Equivalent Instances of the Simple Plant Location Problem

Bader F. AlMohammad
Boris Goldengorin*
Diptesh Ghosh
Gerard Sierksma
Faculty of Economic Sciences, University of Groningen, The Netherlands

SOM-theme A Primary Processes within Firms

Abstract

In this paper we deal with a pseudo-Boolean representation of the simple plant location problem. We define instances of this problem that are equivalent, in the sense that each feasible solution has the same goal function value in all such instances. We further define a collection of polytopes whose union describes the set of instances equivalent to a given instance. We use the concept of equivalence to develop a method by which we can extend the set of instances that we can solve using our knowledge of polynomially solvable special cases. We also present a new preprocessing rule that allows us to determine sites in which facilities will not be located in an optimal solution and thereby reduce the size of a problem instance.

Keywords: Simple Plant Location Problem, Pseudo-Boolean function, Polytopes, Equivalence, Preprocessing, Polynomially Solvable Special Cases.

* Corresponding author. Email: b.goldengorin@eco.rug.nl

1. Introduction

In this paper, we study the Simple Plant Location Problem (SPLP). A detailed introduction to this problem appears in Cornuejols *et al.* [3]. The goal of the problem is one of determining the cheapest method of meeting the demands of a set of clients from plants that can be located at some candidate sites. The costs involved in meeting the client demands include the fixed cost of setting up a plant at a given site, and the per unit transportation cost of supplying a given client from a plant located at a given site. This problem forms the underlying model in several combinatorial problems, like set covering, set partitioning, information retrieval, simplification of logical Boolean expressions, airline crew scheduling, vehicle despatching (see Christofides [4]), assortment (see Beresnev *et al.* [2], Goldengorin [9], Jones *et al.* [12], Pentico [14, 15], Tripathy *et al.* [17]) and is a subproblem for various location analysis problems (see Revelle and Laporte [16]). We will assume that the capacity at each plant is sufficient to meet the demand of all clients. We will further assume that each client has a demand of one unit, which must be met by one of the opened plants. If a client's demand is different from one unit, we can scale the demand to unity by scaling the transportation costs accordingly. Conventional solution methods for the problem are based on branch and bound techniques (see Cornuejols [3] for a detailed treatment). However there is another approach that uses pseudo-Boolean functions.

It is easy to see that any instance of the SPLP has an optimal solution in which each customer is satisfied by exactly one plant. In Hammer [11] this fact is used to derive a pseudo-Boolean representation of this problem. The pseudo-Boolean function developed in that work has terms that contain both a literal and its complement. Subsequently, in Beresnev [1] a different pseudo-Boolean form has been developed in which each term contains only literals or only their complements. We find this form easier to manipulate, and hence use Beresnev's formulation in this paper.

In Section 2 of this paper we use Beresnev's pseudo-Boolean formulation of the SPLP, and develop the concept of equivalent instances, i.e. instances that have the same goal function values for the same solution. We then illustrate the use of the concept of equivalence in Section 3 to develop heuristics that recognize whether a given instance is solvable using our knowledge of polynomially solvable cases. We also demonstrate the use of equivalence to develop preprocessing rules that are stronger than the existing rules. We conclude the paper in Section 4 with a summary of the contributions of this paper, and brief remarks on possible directions for future research.

2. A Pseudo-Boolean Formulation and Equivalent Instances

Given sets $I = \{1, 2, \ldots, m\}$ of sites in which plants can be located, and $J = \{1, 2, \ldots, n\}$ of clients, a vector $F = (f_i)$ of fixed costs for setting up plants at sites $i \in I$, a matrix $C = [c_{ij}]$ of transportation costs from $i \in I$ to $j \in J$, and an unit demand at each client site, the Simple

Plant Location Problem (SPLP) is the problem of finding a set S, $\emptyset \subset S \subseteq I$, at which plants can be located so that the total cost of satisfying all client demands is minimal. An instance of the problem is described by a m-vector $F = (f_i)$, and a $m \times n$ matrix $C = [c_{ij}]$. We assume that F and C are nonnegative and finite, i.e. $F \in \Re^m_+$, and $C \in \Re^{mn}_+$. We will use the $m \times (n+1)$ *augmented matrix* $[F|C]$ as a shorthand for describing an instance of the SPLP. The total cost $f_{[F|C]}(S)$ associated with a solution S consists of two components, the fixed costs $\sum_{i \in S} f_i$, and the transportation costs $\sum_{j \in J} \min\{c_{i,j} | i \in S\}$, i.e.

$$f_{[F|C]}(S) = \sum_{i \in S} f_i + \sum_{j \in J} \min\{c_{i,j} | i \in S\},$$

and the SPLP is problem of finding

$$S^\star \in \arg\min\{f_{[F|C]}(S) : \emptyset \subset S \subseteq I\}. \tag{1}$$

A $m \times n$ *ordering matrix* $\Pi = [\pi_{ij}]$ is a matrix each of whose columns $\Pi_j = (\pi_{1j}, \dots, \pi_{mj})^T$ define a permutation of $1, \dots, m$. Given a transportation matrix C, the set of all ordering matrices Π such that $c_{\pi_{1j}j} \leq c_{\pi_{2j}j} \leq \cdots \leq c_{\pi_{mj}j}$, for $j = 1, \dots, n$, is denoted by $perm(C)$.

Defining

$$y_i = \begin{cases} 0 & \text{if } i \in S \\ 1 & \text{otherwise,} \end{cases} \quad \text{for each } i = 1, \dots, m \tag{2}$$

we can indicate any solution S by a vector $\mathbf{y} = (y_1, y_2, \dots, y_m)$. The fixed cost component of the total cost can be written as

$$\mathcal{F}_F(\mathbf{y}) = \sum_{i=1}^{m} f_i (1 - y_i). \tag{3}$$

Given a transportation cost matrix C, and an ordering matrix $\Pi \in perm(C)$, we can denote differences between the transportation costs for each $j \in J$ as

$$\begin{aligned} \Delta c[0, j] &= c_{\pi_{1j}j}, \quad \text{and} \\ \Delta c[l, j] &= c_{\pi_{(l+1)j}j} - c_{\pi_{lj}j}, \quad l = 1, \dots, m-1. \end{aligned}$$

Then, for each $j \in J$,

$$\begin{aligned} \min\{c_{i,j} | i \in S\} &= \Delta c[0, j] + \Delta c[1, j] \cdot y_{\pi_{1j}} + \Delta c[2, j] \cdot y_{\pi_{1j}} \cdot y_{\pi_{2j}} \\ &\quad + \cdots + \Delta c[m-1, j] \cdot y_{\pi_{1j}} \cdots y_{\pi_{(m-1)j}} \\ &= \Delta c[0, j] + \sum_{k=1}^{m-1} \Delta c[k, j] \cdot \prod_{r=1}^{k} y_{\pi_{rj}}, \end{aligned}$$

so that the transportation cost component of the cost of a solution $\mathbf{y}$ corresponding to an ordering matrix $\Pi \in perm(C)$ is

$$\mathcal{T}_{C,\Pi}(\mathbf{y}) = \sum_{j=1}^{n} \left\{ \Delta c[0, j] + \sum_{k=1}^{m-1} \Delta c[k, j] \cdot \prod_{r=1}^{k} y_{\pi_{rj}} \right\}. \tag{4}$$

Lemma 2.1 *$\mathcal{T}_{C,\Pi}(\cdot)$ is identical for all $\Pi \in perm(C)$.*

PROOF. Let $\Pi = [\pi_{ij}]$, $\Psi = [\psi_{ij}] \in perm(C)$, and any $\mathbf{y} \in \{0, 1\}^m$. It is sufficient to prove that $\mathcal{T}_{C,\Pi}(\mathbf{y}) = \mathcal{T}_{C,\Psi}(\mathbf{y})$ when

$$\pi_{kl} = \psi_{(k+1)l}, \tag{5}$$
$$\pi_{(k+1)l} = \psi_{kl}, \tag{6}$$
$$\pi_{ij} = \psi_{i,j} \quad \text{if } (i, j) \neq (k, l). \tag{7}$$

Then

$$\mathcal{T}_{C,\Pi}(\mathbf{y}) - \mathcal{T}_{C,\Psi}(\mathbf{y}) = (c_{\pi_{(k+1)l}l} - c_{\pi_{kl}l}) \cdot \prod_{i=1}^{k} y_{\pi_{il}} - (c_{\psi_{(k+1)l}l} - c_{\psi_{kl}l}) \cdot \prod_{i=1}^{k} y_{\psi_{il}}.$$

But (5) and (6) imply that $c_{\pi_{(k+1)l}l} = c_{\pi_{kl}l}$ and $c_{\psi_{(k+1)l}l} = c_{\psi_{kl}l}$ which in turn imply that $\mathcal{T}_{C,\Pi}(\mathbf{y}) = \mathcal{T}_{C,\Psi}(\mathbf{y})$. ■

Combining (3) and (4), the total cost of a solution $\mathbf{y}$ to the instance $[F|C]$ corresponding to an ordering matrix $\Pi \in perm(C)$ is

$$\begin{aligned} f_{[F|C],\Pi}(\mathbf{y}) &= \mathcal{F}_F(\mathbf{y}) + \mathcal{T}_{C,\Pi}(\mathbf{y}) \\ &= \sum_{i=1}^{m} f_i(1 - y_i) + \sum_{j=1}^{n} \left\{ \Delta c[0, j] + \sum_{k=1}^{m-1} \Delta c[k, j] \cdot \prod_{r=1}^{k} y_{\pi_{rj}} \right\}. \end{aligned} \tag{8}$$

Lemma 2.2 *The total cost function $f_{[F|C],\Pi}(\cdot)$ is identical for all $\Pi \in perm(C)$.*

PROOF. This is a direct consequence of Lemma 2.1. ■

A *pseudo-Boolean polynomial* of degree n is a polynomial of the form

$$P(\mathbf{y}) = \sum_{T \in 2^n} \alpha_T \cdot \prod_{i \in T} y_i,$$

where 2^n is the power set of $\{1, 2, \ldots, n\}$ and α_T can assume arbitrary values. We call a pseudo-Boolean polynomial $P(\mathbf{y})$ a *Beresnev function* if there exists a SPLP instance $[F|C]$ and $\Pi \in perm(C)$ such that $P(\mathbf{y}) = f_{[F|C],\Pi}(\mathbf{y})$ for $\mathbf{y} \in \{0, 1\}^m$. We denote a Beresnev function corresponding to a given SPLP instance $[F|C]$ by $\mathcal{B}_{[F|C]}(\mathbf{y})$ and define it as

$$\mathcal{B}_{[F|C]}(\mathbf{y}) = f_{[F|C],\Pi}(\mathbf{y}) \text{ where } \Pi \in perm(C). \tag{9}$$

Theorem 2.1 *A general pseudo-Boolean function is a Beresnev function if and only if*

(a) All coefficients of the pseudo-Boolean function except those of the linear terms are non-negative.
(b) The sum of the constant term and the coefficients of all the negative linear terms in the pseudo-Boolean function is non-negative.

PROOF. The "if" statement is trivial. In order to prove the "only if" statement, consider a SPLP instance $[F|C]$, an ordering matrix $\Pi \in perm(C)$, and a Beresnev function $\mathcal{B}_{[F|C]}(\mathbf{y})$ in which there is a non-linear term of degree k with a negative coefficient. Since non-linear terms are contributed by the transportation costs only, a non-linear term with a negative coefficient implies that $\Delta[k, j]$ for some $j \in \{1, \ldots, n\}$ is negative. But this contradicts the fact that $\Pi \in perm(C)$. Next suppose that in $\mathcal{B}_{[F|C]}(\mathbf{y})$, the sum of the constant term and the coefficients of the negative linear terms is negative. This implies that the coefficient of some linear term in the transportation cost function is negative. But this also contradicts the fact that $\Pi \in perm(C)$. The logic above holds true for all members of $perm(C)$ as a consequence of Lemma 2.1. ■

We can formulate (1) in terms of Beresnev functions as

$$\mathbf{y}^\star \in \arg\min\{\mathcal{B}_{[F|C]}(\mathbf{y}) : \mathbf{y} \in \{0, 1\}^m, \mathbf{y} \neq \mathbf{1}\}. \tag{10}$$

As an example, consider the SPLP instance:

$$[F|C] = \left[\begin{array}{c|ccccc} 7 & 7 & 15 & 10 & 7 & 10 \\ 3 & 10 & 17 & 4 & 11 & 22 \\ 3 & 16 & 7 & 6 & 18 & 14 \\ 6 & 11 & 7 & 6 & 12 & 8 \end{array}\right]. \tag{11}$$

Two of the four possible ordering matrices corresponding to C are

$$\Pi_1 = \begin{bmatrix} 1 & 3 & 2 & 1 & 4 \\ 2 & 4 & 3 & 2 & 1 \\ 4 & 1 & 4 & 4 & 3 \\ 3 & 2 & 1 & 3 & 2 \end{bmatrix} \text{ and } \Pi_2 = \begin{bmatrix} 1 & 4 & 2 & 1 & 4 \\ 2 & 3 & 4 & 2 & 1 \\ 4 & 1 & 3 & 4 & 3 \\ 3 & 2 & 1 & 3 & 2 \end{bmatrix}. \tag{12}$$

The Beresnev function is $\mathcal{B}_{[F|C]}(\mathbf{y}) = [7(1-y_1)+3(1-y_2)+3(1-y_3)+6(1-y_4)]+[7+3y_1+1y_1y_2+5y_1y_2y_4]+[7+0y_3+8y_3y_4+2y_1y_3y_4]+[4+2y_2+0y_2y_3+4y_2y_3y_4]+[7+4y_1+1y_1y_2+6y_1y_2y_4]+[8+2y_4+4y_1y_4+8y_1y_3y_4]$
$= 52 - y_2 - 3y_3 - 4y_4 + 2y_1y_2 + 8y_3y_4 + 4y_1y_4 + 11y_1y_2y_4 + 10y_1y_3y_4 + 4y_2y_3y_4$.

In general, there are many different SPLP instances that yield the same Beresnev function. This is due to the fact that we can aggregate terms in the Beresnev function. If two SPLP instances of the same size have the same Beresnev function, then any solution $\mathbf{y}$ has the same objective function value in both instances. Therefore, a solution that is optimal to one of the instances is optimal to the other as well. We call such instances equivalent. Formally defined, two SPLP instances $[F|C]$ and $[S|D]$ are called *equivalent* if they are of the same size and if $\mathcal{B}_{[F|C]} = \mathcal{B}_{[S|D]}$. Beresnev functions of SPLP instances can be generated in polynomial time, and have a number of terms that is polynomial in the size of the instance. Therefore it is possible to check the equivalence of two instances in polynomial time, even though the SPLP is a $\mathcal{NP}$-hard problem.

Note however that the condition of equivalence is only a sufficient condition for two SPLP instances to have the same optimal solution. For instance the two instances

$$[F|C] = \left[\begin{array}{c|cc} 1 & 3 & 3 \\ 2 & 5 & 5 \end{array}\right] \text{ and } [S|D] = \left[\begin{array}{c|cc} 1 & 1 & 1 \\ 3 & 2 & 2 \end{array}\right]$$

have different Beresnev functions ($\mathcal{B}_{[F|C]}(\mathbf{y}) = 9 + 3y_1 - 2y_2$ and $\mathcal{B}_{[S|D]}(\mathbf{y}) = 6 + y_1 - 3y_2$) but the same (and unique) optimal solution, (0, 1).

Let us now consider the set of all SPLP instances $[S|D]$ that are equivalent to a given SPLP instance $[F|C]$. This set can be defined as

$$\mathcal{P}_{[F|C]} = \left\{[S|D] \in \Re_+^{m\times(n+1)} : \mathcal{B}_{[F|C]} = \mathcal{B}_{[S|D]}\right\}. \tag{13}$$

$\mathcal{P}_{[F|C]}$ can be rewritten as

$$\mathcal{P}_{[F|C]} = \bigcup_{\Pi\in perm(E)} P_{[F|C],\Pi},$$

where E is the $m \times (n+1)$ all-unit matrix and

$$P_{[F|C],\Pi} = \left\{[S|D] \in \Re_+^{m\times(n+1)} : \mathcal{B}_{[F|C]} = \mathcal{B}_{[S|D]},\, \Pi \in perm(D)\right\}. \tag{14}$$

We show below that each of the sets $P_{[F|C],\Pi}$ can be described by a system of linear ineqalities.

Let us assume that $\Psi \in perm(C)$ and $\Pi \in perm(D)$. The Beresnev function for $[F|C]$ is

$$\begin{aligned}\mathcal{B}_{[F|C]}(\mathbf{y}) &= \sum_{i=1}^{m} f_i(1-y_i) + \sum_{j=1}^{n} \Delta c[0,j] + \sum_{j=1}^{n} \Delta c[1,j] y_{\psi_{1j}} + \\ &\quad \sum_{k=2}^{m-1}\sum_{j=1}^{n} \Delta c[k,j] \prod_{r=1}^{k} y_{\psi_{rj}}. \end{aligned} \tag{15}$$

The Beresnev function for $[S|D]$ is

$$\begin{aligned}\mathcal{B}_{[S|D]}(\mathbf{y}) &= \sum_{i=1}^{m} s_i(1-y_i) + \sum_{j=1}^{n} \Delta d[0,j] + \sum_{j=1}^{n} \Delta d[1,j] y_{\pi_{1j}} + \\ &\quad \sum_{k=2}^{m-1}\sum_{j=1}^{n} \Delta d[k,j] \prod_{r=1}^{k} y_{\psi_{rj}}. \end{aligned} \tag{16}$$

Since $[F|C]$ and $[S|D]$ are identical, we can equate like terms.

Equating the coefficients of the constant and linear terms in (15) and (16) yields

$$\sum_{i=1}^{m} s_i + \sum_{j=1}^{n} \Delta d[0,j] = \sum_{i=1}^{m} f_i + \sum_{j=1}^{n} \Delta c[0,j] \tag{17}$$

$$\sum_{j:\pi_{1j}=k} \Delta d[1,j] - s_k = \sum_{j:\psi_{1j}=k} \Delta c[1,j] - f_k \quad k = 1,\ldots,m. \tag{18}$$

Equating the non-linear terms we get the equations

$$\sum_{\{\psi_{1j},\ldots,\psi_{kj}\}=\{\pi_{1j},\ldots,\pi_{kj}\}} \Delta d[k,j] - \Delta c[k,j] = 0 \qquad k = 2,\ldots m-1, \quad j = 1,\ldots n. \tag{19}$$

Finally, since $\psi \in perm(C)$ and $\Pi \in perm(D)$, and since all entries in the instances are assumed to be non-negative, we have that

$$\Delta d[k,j] \geq 0 \qquad k = 0,\ldots,m-1; \quad j = 1,\ldots,n. \tag{20}$$

$$s_i, d_{ij} \geq 0 \qquad i = 1,\ldots,m; \quad j = 1,\ldots,n; \tag{21}$$

Consider the instance in (11). Then $P_{[F|C],\Pi_1}$ (where Π_1 is defined in (12)) is defined by the following system.

Equations corresponding to (17):

$$s_1 + s_2 + s_3 + s_4 + d_{11} + d_{32} + d_{23} + d_{14} + d_{45} = 52.$$

Equations corresponding to (18):

$$
\begin{aligned}
&s_1 - (d_{21} - d_{11}) - (d_{24} - d_{14}) = 0,\\
&s_2 - (d_{33} - d_{23}) = 1,\\
&s_3 - (d_{42} - d_{32}) = 3,\\
&s_4 - (d_{15} - d_{45}) = 4.
\end{aligned}
$$

Equations corresponding to (19):

$$
\begin{aligned}
&(d_{41} - d_{21}) + (d_{44} - d_{24}) = 2,\\
&(d_{12} - d_{42}) = 8,\\
&(d_{43} - d_{33}) = 0,\\
&(d_{35} - d_{15}) = 4,\\
&(d_{31} - d_{41}) + (d_{34} - d_{44}) = 11,\\
&(d_{22} - d_{12}) + (d_{25} - d_{35}) = 10,\\
&(d_{13} - d_{43}) = 4.
\end{aligned}
$$

Inequalities corresponding to (20):

$$
\begin{aligned}
&d_{21} - d_{11}, d_{42} - d_{32}, d_{33} - d_{23}, d_{24} - d_{14}, d_{15} - d_{45} \geq 0\\
&d_{41} - d_{21}, d_{12} - d_{42}, d_{43} - d_{33}, d_{44} - d_{24}, d_{35} - d_{15} \geq 0\\
&d_{31} - d_{41}, d_{22} - d_{12}, d_{13} - d_{43}, d_{34} - d_{44}, d_{25} - d_{35} \geq 0
\end{aligned}
$$

Inequalities corresponding to (21):

$$
s_1, s_2, s_3, s_4, d_{11}, d_{12}, \ldots , d_{44}, d_{45} \geq 0.
$$

Note that there may exist $\Pi \in perm(E)$ for which $P_{[F|C],\Pi}$ is empty. For example, for the instance in (11), $P_{[F|C],\Pi}$ corresponding to

$$
\Pi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}
$$

is empty.

Lemma 2.3 *Given a SPLP instance $[F|C]$ of size $m \times (n+1)$, a lower bound to the number of non-empty polytopes $P_{[F|C],\Pi}$, $\Pi \in perm(E)$ is a Stirling number of the second kind,*

$$
S(n,k) = \frac{n^k}{\prod_{j=1}^{k}(1 - j \cdot n)}
$$

where k is the maximum number of distinct terms of the same degree in $\mathcal{B}_{[F|C]}(\mathbf{y})$.

PROOF. Consider any $m \times n$ ordering matrix Π. It is clear that any two columns of Π will give rise to distinct terms in a Beresnev function only if the two columns are distinct. So in order for an instance $[S|D]$ to have a Beresnev function identical to that of $[F|C]$, the number of distinct columns in an ordering matrix $\Pi_D \in perm(D)$ must be at least as large as the maximum number of distinct terms of any degree in $\mathcal{B}_{[F|C]}(\mathbf{y})$. Let there be k distinct columns in $\Pi_D \in perm(D)$. If $F(n, k)$ denotes the number of ordering matrices that can be formed with this stipulation, then

$$F(n, k) = k \cdot (F(n-1, k-1) + F(n-1), k),$$

with boundary conditions

$$\begin{aligned} F(k, k) &= k!, \\ F(k-1, k) &= 0. \end{aligned}$$

The solution to this set of recurrence equations is

$$F(n, k) = S(n, k),$$

(refer David *et al.* [5], Lindquist and Sierksma [13]), which proves the desired result. ■

3. Solving the SPLP

In this section we address the problem of solving a given instance of SPLP. Solution procedures to NP-hard optimization problems generally first try to see if the problem is of a form known to be polynomially solvable. If it is, then the problem is solved using a polynomial time algorithm. Otherwise, pre-processing operations are carried out to reduce the size of the instance. If the reduced instance is also not polynomilally solvable, then a general (exponential time) algorithm, or a heuristic is employed to solve the reduced instance. In this section we will discuss the use of Beresnev functions for recognizing whether a problem is polynomially solvable and for performing pre-processing operations.

3.1 Solving SPLP instances using Polynomially Solvable Cases

The conventional method of using our knowledge of polynomially solvable cases of $\mathcal{NP}$-hard optimization is the following. Given an instance of the problem, we check, using a polynomial recognition algorithm, whether the problem data corresponds to that of a pre-defined polynomially solvable case. If it does, (for example, for instance I_1 in Figure 3.1(a)) then an optimal solution to the instance is obtained using a polynomial time algorithm. If it does not, (for example, for instance I_2 in Figure 3.1(a)) then conventional approaches terminate reporting that the instance is not polynomially solvable.

Our approach to the problem of solving simple plant location problems through knowledge about polynomially solvable special cases is different. If we recognize, using a polynomial recognition algorithm, that the data in a given instance matches that of a pre-specified polynomially solvable case (for example, I_1 in Figure 3.1(b)), then we obtain an optimal solution to the instance using a polynomial time algorithm. In case the instance data does not correspond to that of a polynomially solvable case, (for example, I_2 in Figure 3.1(b)), we use the concept of equivalence in order to attempt to solve the instance polynomially. In case the set of instances equivalent to the given instance has a non-empty intersection with the pre-defined set of polynomially solvable instances, then we could solve an instance in the intersection of the two sets (for example, instance I_3 in Figure 3.1(b)), to obtain a solution to the given instance. However, finding an instance in the intersection of the set of equivalent instances and the set of polynomially solvable instances usually takes exponential time, so in our approach, the checking is carried out in an inexact but polynomial time manner. The set of instances that are solved polynomially using our approach is therefore a superset of the set of instances solved polynomially using the conventional approach.

Many polynomially solvable special cases of the SPLP have been reported in the literature (see Beresnev *et al.* [2], Goldengorin [9]) and references within, Jones *et al.* [12]). Most of these are obtained by imposing certain conditions on the transportation cost matrix. In this subsection we show how we can use the concept of equivalence to solve the recognition problem mentioned above for the special case of *quasiconcave* matrices.

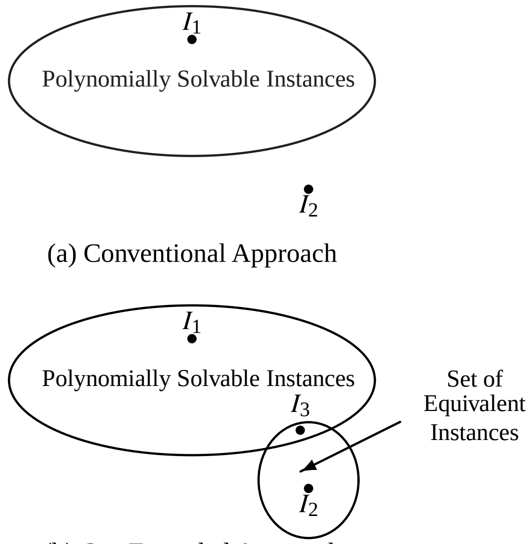
A $m \times n$ matrix $A = [a_{ij}]$ is called *quasiconcave* if there exists a permutation of rows $\langle r[1], \ldots, r[m] \rangle$, and an index k_j, $1 \leq k_j \leq m$, for each column $j \in J$, such that

$$a_{r[1]j} \leq \cdots \leq a_{r[k_j]j} \geq \cdots \geq a_{r[m]j}.$$

Let $A^{(k)}$ be the set of all elements in the first k rows of a $m \times n$ matrix A. A is said to be *2-compact* iff

$$|A^{(k+1)} - A^{(k)}| \leq 2 \quad \text{for } k = 1, \ldots, m-1.$$

In Goldengorin [9] it is shown that if the transportation cost matrix of a given SPLP instance is quasiconcave then there exists an optimal solution in which there are at most two opened plants. It is also shown that a transportation cost matrix C is quasiconcave if there is a $\Pi \in perm(C)$ such that Π is 2-compact.



(a) Conventional Approach

(b) Our Extended Approach

Figure 3.1: Handling polynomially solvable special cases

The RECOGNIZE heuristic (see Figure 3.2) uses the concept of equivalence and the observation in Goldengorin [9] to recognize whether a given transportation matrix C is quasiconcave.

RECOGNIZE accepts a SPLP instance $[F|C]$, and creates an ordering matrix $\Pi \in perm(C)$. It then attempts to transform this ordering matrix to a 2-compact matrix so that in all intermediate steps it is certain that there exists a SPLP instance equivalent to $[F|C]$ for which the intermediate ordering matrix corresponds to the transportation cost matrix. For any given row r, the heuristic tries to generate an ordering matrix $\Pi_{,}$ for which $|\Pi_{,}^{(r)} - \Pi_{,}^{(r-1)}| \leq 2$. In order to do this, it tries to create permutations (Statements 8 through 22) containing at most two indices not present in rows 1 through $(r-1)$. If it fails to achieve this, then it returns "NO", meaning that it could not recognize the given instance as equivalent to a polynomially solvable special case. If the heuristic can create a 2-compact ordering matrix, then it returns a "YES" signifying that there indeed exists a polynomially solvable instance equivalent to $[F|C]$.

Note that RECOGNIZE is a heuristic. If it returns "YES" then there surely exists a polynomially solvable instance equivalent to the instance input. However if it returns "NO", then there is no guarantee that a polynomially solvable instance equivalent to the instance input does not exist.

Heuristic RECOGNIZE.

Input:

SPLP instance $[F|C]$.

Output:

“YES” if RECOGNIZE recognizes an polynomially solvable instance equivalent to $[F|C]$;
“NO” otherwise.

Parameters:

C	: Transportation matrix ($m \times n$)
Π	: Ordering matrix ($m \times n$)
$unused$	: Set of indices (m)
r	: Counter for rows
k	: Counter for columns
i, j, t	: Temporary indices

```
begin
1         create an ordering matrix Π ∈ perm(C);
2         unused := {1, 2, ... , m};
3         for (r := 1 to m) do
4         begin /* iteration */
5                 if (row r in Π has not more than two indices from unused) then
6                         remove these two indices from unused;
7                 else
8                 begin
9                         choose a pair of indices i, j from unused that have
10                        not been chosen in this iteration;
11                        for (k := 1 to n) do
12                        begin
13                                if (π_rk ∉ unused \ {i, j}) then
14                                begin
15                                        t := π_rk;
16                                        π_rk := i;
17                                        if (P_[F|C],Π = ∅) then
18                                                π_rk := j;
19                                        if (P_[F|C],Π = ∅) then
20                                                go to Statement 9;
21                                end
22                        end
23                end
24                if (|Π^(r) − Π^(r−1)| > 2) then /* Assume that Π^(0) = ∅ */
25                        return “NO”;
26        end /* Iteration */
27        return “YES”;
end
```

Figure 3.2: Pseudocode for RECOGNIZE

Consider the ordering matrix Π_1 (defined in (12)) corresponding to the SPLP instance of (11). Since the first row of Π_1 contains three different indices, it is not 2-compact. However, RECOGNIZE can transform Π_1 to

$$\Pi_{'} = \begin{bmatrix} 1 & 4 & 4 & 1 & 4 \\ 2 & 3 & 2 & 2 & 1 \\ 4 & 1 & 3 & 4 & 3 \\ 3 & 2 & 1 & 3 & 2 \end{bmatrix}$$

which is 2-compact. We can therefore construct $P_{[F|C],\Pi_{'}}$ and obtain the following equivalent instance by transferring 2 units from the fixed cost of the second site to the cost of transporting a unit from site 2 to client 3.

$$[S|D] = \left[\begin{array}{c|ccccc} 7 & 7 & 15 & 10 & 7 & 10 \\ 1 & 10 & 17 & 6 & 11 & 22 \\ 3 & 16 & 7 & 6 & 18 & 14 \\ 6 & 11 & 7 & 6 & 12 & 8 \end{array}\right].$$

This instance is polynomially solvable using the observation in Goldengorin [9], and the optimal solution is to set up plants at 1 and 3 with a total cost of 47 units. Again since $[S|D]$ is equivalent to $[F|C]$, we can conclude that an optimal solution to $[F|C]$ would be to set up plants at 1 and 3, and the total cost for the solution would be 47 units.

3.2 A New Pre-processing Rule

Suppose that the given instance is not recognized to correspond to a known polynomially solvable special case. Then we have to use an exact algorithm for solving this instance. The execution times of exact algorithms for the SPLP are exponential in the parameters m and n. So any preprocessing rules, i.e. quick methods of reducing the size of the given instance, are of much practical importance. There are two rules available in the literature. These are the following.

The first one, due to Beresnev [1], states that if there are two clients that have the same sub-permutations of the transportation costs in any ordering matrix, then they can be aggregated into a single virtual client.

The second rule is due to Cornuejols *et al.* [3], and Dearing *et al.* [6]. It states that if there is a site where a plant can be opened with zero fixed cost, and if there is any client that can be served from that site at zero cost, then there exists an optimal solution in which a plant will be opened in that site.

In the remainder of this section we will show that these two rules are special cases of a more general rule, formed using Beresnev functions. In Beresnev [1], Cornuejols *et al.* [3], Dearing *et al.* [6], and Veselovsky [18] it is observed that terms can be aggregated in the Beresnev function corresponding to a SPLP instance. If two clients have the same sub-permutation, then each of the terms in $\mathcal{T}_{C,\Pi}(\mathbf{y})$ for any $\Pi \in perm(C)$ corresponding to these clients can be aggregated. This shows that the existing first rule is automatically applied when we construct a Beresnev function.

We will now state a theorem that forms the basis of our strong pre-processing rule for SPLP instances.

Theorem 3.1 *Let $\mathcal{B}_{[F|C]}(\mathbf{y})$ be the Beresnev function corresponding to the SPLP instance $[F|C]$ and $\mathbf{y} \in \{0, 1\}^m$. Then the following statements are true.*

(a) If the coefficient of the linear term involving y_k is zero, then for any optimal solution with $y_k = 1$, there exists an optimal solution with $y_k = 0$.
(b) Let the coefficient of the linear term in y_k be negative and its magnitude be f_k. Let the sum of the coefficients of all non-linear terms involving y_k be t_k. Then $y_k = 1$ in an optimal solution if $f_k \geq t_k$.

PROOF. (a) Let us consider a vector $\mathbf{y}$ corresponding to an optimal solution in which $y_k = 1$. Let us construct a vector $\mathbf{y}'$ in which $y_i = y'_i\ \forall i \neq k$, $y'_k = 0$. Now $\mathcal{B}_{[F|C]}(\mathbf{y}') - \mathcal{B}_{[F|C]}(\mathbf{y}) \leq f_k = 0$. Hence if $\mathbf{y}$ is optimal, so is $\mathbf{y}'$. The proof of (b) is similar to that of (a). ∎

This theorem leads us to the following new rule.

Let $\mathcal{B}_{[F|C]}(\mathbf{y})$ be the Beresnev function corresponding to a SPLP instance $[F|C]$ in which like terms have been aggregated. Let f_k be the coefficient of the linear term corresponding to y_k and let t_k correspond to the sum of all non-linear terms containing y_k. Then

(a) If $f_k = 0$ then there is an optimal solution in which $y_k = 0$.
(b) If $f_k < 0$, and $|f_k| \geq t_k$ then there is an optimal solution with $y_k = 1$.

Notice that the first part of this rule covers the existing second rule and extends it using the concept of equivalence. The second part of this rule is absolutely new. The two parts of the new rule can be used to pre-process a given instance.

Consider the SPLP instance

$$[F|C] = \left[\begin{array}{c|cccccc} 7 & 7 & 15 & 10 & 7 & 10 \\ 3 & 10 & 17 & 8 & 11 & 22 \\ 3 & 16 & 7 & 6 & 18 & 14 \\ 6 & 11 & 7 & 6 & 12 & 8 \end{array}\right].$$

The Beresnev function for this instance is $\mathcal{B}_{[F|C]}((y_1, y_2, y_3, y_4)) = 54 + 0y_1 - 3y_2 - 3y_3 - 4y_4 + 2y_1y_2 + 4y_1y_4 + 10y_3y_4 + 11y_1y_2y_4 + 10y_1y_3y_4 + 2y_2y_3y_4$. Since the coefficient of y_1 is zero we can set $y_1 = 0$. The Beresnev function then becomes $\mathcal{B}_{[F|C]}((0, y_2, y_3, y_4)) =$

$54 - 3y_2 - 3y_3 - 4y_4 + 10y_3y_4 + 2y_2y_3y_4$. The coefficient of the linear term involving y_2 is negative and its magnitude in the revised Beresnev polynomial is 3 while the sum of all terms containing y_2 in the transportation cost component is 2. So we can set $y_2 = 1$. The Beresnev function then changes to $\mathcal{B}_{[F|C]}((0, 1, y_3, y_4)) = 51 - 3y_3 - 4y_4 + 12y_3y_4$. One of the instances that such a Beresnev function corresponds to is the following one (with rows corresponding to y_1, y_3 and y_4, respectively, since we have deleted the row corresponding to y_2).

$$[S|D] = \left[\begin{array}{c|c} 0 & 56 \\ 3 & 44 \\ 4 & 44 \end{array}\right].$$

It is easy to see that an optimal solution to this instance is $y_1 = y_3 = 0$ and $y_4 = 1$. So an optimal solution to the SPLP instance is $y_1 = y_3 = 0$ and $y_2 = y_4 = 1$, i.e. to set up plants at sites 1 and 3.

Hence we have reduced the size of the instance at hand, and in this case, arrive at an optimal solution to the original instance using the preprocessing rules described above.

Notice that if at any preprocessing step, we can determine that $y_k = 1$ for a certain site k, then we need not include the row corresponding to site k in our calculations, and can therefore drop this row from the extended matrix in the succeeding steps. This deletion of rows is not possible if $y_k = 0$, since we do not know beforehand the whole set of clients that be served by a plant located at this site in any equivalent instance of the SPLP.

4. Summary and Directions for Future Research

In this paper we consider a pseudo-Boolean representation of the SPLP. There are two such representations available in the literature. The one described in Hammer [11] is the oldest and has a form in which terms contain both literals and complements of literals. The one that we use is described in Beresnev [1]. The terms in this representation contain either literals or complements of literals but not both. Using Beresnev's pseudo-Boolean formulation, we first describe the concept of equivalence. We call two instances equivalent if they share the same goal function values for every solution. We show that it is possible to check the equivalence of two instances in time polynomial in the size of the problems. We next define the set of all instances equivalent to a given instance and show that it can be represented as a union of polytopes. We show that the number of non-empty polytopes, the union of which describes the whole set of equivalent instances, is exponential and bounded below by a Stirling number of the second kind. Finally we show how we can use the concept of equivalence to recognize whether an instance at hand is polynomially solvable, and to perform pre-processing operations that can reduce the size of the problem before it is solved using exact methods.

One clear extension of this paper is in algorithm development. The RECOGNIZE heuristic is a simple, and not particularly efficient heuristic to check whether a given instance is equivalent to an instance with quasiconcave transportation cost matrices. There is a need for systematic development of faster and more powerful recognition algorithms to check if a given instance is equivalent to a known polynomially solvable special case of the SPLP. One can also develop pre-processing rules that are stronger than the one developed in this paper. The second important direction in this type of research is to exploit equivalences to develop exact algorithms. The data-correcting algorithm (see Goldengorin *et al.* [10]) is a strong candidate algorithm for this type of research.

Another extension of this research is to examine the polyhedral properties of $\mathcal{P}_{[F|C]}$. It is interesting that this set can be represented as a union of polytopes. However, the topology of the set, i.e. the intersection among $P_{[F|C],\Pi}$'s for various ordering matrices Π has not been studied. A study of the tightness of the bound for the number of non-empty polytopes in $\mathcal{P}_{[F|C]}$ is also interesting.

A third direction of research, and one that we plan to pursue in the immediate future is to use of the properties of $\mathcal{P}_{[F|C]}$ for post-optimality analysis. Since the polytopes are defined in terms of the coefficients in equivalent instances, it is easy to use these with various objective functions to perform heuristic sensitivity and stability analysis. The analyses are of a heuristic nature since each polytope represents only a part of the full set of equivalent instances. An important advantage of using this approach for post-optimality analysis is that we can use information regarding inter-connections of the various coefficients (contrary to conventional post-optimality analysis where the variations of various problem coefficients are assumed to be independent of each other).

References

[1] V. L. Beresnev. On a Problem of Mathematical Standardization Theory. Upravliajemyje Sistemy 11, 43–54, 1973 (in Russian).

[2] V. L. Beresnev, E. Kh. Gimadi, V. T. Dementyev. Extremal Standardization Problems, Novosibirsk, Nauka, 1978 (in Russian).

[3] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. The Uncapacitated Facility Location Problem. Ch.3, Discrete Location Theory, R. L. Francis and P. B. Mirchandani (eds), Wiley-Interscience, New York, 1990.

[4] N. Christofides. Graph Theory: An Algorithmic Approach. Academic Press Inc. Ltd., London, 1975.

[5] F. N. David, M. G. Kendall and D. E. Barton. Symmetric Function and Allied Tables, Cambridge, 1966.

[6] P. M. Dearing, P. L. Hammer, B. Simeone, Boolean and Graph Theoretic Formulations of the Simple Plant Location Problem. Transportation Science 26, 138–148, 1992.

[7] B. I. Goldengorin. A Decomposition Algorithm for the Unification Problem and New Polynomially Solvable Cases. Soviet Math. Dokl. 33, 578–583, 1986.

[8] B. I. Goldengorin. On the Exact Solution of Standardization Problems by Correcting Algorithms. Soviet Phys. Dokl. 32, 432–434, 1987.

[9] B. Goldengorin. Requirements of Standards: Optimization Models and Algorithms. ROR, Hoogezand, The Netherlands, 1995.

[10] B. Goldengorin, G. Sierksma, G. A. Tijssen, M. Tso. The Data-Correcting Algorithm for Minimization of Supermodular Functions. Management Science 45, 1539–1551, 1999.

[11] P. L. Hammer. Plant Location — A Pseudo-Boolean Approach. Israel Journal of Technology 6, 330–332, 1968.

[12] P. C. Jones, T. J. Lowe, G. Muller, N. Xu, Y. Ye and J. L. Zydiak. Specially Structured Uncapacitatied Facility Location Problems. Operations Research 43, 661–669, 1995.

[13] N. Lindquist and G. Sierksma. Extensions of Set Partitions. Journal of Combinatorial Theory, Series A 31, 190–198, 1981.

[14] D. W. Pentico. The Assortment Problem with Nonlinear Cost Functions. Operations Research 24, 1129–1142, 1976.

[15] D. W. Pentico. The Discrete Two-Dimensional Assortment Problem. Operations Research 36, 324–332, 1988.

[16] C.S. Revelle and G. Laporte. The Plant Location Problem: New Models and Research Prospects. Operations Research 44, 864–874, 1996.

[17] A. Tripathy, Süral, and Y. Gerchak. Multidimensional Assortment Problem with an Application. Networks 33, 239–245, 1999.

[18] V. E. Veselovsky. Some Algorithms for Solution of a Large-Scale Allocation Problem. Ekonom. Mat. Metody 12, 732–737, 1977 (in Russian).