University of Groningen

Equivalent instances of the simple plant location problem

Albdaiwi, Bader F.; Goldengorin, Boris; Sierksma, Gerard

Published in:
Computers & Mathematics with Applications

DOI:
10.1016/j.camwa.2008.10.081

**IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite fro
it. Please check the document version below.**

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2009

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Albdaiwi, B. F., Goldengorin, B., & Sierksma, G. (2009). Equivalent instances of the simple plant location problem. *Computers & Mathematics with Applications*, *57*(5), 812-820.
https://doi.org/10.1016/j.camwa.2008.10.081

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license.
More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediate
and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons t
number of authors shown on this cover page is limited to 10 maximum.*

Download date: 11-10-2022

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Equivalent instances of the simple plant location problem

Bader F. AlBdaiwi [a,*], Boris Goldengorin [b], Gerard Sierksma [b]

[a] *Department of Mathematics and Computer Science, Kuwait University, Kuwait*

[b] *Department of Operations, University of Groningen, The Netherlands*

ARTICLE INFO

Article history:
Received 22 July 2008
Accepted 16 October 2008

Keywords:
Simple plant location problem
Pseudo-Boolean function
Polytopes
Equivalence
Polynomially solvable special cases

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.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

In this paper, we study the Simple Plant Location Problem (SPLP). A detailed introduction to this problem appears in Cornuejols et al. [1]. The goal of the problem is one of determining a 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 dispatching (see Christofides [2]), assortment (see Beresnev et al. [3], Goldengorin [4], Jones et al. [5], Pentico [6], Tripathy et al. [7]) and is a subproblem for various location analysis problems (see Revelle and Laporte [8]). 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 this problem are based on branch and bound techniques (see Cornuejols [1] 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 [9] 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 [10] a different pseudo-Boolean form is presented, 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, which we term as the *Hammer–Beresnev function*.

Since the SPLP is hard, unless $\mathcal{P} = \mathcal{NP}$, SPLP instances cannot be solved efficiently. However there are classes of SPLP instances that can be solved polynomially. These classes are often defined by special relations between their fixed and transportation costs. In this regard, an open question in Burkard [11, p. 155] can be paraphrased as follows:

[*] Corresponding author.
E-mail addresses: bdaiwi@sci.kuniv.edu.kw (B.F. AlBdaiwi), b.goldengorin@rug.nl (B. Goldengorin), g.sierksma@rug.nl (G. Sierksma).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.081

> "Given fixed and transportation costs for a SPLP instance I, how can one find a polynomially solvable instance I' such that the value of the optimal solution of I' is as close as possible to that of the optimal solution of the original instance I?"

This paper provides an approach to answer this question by developing the concept of equivalent SPLP instances through Hammer–Beresnev functions. Two SPLP instances are said to be equivalent if they are of the same size and if the objective function values corresponding to the same solution are the same for both instances, even though the fixed and transportation costs of the two instances are different. The main contribution of the paper is that, given a SPLP instance I and the description of a class of polynomially solvable instances for the SPLP, it presents a heuristic that tries to output a modified polynomially solvable SPLP instance I', often with drastically different instance data values, whose optimal solution itself is identical to the optimal solution of I.

The remainder of this paper is organized as follows. In Section 2 we use Beresnev's pseudo-Boolean formulation of the SPLP, and develop the concept of equivalent instances. 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 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 $I = \{1, 2, \dots, m\}$ of sites in which plants can be located, and $J = \{1, 2, \dots, n\}$ of clients. Let $(i, j) \in I \times J$. A vector $F = (f_i)$ of fixed costs for setting up plants at site i, a matrix $C = [c_{ij}]$ of transportation costs from i to 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, 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 the 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 $\Pi_j = (\pi_{1j}, \dots, \pi_{mj})^{\mathrm{T}}$ defines a permutation of $1, \dots, m$. Given a transportation matrix C, the set of all ordering matrices Π such that $c_{\pi_{1j}j} \le c_{\pi_{2j}j} \le \cdots \le c_{\pi_{mj}j}$, 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

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

Then, for each $j \in J$,

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

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 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 take 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

$$\pi_{kl} = \psi_{(k+1)l}, \tag{5}$$

$$\pi_{(k+1)l} = \psi_{kl}, \tag{6}$$

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

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. *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 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 $\{1, 2, \dots, n\}$ and α_T can assume arbitrary values. We call a pseudo-Boolean polynomial $P(\mathbf{y})$ a *Hammer–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 each $\mathbf{y} \in \{0, 1\}^m$. We denote a Hammer–Beresnev function corresponding to a given SPLP instance $[F|C]$ by $\mathcal{B}_{[F|C]}(\mathbf{y})$ and define it as

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

Theorem 3. *A general pseudo-Boolean function is a Hammer–Beresnev function if, and only if,*

(a) *all coefficients of the pseudo-Boolean function except those of the linear terms are non-negative, and*
(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 Hammer–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 $\Delta c[k, j]$ is negative for some $j \in \{1, \dots, n\}$. 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 contradicts the fact that $\Pi \in perm(C)$. The logic above holds true for all members of $perm(C)$ as a consequence of Lemma 1. ■

We can formulate (1) in terms of Hammer–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

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

The Hammer–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 Hammer–Beresnev function. This is due to the fact that we can aggregate terms in the Hammer–Beresnev function. If two SPLP instances of the same size have the same Hammer–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 $\mathcal{B}_{[F|C]} = \mathcal{B}_{[S|D]}$. Hammer–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

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

have different Hammer–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 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 inequalities.

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

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

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

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

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

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

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

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

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

Consider the instance in (11). Then $P_{[F|C],\Pi_1}$ (where Π_1 is defined in (12) is defined by the following system, corresponding to (17)–(21).

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.

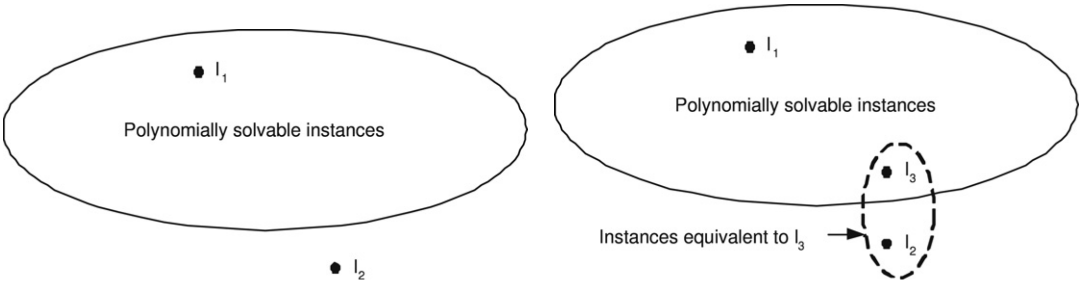
Lemma 4. *Given a SPLP instance $[F|C]$ of size $m \times (n+1)$, a lower bound to the number of non-empty polytopes $P_{[F|C],\Pi}$, $\Pi \in perm(E)$ is a Stirling number of the second kind,*

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

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 Hammer–Beresnev function only if the two columns are distinct. So in order for an instance $[S|D]$ to have a Hammer–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

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


I1
Polynomially solvable instances
I2
I1
Polynomially solvable instances
I3
Instances equivalent to I3
I2

(a) Conventional approach.

(b) Our extended approach.

Fig. 1. Handling polynomially solvable special cases.

with boundary conditions

$$F(k,k)=k!,$$
$$F(k-1,k)=0.$$

The solution to this set of recurrence equations is

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

(refer David et al. [12], 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 the 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 polynomially 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 Hammer–Beresnev functions for recognizing whether a problem is polynomially solvable.

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 Fig. 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 Fig. 1(a)) then conventional approaches terminate, reporting that the instance is not polynomially solvable.

Our approach to the problem of solving SPLPs 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 Fig. 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 Fig. 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 Fig. 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. [3], Goldengorin [4]) and references within, Jones et al. [5]). 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 $\langle r[1],\dots,r[m]\rangle$, of rows and an index k_j, $1\le k_j\le m$, for each column $j\in J$, such that

$$a_{r[1]j}\le\cdots\le a_{r[k_j]j}\ge\cdots\ge a_{r[m]j}.$$

A $m\times n$ matrix $A=[a_{ij}]$ is called *2-compact* if each of its submatrices, obtained by deleting any subset of rows or columns, has, at most, two rows containing elements which are minimal in their columns.

Let $\Pi^{(k)}$ be the set of all elements in the first k rows of a $m\times n$ ordering matrix $\Pi\in perm(A)$. The following sufficient conditions of 2-compactness for $m\times n$ matrix A are only necessary conditions of quasiconcavity of matrix A. A is said to be 2-*compact* iff

$$|\Pi^{(k+1)}-\Pi^{(k)}|\le 2\quad\text{for } k=1,\dots,m-1.$$

The following example of 4 × 2 matrix A shows that minimal elements in every submatrix of A in at most two rows are contained despite this matrix not being a quasiconcave matrix, but a 2-compact matrix.

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 3 & 3 \\ 4 & 1 \end{bmatrix} \quad \text{and} \quad \Pi = \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 3 & 3 \\ 4 & 1 \end{bmatrix}. \tag{22}$$

In Goldengorin [4] 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.

The RECOGNIZE heuristic uses the concept of equivalence and the observation in Goldengorin [4] to recognize whether a given transportation matrix C is quasiconcave.

```
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 ($m \times n$)
     $\Pi$         : 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 $\Pi \in perm(C)$;
  2        unused := $\{1, 2, \ldots, m\}$;
  3        for ($r := 1$ to $m$) do
  4        begin /* iteration */
  5                if (row $r$ in $\Pi$ 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 ($\pi_{rk} \notin unused \setminus \{i, j\}$) then
  14                           begin
  15                                   $t := \pi_{rk}$;
  16                                   $\pi_{rk} := i$;
  17                                   if ($P_{[F|C],\Pi} = \emptyset$) then
  18                                           $\pi_{rk} := j$;
  19                                   if ($P_{[F|C],\Pi} = \emptyset$) then
  20                                           go to Statement 9;
  21                           end
  22                   end
  23           end
  24           if ($|\Pi^{(r)} - \Pi^{(r-1)}| > 2$) then /* Assume that $\Pi^{(0)} = \emptyset$ */
  25                   return "NO";
  26       end /* Iteration */
  27       return "YES";
  end
```

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 Π_r for which $|\Pi_r^{(r)} - \Pi_r^{(r-1)}| \leq 2$. 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.

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 two units from the fixed cost of the second site to the cost of transporting a unit from site 2 to client 3.

$$[S|D] = \left[\begin{array}{c|ccccc} 7 & 7 & 15 & 10 & 7 & 10 \\ 1 & 10 & 17 & 6 & 11 & 22 \\ 3 & 16 & 7 & 6 & 18 & 14 \\ 6 & 11 & 7 & 6 & 12 & 8 \end{array}\right]. \tag{23}$$

This instance is polynomially solvable using the observation in Goldengorin [4], 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.

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 [9] 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 [10]. The terms in this representation contain either literals or complements of literals, but not both. We call this representation the Hammer–Beresnev representation. Using the Hammer–Beresnev representation of the SPLP, we first describe the concept of equivalence. We call two instances equivalent if they are of the same size and if each of their feasible solutions have the same objective function value in both instances. We show that it is possible to check the equivalence of two instances with very different fixed cost vectors and transportation cost matrices 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 can be transformed to one that is polynomially solvable. For the SPLP, this result is a step toward answering an open question in Burkard et al. [11, p. 155, open problem (1)] which asks whether it is possible to find a polynomially solvable instance to a combinatorial optimization problem whose optimal solution is closest in value to a given instance of the problem.

There are several interesting extensions to the work done in this paper. One clear extension is in the development of heuristics. The RECOGNIZE heuristic presented here 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.

A second important direction in this type of research is to exploit equivalences to develop exact algorithms. The data-correcting algorithms (see Goldengorin et al. [14,15], Ghosh et al. [16]) and tolerance based algorithms (see Goldengorin et al. [17,18], Turkensteen et al. [19]) are strong candidate algorithms for this type of research.

A third 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 fourth 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).

Acknowledgment

The authors thank Professor Alex Belenky for helpful discussion and suggestions.

References

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

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

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

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

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

[6] D.W. Pentico, The assortment problem: a survey, European Journal of Operational Research 190 (2008) 295–309.

[7] A. Tripathy, Süral, Y. Gerchak, Multidimensional Assortment problem with an application, Networks 33 (1999) 239–245.

[8] C.S. Revelle, G. Laporte, The plant location problem: New models and research prospects, Operations Research 44 (1996) 864–874.

[9] P.L. Hammer, Plant location — A pseudo-Boolean approach, Israel Journal of Technology 6 (1968) 330–332.

[10] V.L. Beresnev, On a problem of mathematical standardization theory, Upravliajemyje Sistemy 11 (1973) 43–54 (in Russian).

[11] R.E. Burkard, B. Klinz, R. Rudolf, Perspective of monge properties, Discrete Applied Mathematics 70 (1996) 95–161.

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

[13] N. Lindquist, G. Sierksma, Extensions of set partitions, Journal of Combinatorial Theory, Series A 31 (1981) 190–198.

[14] B. Goldengorin, G. Sierksma, G.A. Tijssen, M. Tso, The data-correcting algorithm for minimization of supermodular functions, Management Science 45 (1999) 1539–1551.

[15] B. Goldengorin, G.A. Tijssen, D. Ghosh, G. Sierksma, A data correcting algorithm for the simple plant location problem, Journal of Global Optimization 25 (4) (2003) 377–406.

[16] D. Ghosh, B. Goldengorin, G. Sierksma, Data Correcting Algorithms in Combinatorial Optimization, in: D.-Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Vol. 5, Supplement Volume B, Springer, 2005, pp. 1–53.

[17] B. Goldengorin, G. Sierksma, M. Turkensteen, Tolerance based algorithms for the ATSP, Lecture Notes in Computer Science 3353 (2004) 222–234.

[18] B. Goldengorin, G. Jager, P. Molitor, Tolerances applied in combinatorial Optimization, Journal of Computer Science 2 (9) (2006) 716–734.

[19] M. Turkensteen, D. Ghosh, B. Goldengorin, G. Sierksma, Tolerance-based branch and bound algorithms for the ATSP, European Journal of Operational Research 189 (2008) 775–778.