	4	4	4	54	1276
77	4	4	4	68	1246
74	4	4	4	66	1276
71	4	4	4	54	1250
68	4	4	4	64	1336
65	4	4	4	64	1336
62	4	4	4	54	1276
59	4	4	4	53	1059
56	4	4	4	54	1276
53	4	4	4	68	1246
50	4	4	4	66	1276
47	4	4	4	54	1250
44	4	4	4	64	1336
41	4	4	4	64	1336
38	4	4	4	54	1276
35	4	4	4	53	1059
32	4	4	4	54	1276
29	4	4	4	68	1246
26	4	4	4	66	1276
23	4	4	4	54	1250
20	4	4	4	64	1336
17	4	4	4	64	1336
14	4	4	4	54	1276
11	4	4	4	53	1059
8	4	4	4	54	1276
5	4	4	4	68	1246
2	4	4	4	66	1276
99	5	5	5	54	1276
96	5	5	5	66	1336
93	5	5	5	64	1336
90	5	5	5	54	1276
87	5	5	5	53	1059
84	5	5	5	54	1276
81	5	5	5	68	1246
78	5	5	5	66	1276
75	5	5	5	54	1250
72	5	5	5	64	1336
69	5	5	5	64	1336
66	5	5	5	54	1276
63	5	5	5	53	1059
60	5	5	5	54	1276
57	5	5	5	68	1246
54	5	5	5	66	1276
51	5	5	5	54	1250
48	5	5	5	64	1336
45	5	5	5	64	1336
42	5	5	5	54	1276
39	5	5	5	53	1059
36	5	5	5	54	1276
33	5	5	5	68	1246
30	5	5	5	66	1276
27	5	5	5	54	1250
24	5	5	5	64	1336
21	5	5	5	64	1336
18	5	5	5	54	1276
15	5	5	5	53	1059
12	5	5	5	54	1276
9	5	5	5	68	1246
6	5	5	5	66	1276
3	5	5	5	54	1250
20	6	6	6	64	1336
17	6	6	6	64	1336
14	6	6	6	54	1276
11	6	6	6	53	1059
8	6	6	6	54	1276
5	6	6	6	68	1246
2	6	6	6	66	1276
99	7	7	7	54	1276
96	7	7	7	66	1336
93	7	7	7	64	1336
90	7	7	7	54	1276
87	7	7	7	53	1059
84	7	7	7	54	1276
81	7	7	7	68	1246
78	7	7	7	66	1276
75	7	7	7	54	1250
72	7	7	7	64	1336
69	7	7	7	64	1336
66	7	7	7	54	1276
63	7	7	7	53	1059
60	7	7	7	54	1276
57	7	7	7	68	1246
54	7	7	7	66	1276
51	7	7	7	54	1250
48	7	7	7	64	1336
45	7	7	7	64	1336
42	7	7	7	54	1276
39	7	7	7	53	1059
36	7	7	7	54	1276
33	7	7	7	68	1246
30	7	7	7	66	1276
27	7	7	7	54	1250
24	7	7	7	64	1336
21	7	7	7	64	1336
18	7	7	7	54	1276
15	7	7	7	53	1059
12	7	7	7	54	1276
9	7	7	7	68	1246
6	7	7	7	66	1276
3	7	7	7	54	1250
20	8	8	8	64	1336
17	8	8	8	64	1336
14	8	8	8	54	1276
11	8	8	8	53	1059
8	8	8	8	54	1276
5	8	8	8	68	1246
2	8	8	8	66	1276
99	9	9	9	54	1276
96	9	9	9	66	1336
93	9	9	9	64	1336
90	9	9	9	54	1276
87	9	9	9	53	1059
84	9	9	9	54	1276
81	9	9	9	68	1246
78	9	9	9	66	1276
75	9	9	9	54	1250
72	9	9	9	64	1336
69	9	9	9	64	1336
66	9	9	9	54	1276
63	9	9	9	53	1059
60	9	9	9	54	1276
57	9	9	9	68	1246
54	9	9	9	66	1276
51	9	9	9	54	1250
48	9	9	9	64	1336
45	9	9	9	64	1336
42	9	9	9	54	1276
39	9	9	9	53	1059
36	9	9	9	54	1276
33	9	9	9	68	1246
30	9	9	9	66	1276
27	9	9	9	54	1250
24	9	9	9	64	1336
21	9	9	9	64	1336
18	9	9	9	54	1276
15	9	9	9	53	1059
12	9	9	9	54	1276
9	9	9	9	68	1246
6	9	9	9	66	1276
3	9	9	9	54	1250
20	10	10	10	64	1336
17	10	10	10	64	1336
14	10	10	10		

PROBLEM NO. 9 TOTAL COST = 6128 VARIABLE COST = 3986 FIXED COST = 2142

SUPPLY POINTS		DEMAND POINTS	
S.P.	CAPACITY	D.P.	DEMAND
1	32	1	4
2	32	2	7
3	36	3	4
4	32	4	10
5	62	5	13
6	272	6	16
7	111	7	31
8	222	8	63
9	149	9	74
10	275	10	149
11	275	11	225
12	145	12	272
13	99	13	74
14	202	14	225
15	61	15	149
16	266	16	272
17	0	17	63
18	0	18	28
19	0	19	25
20	0	20	28
21	0	21	13
22	0	22	19
23	0	23	22
24	0	24	15
25	0	25	7
26	0	26	42
27	0	27	62
28	0	28	15
29	0	29	12
30	0	30	17
31	0	31	20
32	0	32	22
33	0	33	18
34	0	34	20
35	0	35	10

SHIPPING ROUTES

FROM		TO		COST		XSTAR	
ROUTE	FROM	ROUTE	TO	COST	ROUTE	FROM	TO
1	1	1	1	17	1	1	1
2	2	2	2	28	2	2	2
3	3	3	3	24	3	3	3
4	4	4	4	27	4	4	4
5	5	5	5	15	5	5	5
6	6	6	6	16	6	6	6
7	7	7	7	11	7	7	7
8	8	8	8	19	8	8	8
9	9	9	9	11	9	9	9
10	10	10	10	11	10	10	10
11	11	11	11	11	11	11	11
12	12	12	12	13	12	12	12
13	13	13	13	15	13	13	13
14	14	14	14	16	14	14	14
15	15	15	15	16	15	15	15
16	16	16	16	16	16	16	16
17	17	17	17	16	17	17	17
18	18	18	18	16	18	18	18
19	19	19	19	16	19	19	19
20	20	20	20	17	20	20	20
21	21	21	21	17	21	21	21
22	22	22	22	17	22	22	22
23	23	23	23	17	23	23	23
24	24	24	24	17	24	24	24
25	25	25	25	17	25	25	25
26	26	26	26	17	26	26	26
27	27	27	27	17	27	27	27
28	28	28	28	17	28	28	28
29	29	29	29	17	29	29	29
30	30	30	30	17	30	30	30
31	31	31	31	17	31	31	31
32	32	32	32	17	32	32	32
33	33	33	33	17	33	33	33
34	34	34	34	17	34	34	34
35	35	35	35	17	35	35	35
36	36	36	36	17	36	36	36
37	37	37	37	17	37	37	37
38	38	38	38	17	38	38	38
39	39	39	39	17	39	39	39
40	40	40	40	17	40	40	40
41	41	41	41	17	41	41	41
42	42	42	42	17	42	42	42
43	43	43	43	17	43	43	43
44	44	44	44	17	44	44	44
45	45	45	45	17	45	45	45
46	46	46	46	17	46	46	46
47	47	47	47	17	47	47	47
48	48	48	48	17	48	48	48
49	49	49	49	17	49	49	49
50	50	50	50	17	50	50	50
51	51	51	51	17	51	51	51
52	52	52	52	17	52	52	52
53	53	53	53	17	53	53	53
54	54	54	54	17	54	54	54
55	55	55	55	17	55	55	55
56	56	56	56	17	56	56	56
57	57	57	57	17	57	57	57
58	58	58	58	17	58	58	58
59	59	59	59	17	59	59	59
60	60	60	60	17	60	60	60
61	61	61	61	17	61	61	61
62	62	62	62	17	62	62	62
63	63	63	63	17	63	63	63
64	64	64	64	17	64	64	64
65	65	65	65	17	65	65	65
66	66	66	66	17	66	66	66
67	67	67	67	17	67	67	67
68	68	68	68	17	68	68	68
69	69	69	69	17	69	69	69
70	70	70	70	17	70	70	70
71	71	71	71	17	71	71	71
72	72	72	72	17	72	72	72
73	73	73	73	17	73	73	73
74	74	74	74	17	74	74	74
75	75	75	75	17	75	75	75
76	76	76	76	17	76	76	76
77	77	77	77	17	77	77	77
78	78	78	78	17	78	78	78
79	79	79	79	17	79	79	79
80	80	80	80	17	80	80	80
81	81	81	81	17	81	81	81
82	82	82	82	17	82	82	82
83	83	83	83	17	83	83	83
84	84	84	84	17	84	84	84
85	85	85	85	17	85	85	85
86	86	86	86	17	86	86	86
87	87	87	87	17	87	87	87
88	88	88	88	17	88	88	88
89	89	89	89	17	89	89	89
90	90	90	90	17	90	90	90
91	91	91	91	17	91	91	91
92	92	92	92	17	92	92	92
93	93	93	93	17	93	93	93
94	94	94	94	17	94	94	94
95	95	95	95	17	95	95	95
96	96	96	96	17	96	96	96
97	97	97	97	17	97	97	97
98	98	98	98	17	98	98	98
99	99	99	99	17	99	99	99
100	100	100	100	17	100	100	100

References

1. L. B. Ellwein, Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints, Technical Report 70-2, Department of Industrial Engineering, Stanford University, 1970.
2. J. M. Fleisher and R. R. Meyer, A New Class of Sufficient Optimality Conditions for Integer Programming, Computer Sciences Technical Report No. 248, University of Wisconsin, Madison, April, 1975.
3. R. Garfinkel and G. Nemhauser, Integer Programming, John Wiley & Sons, N.Y., 1972.
4. O. L. Mangasarian, Nonlinear Programming, McGraw-Hill Book Co., N.Y., 1969.
5. R. Shreshain, Branch and Bound Mixed Integer Programming - BBMP, IBM, New York Scientific Center, Share Program Library, 360D-15.2.005, 1967.
6. H. M. Stark, An Introduction to Number Theory, Markham Publishing Co., Chicago, 1970.
7. L. Trotter, Jr., User's Instructions for the Integer Programming Code ENUMER8, MRC Technical Summary Report No. 1347, Mathematics Research Center, University of Wisconsin, Madison, December, 1973.
8. L. Trotter, Jr. and C. Shetty, An Algorithm for the Bounded Variable Integer Programming Problem, MRC Technical Summary Report No. 1355, Mathematics Research Center, University of Wisconsin, Madison, December, 1973.
9. H. Wagner, Principles of Operations Research, Prentice Hall, Inc., N.J., 1969.