Data aggregation for p-median problems

AlBdaiwi, Bader F.; Ghosh, Diptesh; Goldengorin, Boris

Published in:
Journal of Combinatorial Optimization

DOI:
10.1007/s10878-009-9251-8

Publication date:
2011

J Comb Optim (2011) 21: 348–363
DOI 10.1007/s10878-009-9251-8

Data aggregation for p-median problems

**Bader F. AlBdaiwi · Diptesh Ghosh ·
Boris Goldengorin**

Published online: 27 June 2009

Abstract In this paper, we use a pseudo-Boolean formulation of the p-median problem and using data aggregation, provide a compact representation of p-median problem instances. We provide computational results to demonstrate this compactification in benchmark instances. We then use our representation to explain why some p-median problem instances are more difficult to solve to optimality than other instances of the same size. We also derive a preprocessing rule based on our formulation, and describe equivalent p-median problem instances, which are identical sized instances which are guaranteed to have identical optimal solutions.

Keywords p-Median problem · Pseudo-Boolean polynomial · Data aggregation · Equivalent instances

1 Introduction

The p-Median Problem (PMP) is a well known problem among the class of minisum location-allocation problems. It is defined as follows.

> Given a set I of m potential facilities, a set J of n users (or customers), a distance function $c : I \times J \to \Re_+$, and a constant $p \leq m$, determine which p facilities to open so as to minimize the sum of the distances from each user to its closest open facility.

B.F. AlBdaiwi (✉)
Department of Mathematics and Computer Science, Kuwait University, Kuwait, Kuwait
e-mail: bdaiwi@sci.kuniv.edu.kw

D. Ghosh
P&QM Area, Indian Institute of Management, Ahmedabad, India

B. Goldengorin
Department of Operations, University of Groningen, Groningen, The Netherlands

The PMP is NP-hard (Kariv and Hakimi 1979), and has many applications in location analysis (see Revelle et al. 2008 and references within) and cluster analysis (see e.g., Mirkin 2005 and references within). A detailed introduction to this problem and solution methods appear in Reese (2006) and Mladenovic et al. (2007).

The PMP is a generalization of classical Fermat problem defined on three distinct points in a plane, where the purpose is to find a median point in the plane such that the sum of the distances from each of the points to the median point induced by the triangle spanned on these points is minimized. It is also a generalization of the Weber problem (see Weber 1909) which generalizes the Fermat problem by allowing n points on a plane, each with a certain weight associated with it (to model client demands), and finding the weighted median point induced by the polygon spanned on the n points. For a discussion of Fermat and Weber problems see Krarup and Vajda (1997).

Hakimi (1964, 1965) generalized the Weber problem to the problem of finding an *absolute median* on a graph that minimizes the sum of the weighted distances between that absolute median and the vertices of the graph. Hakimi has shown that an optimal absolute median is always located at a vertex of the graph. Similarly Hakimi (1965) generalized the absolute median to the PMP, again providing a discrete representation of a continuous problem by restricting the set of feasible solutions to the vertices.

Another common problem within the class of minisum location-allocation problems is the Simple Plant Location Problem (SPLP), also called the Uncapacitated Facility Location Problem (UFLP) or the warehouse location problem (see e.g. Revelle et al. 2008). The SPLP is similar to the PMP, and the methods used to solve one are often adapted to solve the other. The objective function of the SPLP is one of determining the cheapest method of meeting the demands of a set of clients $J = \{1, \dots, n\}$ from plants that can be located at a subset of a set of candidate sites $I = \{1, \dots, m\}$. The costs involved in meeting the client demands include the fixed cost of setting up a plant at a chosen site, and the per unit transportation cost of supplying a given client from a plant located at a given site. Each plant is assumed to be able to supply an unlimited quantity of demand. The PMP and SPLP differ in the following details. First, the SPLP involves a fixed cost for locating a facility at a given location, and the PMP does not. Second, unlike the PMP, SPLP does not have a constraint on the maximum number of facilities. Typical SPLP formulations separate the set of candidate sites from the set of clients. In the PMP these sets are identical, i.e. $I = J$. A recent computational study by Avella et al. (2007) shows that PMP instances with $|I \times J| > 360\,000$ are difficult for commercial MIP codes, mainly due to memory restrictions.

These two problems are well known problems in cluster analysis (see e.g., Mirkin 2005 and references within). Both form underlying models in several combinatorial problems, such as set covering, set partitioning, information retrieval, simplification of logical Boolean expressions, airline crew scheduling, vehicle dispatching (see Goldengorin et al. 2003b and references within) and are subproblems for various location analysis problems (see Revelle et al. 2008).

As is the case with PMP, each instance of the SPLP admits an optimal solution in which each client is satisfied by exactly one plant. In Hammer (1968) this fact is used to derive a pseudo-Boolean representation of the SPLP. The pseudo-Boolean polynomial developed in that work has monomials that contain both a literal and its complement. Subsequently, in Beresnev (1973) a different pseudo-Boolean form has been developed in which each monomial contains only literals or only their complements but not both. We find this form easier to manipulate, and hence use Beresnev's formulation in this paper which we term as *Hammer-Beresnev polynomial*.

Based on the Hammer-Beresnev formulation for the SPLP, Goldengorin et al. (2003a) have derived a pegging rule within a branch-and-peg algorithm. Computational experiments reveal the advantage of using the branch and peg algorithm, whose computation times are significantly lower than that of comparable branch and bound techniques (see Revelle et al. 2008). Goldengorin et al. (2003b) have incorporated the Hammer-Beresnev function in the data correction approach. These authors present reduction rules that are significantly more powerful than those suggested by Khumawala (1972).

Problem reduction is a common technique in integer programming and combinatorial optimization; see, for example, Briant and Naddef (2004) and references within. Classical reduction techniques for PMP instances are based either on good lower bounds (see e.g., Briant and Naddef 2004) or on reduction tests (see e.g., Avella and Sforya 1999). In this paper, we present two reduction techniques for PMP instances using a pseudo-Boolean formulation of PMP due to Hammer (1968) and Beresnev (1973).

Since the PMP is NP-hard and many polynomially solvable special cases are well known in the literature (see e.g., the 1-median problem on a cactus in Burkard and Krarup 1998) it is interesting to examine the use of polynomially solvable special cases of the PMP to find either an exact or approximate solution to a PMP instance which is not polynomially solvable. Burkard et al. (1996), p. 155 presents this question as an open problem. For the PMP, the question posed in Burkard et al. (1996) can be phrased as follows:

> "Suppose we are given a PMP instance defined on a cost matrix C which does not belong to a polynomially solvable class of PMP instances. Is it possible to modify C into a cost matrix D belonging to a polynomially solvable class of PMP such that an optimal solution to the original problem instance is as close as possible to the optimal solution of the modified instance?"

In this paper we show that the pseudo-Boolean formulation of the PMP allows us to find such modifications if the polynomially solvable class of PMP instances is defined algebraically in terms of the elements in its cost matrix. For this, we describe the concept of equivalent instances. Moreover, we reduce the problem of finding an equivalent cost matrix D with the minimum number of columns to the given matrix C to the well known Dilworth's decomposition theorem (see e.g. Theorem 14.2 in Schrijver 2003).

While this paper does not suggest any new algorithm for solving the PMP it presents some fundamental properties of PMP derived from its pseudo-Boolean representation. Our paper is organized as follows. In Sect. 2 of this paper, we adjust the Hammer-Beresnev pseudo-Boolean formulation of the Simple Plant Location Problem to the PMP, and show that reducing of similar monomials in the Hammer-Beresnev pseudo-Boolean polynomial leads to the aggregation of entries in the given

PMP instance. Section 3 describes the reductions and truncations in the PMP. In Sect. 4, based on the truncation of degree of Hammer-Beresnev polynomial from $(m-1)$ to $(m-p)$ we are further able to aggregate the entries in the Hammer-Beresnev polynomial use it to develop rules for preprocessing PMP instances. We also show that the pseudo-Boolean representation allows us to comment on relative difficulties in obtaining provably optimal solutions to different PMP instances. Both Sects. 2 and 4 include computational analysis of benchmark instances similar to instances used in Avella et al. (2007). Section 5 defines the concept of equivalent instances and describes an algebraic method of modifying the cost matrix of a PMP instance without disturbing the optimality of any optimal solution to the original instance. This answers Burkard et al.'s open problem affirmatively in the context of the PMP. It also indicates the relationships with the minimum number of aggregated columns and Dilworth's decomposition theorem. Section 6 summarizes the main results of the paper, and points to directions for future research.

2 A pseudo-Boolean formulation of the PMP

Recall that given sets $I=\{1,2,\ldots,m\}$ of sites in which plants can be located, $J=\{1,2,\ldots,n\}$ of clients, a matrix $C=[c_{ij}]$ of costs of supplying each $j\in J$ from each $i\in I$, the number p of plants to be opened, and unit demand at each client site, the p-Median Problem (PMP) is one of finding a set $S\subseteq I$ with $|S|=p$, such that the total cost

$$f_C(S)=\sum_{j\in J}\min\{c_{ij}|i\in S\}$$

is minimized. An instance of the problem is described by an $m\times n$ matrix $C=[c_{ij}]$ and the number $1\le p\le |I|$. We assume that the entries of C are nonnegative and finite. The PMP is thus the problem of finding

$$S^\star\in\arg\min\{f_C(S):\emptyset\subset S\subseteq I,\ |S|=p\}. \tag{1}$$

We now formulate the objective function $f_C(S)$ of the PMP in (1) in terms of a pseudo-Boolean polynomial $\mathcal{B}_C(\cdot)$. A pseudo-Boolean polynomial is a mapping $f:\{0,1\}^n\to\Re$. All pseudo-Boolean polynomials can be uniquely represented as *multi-linear polynomials* of the form (see e.g. Boros and Hammer 2002)

$$f(\mathbf{y})=\sum_{S\subseteq I}c_S\prod_{i\in S}y_i. \tag{2}$$

Two monomials are said to be *similar* if their terms are identical. A pseudo-Boolean polynomial is said to be in the *reduced form* if it does not contain similar monomials. The process of creating a reduced form of a general pseudo-Boolean polynomial by summing up the coefficients of similar monomials is called *reduction of similar monomials*.

Our pseudo-Boolean formulation is a penalty-based formulation. For client j, let $\Pi^j=(\pi_{ij},\ldots,\pi_{mj})$ be an ordering of $1,\ldots,m$ such that $c_{\pi_{ij}j}\le c_{\pi_{kj}j}$ if $i<k$ for all $i,k\in\{1,\ldots,m\}$. Also, let $\Delta^j=(\delta_{1j},\ldots,\delta_{mj})$, where $\delta_{1j}=c_{\pi_{1j}j}$, and $\delta_{rj}=$

$c_{\pi_{rj}j} - c_{\pi_{(r-1)j}j}$ for $r = 2, \ldots, m$. If the plant at location π_{1j} is opened, then the cost of meeting a unit demand at j is $c_{\pi_{1j}j}$, i.e., δ_{1j}. If it is closed, and the plant at location π_{2j} is open, then the cost of meeting a unit demand at j is $c_{\pi_{2j}j}$, i.e., $c_{\pi_{1j}j} + \delta_{2j}$. In other words, if we decide not to open a plant at location π_{1j}, then we incur a penalty of δ_{2j} while meeting a unit demand at j. We call Δ^j the difference vector for client j.

Let us define an m-vector $\mathbf{y} = (y_1, \ldots, y_m)$, where $y_i = 0$ if a plant is opened at location i, and 1 otherwise. Then the cost of satisfying a unit demand at j is given by the pseudo-Boolean polynomial

$$f_C^j(\mathbf{y}) = \delta_{1j} + \sum_{k=2}^{m} \delta_{kj} \cdot \prod_{r=1}^{k-1} y_{\pi_{rj}}.$$

For notational convenience, in order to aggregate the costs for all clients and obtain a pseudo-Boolean polynomial to describe $f_C(\mathbf{y})$, we define an $m \times n$ *ordering matrix* $\Pi = [\pi_{ij}]$ and an $m \times n$ *difference matrix* $\Delta = [\delta_{ij}]$. The jth column of Π is the transpose of Π^j, and the jth column of Δ is the transpose of Δ^j. In terms of Π and Δ, the objective function $f_C(S)$ of PMP is

$$\mathcal{B}_C(\mathbf{y}) = \sum_{j=1}^{n} \left\{ \delta_{1j} + \sum_{k=2}^{m} \delta_{kj} \cdot \prod_{r=1}^{k-1} y_{\pi_{rj}} \right\}. \tag{3}$$

Note that Π is unique for a PMP instance if and only if the entries in each of the columns of C are distinct. Δ however is unique for a PMP instance. Note also, that the polynomial in the right hand side of (3) can have similar monomials. In the remainder of the paper, we assume that such monomials are reduced in the expression of $\mathcal{B}_C(\mathbf{y})$.

The set of all ordering matrices for a PMP instance with cost matrix C is denoted by $perm(C)$. It is easy to see that the representation of $\mathcal{B}_C(\mathbf{y})$ developed in (3) is identical for all ordering matrices in $perm(C)$. This result is derived in AlBdaiwi et al. (2009) for the SPLP; the proof of the result for PMP follows along similar lines.

We call a pseudo-Boolean polynomial $f(\mathbf{y})$ a *Hammer-Beresnev polynomial* if there exists a PMP instance with cost matrix C and $\Pi \in perm(C)$ such that the objective function of the PMP instance can be represented by $f(\mathbf{y})$ for each $\mathbf{y} \in \{0, 1\}^m$. Therefore the function $\mathcal{B}_C(\mathbf{y})$ developed in (3) is a Hammer-Beresnev polynomial. Theorem 1 describes the condition under which a general pseudo-Boolean polynomial is a Hammer-Beresnev polynomial.

Theorem 1 *A general pseudo-Boolean polynomial is a Hammer-Beresnev polynomial if and only if all its coefficients are non-negative.*

Proof The "if" statement is trivial. In order to prove the "only if" statement, consider a PMP instance defined by the cost matrix C, an ordering matrix $\Pi \in perm(C)$, and a Hammer-Beresnev polynomial $\mathcal{B}_C(\mathbf{y})$ in which there is a monomial of degree k with a negative coefficient. Since monomials in $\mathcal{B}_C(\mathbf{y})$ are contributed by the elements of C only, a monomial with a negative coefficient implies that $\delta_{k,j}$ is negative for some $j \in \{1, \ldots, n\}$. But this contradicts the fact that $\Pi \in perm(C)$. $\square$

We can formulate (1) in terms of Hammer-Beresnev polynomials as

$$\mathbf{y}^\star \in \arg\min\left\{\mathcal{B}_C(\mathbf{y}) : \mathbf{y} \in \{0,1\}^m, \sum_{i=1}^{m} y_i = m - p\right\}. \tag{4}$$

3 Reductions and truncations in the PMP

Hammer-Beresnev polynomials allow a compact description of p-median problems, since they allow reduction of similar monomials. The polynomial obtained from the original Hammer-Bersnev polynomial through reduction of similar monomials is called the *reduced Hammer-Beresnev polynomial*. Example 1 demonstrates the reduction process.

Example 1 Consider a PMP instance with $m = 4$, $n = 5$, $p = 2$ and

$$C = \begin{bmatrix} 7 & 15 & 10 & 7 & 10 \\ 10 & 17 & 4 & 11 & 22 \\ 16 & 7 & 6 & 18 & 14 \\ 11 & 7 & 6 & 12 & 8 \end{bmatrix}. \tag{5}$$

A possible ordering matrix for this problem is given by

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

and the difference matrix is given by

$$\Delta = \begin{bmatrix} 7 & 7 & 4 & 7 & 8 \\ 3 & 0 & 2 & 4 & 2 \\ 1 & 8 & 0 & 1 & 4 \\ 5 & 2 & 4 & 6 & 8 \end{bmatrix}.$$

The Hammer-Bersnev polynomial representing total cost function for this instance in the form of (3) is

$$\begin{aligned} \mathcal{B}_C(\mathbf{y}) = {} & [7 + 3y_1 + 1y_1y_2 + 5y_1y_2y_4] \\ & + [7 + 0y_3 + 8y_3y_4 + 2y_1y_3y_4] \\ & + [4 + 2y_2 + 0y_2y_3 + 4y_2y_3y_4] \\ & + [7 + 4y_1 + 1y_1y_2 + 6y_1y_2y_4] \\ & + [8 + 2y_4 + 4y_1y_4 + 8y_1y_3y_4] \end{aligned}$$

whose monomials can be reduced to yield the reduced Hammer-Beresnev polynomial

$$\begin{aligned} \mathcal{B}_C(\mathbf{y}) = {} & 33 + 7y_1 + 2y_2 + 2y_4 + 2y_1y_2 + 8y_3y_4 \\ & + 4y_1y_4 + 11y_1y_2y_4 + 10y_1y_3y_4 + 4y_2y_3y_4. \end{aligned}$$

Table 1 Reductions of number of monomials in the objective functions of benchmark instances

Library	Instance	m	Entries in C matrix	Number of terms in $\mathcal{B}_C(\mathbf{y})$	Reduction(%)
OR	pmed15	300	90 000	17 102	81.00
OR	pmed26	600	360 000	25 083	93.03
OR	pmed40	900	810 000	30 756	96.20
ODM	BN48	42	411	329	19.95
ODM	BN1284	1284	88 542	85416	3.53
ODM	BN3773	3773	349 524	341775	2.22
ODM	BN5535	5535	666 639	654709	1.79
TSP	D657	657	430 992	367355	14.77
TSP	fl1400	1400	1 958 600	836 557	57.29
TSP	pcb3038	3038	9 226 406	5 759 404	37.58

Note that the original Hammer-Beresnev polynomial has 20 monomials including those with zero coefficients, while the reduced Hammer-Beresnev polynomial has just 10 monomials.

The compactification illustrated in Example 1 is also seen in benchmark PMP instances. Table 1 presents the extent of reduction in the number of terms in the objective function obtained by reducing similar monomials in Hammer-Beresnev polynomials for benchmark PMP instances considered in Avella et al. (2007). (Recall that in the conventional integer programming formulation of the PMP, the objective function has as many monomials as the number of entries in the C matrix.) In Table 1 the first four columns describe the benchmark instance being examined. The first column shows the library from which the instance is taken, the second column gives the name of the instance, the third column gives the number of plants considered in the problem, and the fourth provides the number of entries in the cost matrix of the instance. The three libraries considered here, i.e., OR, ODM, and TSP instances are available from OR-Library (http://mscmga.ms.ic.ac.uk/info.html), Briant and Naddef (2004), and TSP-library (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95), respectively. The fifth column in the table gives the number of terms in the reduced Hammer-Beresnev polynomial for the instance. The last column in the table shows the extent of reduction in the number of monomials in the objective function through reduction of similar monomials in the Hammer-Beresnev polynomial corresponding to the instance. For example, in the pmed15 instance, the cost matrix C had 90 000 entries, while the reduced Hammer-Beresnev polynomial had only 17 102 terms, leading to a reduction of $((90\,000 - 17\,102) \times 100)/90\,000 = 81.00\%$.

Hammer-Beresnev polynomials are easily manipulated in computer programs, refer to Goldengorin et al. (2003a) for details on data structures to store and manipulate these polynomials.

The number of monomials in the Hammer-Beresnev representation of a PMP instance can be further reduced by exploiting the fact that for any feasible solution

$\mathbf{y}$ to the PMP instance, $\sum_{i=1}^{m} y_i = m - p$. This implies that any monomial in the Hammer-Beresnev polynomial for a PMP instance which can be expressed as a constant multiplied with more than $m - p$ literals necessarily evaluates to zero. This is formalized in Theorem 2.

Theorem 2 *For any PMP instance C with $p \leq m$ the following assertions hold*:

1. *The degree of truncated Hammer-Beresnev polynomial $\mathcal{B}_{C,p}(\mathbf{y})$ is at most $m - p$*;
2. *The truncated Hammer-Beresnev polynomial is equal to the Hammer-Beresnev polynomial for any feasible solution $\mathbf{y}$ to the PMP instance, i.e. $\mathcal{B}_C(\mathbf{y}) = \mathcal{B}_{C,p}(\mathbf{y})$.*

Proof The assertions follows from the fact that in a PMP instance, exactly p components of any feasible solution $\mathbf{y}$ equal 0, and therefore, any monomial in a Hammer-Beresnev polynomial expressed as a product of a constant and $m - p + 1$ or more literals will evaluate to zero in any feasible solution. □

Example 1 (continued) Consider for example, the reduced Hammer-Beresnev polynomial derived for the PMP instance corresponding to the cost matrix in Example 1. Since $p = 2$, exactly $(4 - 2) = 2$ of the y_i will equal zero in any feasible solution. Therefore, each cubic term in the $\mathcal{B}_C(\mathbf{y})$ will evaluate to zero, and the truncated Hammer-Beresnev polynomial $\mathcal{B}_{C,p=2}(\mathbf{y}) = 33 + 7y_1 + 2y_2 + 2y_4 + 2y_1y_2 + 8y_3y_4 + 4y_1y_4$ adequately describes the PMP instance. Notice that this polynomial has only seven monomials.

As a consequence of Theorem 2, the objective function of a PMP instance with $p \leq m$ and a cost matrix C can be described using the following *truncated Hammer-Beresnev polynomial*:

$$\mathcal{B}_{C,p}(\mathbf{y}) = \sum_{j=1}^{n} \left\{ \delta_{1j} + \sum_{k=2}^{m-p} \delta_{kj} \cdot \prod_{r=1}^{k-1} y_{\pi_{rj}} \right\}. \tag{6}$$

A corollary to Theorem 2 is thus a reformulation of the definition of PMPs in terms of truncated Hammer-Beresnev polynomials.

Corollary 3 *A pseudo-Boolean representation of a PMP instance with m plants, n clients, and a cost matrix C is*

$$\mathbf{y}^{\star} \in \arg\min \left\{ \mathcal{B}_{C,p}(\mathbf{y}) : \mathbf{y} \in \{0,1\}^m, \sum_{i=1}^{m} y_i = m - p \right\}. \tag{7}$$

Proof The proof is trivial. □

It follows from Theorem 2 that the largest p entries in any column of the cost matrix C are inconsequential while obtaining an optimal solution to the PMP. This leads to the following p-truncation operation.

Definition 1 For any $p \leq m$ the column j of matrix C is called *p-truncated* if the values of the largest p elements in the column are replaced by $c_{\pi_{(m-p+1)j}j}$; ties being broken arbitrarily.

Thus the 1-truncation of a column (corresponding to $p = 1$) leaves it unchanged, while a 2-truncation of a column replaces the value of its largest entry with the value of the second largest entry in the column if the largest entry is unique, and leaves it unchanged otherwise.

Corollary 4 *In a PMP problem instance, every column of PMP matrix C may be* p-truncated *without affecting the optimality of the optimal solution to the instance.*

Proof The proof is straightforward from Theorem 2 and the discussion above. □

Example 1 (continued) Consider the PMP instance given in Example 1. A classical integer programming formulation (see e.g., Revelle et al. 2008) for the problem is given below.

Decision Variables: $\bar{y}_i = 1$ if plant i is open, and 0 otherwise, for each $i \in I$, and $x_{ij} = 1$ if client j's demand is supplied by plant i, 0 otherwise, for each $i \in I$, and $j \in J$.

Model:

$$\text{Minimize} \quad \sum_{i=1}^{4}\sum_{j=1}^{5} c_{ij}x_{ij} \tag{8}$$

subject to

$$\sum_{i=1}^{4} x_{ij} = 1, \quad j = 1, \ldots, 5 \tag{9}$$

$$x_{ij} \leq \bar{y}_i, \quad i = 1, \ldots, 4; \; j = 1, \ldots, 5 \tag{10}$$

$$\sum_{i=1}^{4} \bar{y}_i = 2 \tag{11}$$

$$y_i, \; x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, 4; \; j = 1, \ldots, 5. \tag{12}$$

This formulation has 24 binary decision variables, and 46 constraints. Its objective function (8) has 20 terms.

We now present a pseudo-Boolean formulation of the same problem. The 2-truncated pseudo-Boolean objective function for this cost matrix is $\mathcal{B}_{C,p=2}(\mathbf{y}) = 33 + 7y_1 + 2y_2 + 2y_4 + 2y_1y_2 + 8y_3y_4 + 4y_1y_4$. If we define three decision variables $z_5 = y_1y_2$, $z_6 = y_3y_4$, $z_7 = y_1y_4$, we can represent the objective function as $\mathcal{B}_{C,p=2}(\mathbf{y}) = 33 + 7y_1 + 2y_2 + 2y_4 + 2z_5 + 8z_6 + 4z_7$. Even though z_5, z_6, and z_7 can attain values of 0 or 1, given the nature of the objective of the formulation, it suffices to define them as non-negative continuous variables. The mixed Boolean mathematical program based on the pseudo-Boolean formulation is given below.

Decision Variables: $y_i = 0$ if plant i is open, and 0 otherwise, for each $i \in I$, and $z_5 = y_1 y_2$, $z_6 = y_3 y_4$, $z_7 = y_1 y_4$.

Model:

$$\text{Minimize} \quad 7y_1 + 2y_2 + 2y_4 + 2z_5 + 8z_6 + 4z_7 \tag{13}$$

subject to

$$\sum_{i=1}^{4} y_i = 2 \tag{14}$$

$$y_1 + y_2 - z_5 \leq 1 \tag{15}$$

$$y_3 + y_4 - z_6 \leq 1 \tag{16}$$

$$y_1 + y_4 - z_7 \leq 1 \tag{17}$$

$$y_i, \in \{0, 1\}, \quad i = 1, \ldots, 4 \tag{18}$$

$$z_i, \geq 0, \quad i = 5, \ldots, 7. \tag{19}$$

Note that this formulation has only 4 binary decision variables, 3 continuous decision variables and 11 constraints.

The mixed Boolean formulation based on the pseudo-Boolean formulation is practically useful in solving large PMP instances. For example, the conventional model for the fl1400 instance with $p = 400$ could not be solved to optimality using state-of-the-art branch and price and cut algorithm (see Avella et al. 2007), but our mixed Boolean pseudo-Boolean formulation of the instance could be solved to optimality in 598.48 seconds running on a personal computer with Pentium IV with a 1.8 GHz processor and 1 Gb RAM. The code was written in Microsoft Visual C++ using Cplex 8.0. The pseudo-Boolean formulation had 173 992 terms in the objective function compared to 1 958 600 terms in the objective function for the conventional formulation. The optimal cost for this instance was found to be 4914.

4 Preprocessing PMP instances

The p-truncation operation described in the previous section allows us to reduce the search space for an optimal solution to a PMP instance. In particular, it allows us to fix some of the components of an optimal solution to a given PMP instance. This preprocessing operation is formalized in Theorem 5.

Theorem 5 *Assume that in a given PMP instance with $p < m$, all the entries corresponding to a particular row i in the cost matrix C are changed when p-truncation operations are performed on all columns of C. Then there exists an optimal solution $\mathbf{y}^\star = (y_1^\star, \ldots, y_m^\star)$ to the instance with $y_i^\star = 1$.*

Proof Consider any column j of C. Since the entry for row i has changed after the p-truncation operation for this column, $\pi_{ij} > m - p + 1$. Hence there will be no monomial in the truncated Hammer-Beresnev polynomial containing y_i that is derived from column j. If this is true for all columns in C, then the truncated Hammer-Beresnev polynomial for the instance will not contain any monomial containing y_i. The result follows from this observation. □

Example 2 Consider a PMP instance with $m = 4$, $n = 5$, $p = 3$ and

$$C = \begin{bmatrix} 7 & 15 & 10 & 7 & 10 \\ 10 & 17 & 4 & 11 & 22 \\ 16 & 7 & 6 & 18 & 14 \\ 11 & 7 & 6 & 12 & 8 \end{bmatrix}. \tag{20}$$

After 3-truncation, the truncated cost matrix is

$$C_{p=3} = \begin{bmatrix} 7 & 7 & 6 & 7 & 10 \\ 10 & 7 & 4 & 11 & 10 \\ 10 & 7 & 6 & 11 & 10 \\ 10 & 7 & 6 & 11 & 8 \end{bmatrix}. \tag{21}$$

In this case, from Theorem 5, we conclude that there exists an optimal solution with $y_3 = 1$. Since $p = 3$, setting $y_3 = 1$ immediately solves the problem, and the optimal solution is found to be $\mathbf{y} = (0, 0, 1, 0)$.

Theorem 5 provides interesting insights regarding changes in the degree of difficulty of solving a PMP instance when the value of p changes. As the value of p increases, the number of entries in any column whose values are revised through the p-truncation procedure increases. So, the higher the value of p, the greater is the chance that a particular entry in a column of the cost matrix would be revised. Hence the chance that all the entries in a particular row in the C matrix are revised through the p-truncation procedure increases with increasing values of p. This implies that the chance that Theorem 5 can be applied to conclude that an optimal solution to a PMP instance would not require a particular plant to be open (i.e., $y_i^\star = 1$ for that particular plant i) increases with increasing p values. This in turn implies that for a particular cost matrix, if the value of p is increased, y_i values for a larger number of plants would be set to 1, and the instance would be easier to solve. This explains why for a particular $p_0 < m/2$, PMP instances with $p = p_0$ are more difficult to solve than instances on the same cost matrix with $p = m - p_0$ even though the number of feasible solutions to the two are identical.

Let $I^\star$ be a minimum cardinality subset of rows of C such that at least one minimum among the entries of each column of C occurs in a row in $I^\star$. Let $p^\star = |I^\star|$, and p' is the smallest p such that Theorem 5 implies $y_i^* = 1$. Then for a PMP instance with $p_1 > p^\star$, there exist optimal solutions to the instance which would have open plants that do not serve any client. Further, if $p_1 > p^\star$, the number of optimal solutions to the instance is bounded below by $\binom{m-p^\star}{p_1-p^*}$. So for such instances, proving optimality of an optimal solution to the instance becomes progressively more difficult as the value of p increases. This increase in difficulty continues until the value

Table 2 p' and $p^\star$ values for benchmark instances

Library	Instance	m	p'	$p^\star$
OR	pmed15	300	180	285
OR	pmed26	600	452	581
OR	pmed40	900	644	882
ODM	BN48	42	27	35
ODM	BN1284	1284	653	1211
ODM	BN3773	3773	3385	3742
ODM	BN5535	5535	2179	5503
TSP	D657	657	477	653
TSP	fl1400	1400	1177	1395
TSP	pcb3038	3038	3026	3033

of p increases to $(m - p^\star)/2$. After that, it becomes progressively easier to prove when p increases further. Table 2 presents p' and $p^\star$ values for benchmark instances introduced in Table 1.

5 Equivalent PMP instances

Since truncated Hammer-Beresnev polynomials allow reduction of similar monomials, for a particular value of p, it is possible for multiple PMP instances with different cost matrices to have the same truncated Hammer-Beresnev polynomial representation of their objective functions. If two PMP instances of the same size, i.e., with same values of m and n, have the same truncated Hammer-Beresnev polynomial representation of their objective functions, then their set of feasible solutions are identical, and the same solution has identical costs in the two instances. In particular, optimal solutions to the two instances are identical and have identical costs. Such PMP instances are called *p-equivalent*. Formally, p-equivalent PMP instances are defined as follows.

Definition 2 Two PMP instances defined on cost matrices C and D are called *p-equivalent* if C and D are of the same size and if $\mathcal{B}_{C,p}(\mathbf{y}) = \mathcal{B}_{D,p}(\mathbf{y})$.

Remark 1 If two equal sized PMP instances defined on cost matrices C and D are p_0-equivalent, then for each ordering matrix $\Pi^C \in perm(C)$, there exists an ordering matrix in $\Pi^D \in perm(D)$ such that the first p_0 rows of Π^C are identical to the first p_0 rows of Π^D.

Since Hammer-Beresnev polynomials of PMP instances can be generated in polynomial time, and have a number of monomials that is polynomial in the size of the instance, it is possible to check p-equivalence of two instances in polynomial time for any $p \le m$, even though the PMP itself is a $\mathcal{NP}$-hard problem.

The condition of p-equivalence provided in Definition 2 is only a sufficient condition for two PMP instances to have the same optimal solution. Example 3 illustrates that it is not a necessary condition.

Example 3 Consider two PMP instances with $m = n = 2$, $p = 1$ and cost matrices

$$C = \begin{bmatrix} 3 & 3 \\ 5 & 5 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}.$$

Clearly, the truncated Hammer-Beresnev functions $\mathcal{B}_{C,p}(\mathbf{y}) = 6 + 4y_1$ and $\mathcal{B}_{D,p}(\mathbf{y}) = 2 + 2y_1$ are different. However, both instances have the same unique optimal solution, (0, 1).

We next characterize the set of all PMP instances defined on cost matrix D that are p-equivalent to a given PMP instance defined on cost matrix C. This set is can be defined as

$$\mathcal{P}_{C,p} = \left\{ D \in \Re_+^{m\times n} : \mathcal{B}_{C,p} = \mathcal{B}_{D,p} \right\}. \tag{22}$$

$\mathcal{P}_{C,p}$ can be rewritten as

$$\mathcal{P}_{C,p} = \bigcup_{\Pi\in perm(C)} P_{C,p,\Pi},$$

and

$$P_{C,p,\Pi} = \left\{ D \in \Re_+^{m\times n} : \mathcal{B}_{C,p} = \mathcal{B}_{D,p}, \Pi \in perm(D) \right\}. \tag{23}$$

We show that the set $P_{C,p,\Pi}$ can be described by a system of linear inequalities.

Let us choose $\Psi = [\psi_{ij}] \in perm(C)$ and $\Pi = [\pi_{ij}] \in perm(D)$ such that the first p rows of Ψ are identical to the first p rows of Π. Actually the choice of the particular Ψ and Π is unimportant since the truncated Hammer-Beresnev polynomials for all permutations within $perm(\cdot)$ are in a PMP instance are identical (see AlBdaiwi et al. 2009). Let the difference matrices corresponding to C and D be Δ^C and Δ^D respectively. The truncated Hammer-Beresnev polynomial for C and D are

$$\mathcal{B}_{C,p}(\mathbf{y}) = \sum_{j=1}^{n} \delta_{1j}^C + \sum_{j=1}^{n}\sum_{k=2}^{m-p} \delta_{kj}^C \cdot \prod_{r=1}^{k-1} y_{\psi_{rj}}$$

and

$$\mathcal{B}_{D,p}(\mathbf{y}) = \sum_{j=1}^{n} \delta_{1j}^D + \sum_{j=1}^{n}\sum_{k=2}^{m-p} \delta_{kj}^D \cdot \prod_{r=1}^{k-1} y_{\pi_{rj}}$$

respectively.

$P_{C,p}$ is characterized by equating coefficients of corresponding monomials in $\mathcal{B}_{C,p}(\mathbf{y})$ and $\mathcal{B}_{D,p}(\mathbf{y})$. Equating the constant term, we have

$$\sum_{j=1}^{n} \delta_{1j}^{D} = \sum_{j=1}^{n} \delta_{1j}^{C}. \tag{24}$$

Equating the coefficients in similar linear and nonlinear monomials we obtain

$$\sum_{\{\psi_{1j},\dots,\psi_{(k-1)j}\}=\{\pi_{1j},\dots,\pi_{(k-1)j}\}} \delta_{kj}^{D} - \delta_{kj}^{C} = 0 \quad k = 2,\dots,m-p;\ j = 1,\dots,n. \tag{25}$$

Finally, since $\mathcal{B}_{D,p}(\mathbf{y})$ is a Hammer-Beresnev polynomial,

$$\delta_{ij}^{D} \ge 0 \quad \text{for } i = 1,\dots,m;\ j = 1,\dots,n. \tag{26}$$

Given a PMP instance with a cost matrix C, any solution Δ^{D} to the set of inequalities (24)–(26) will be a difference matrix for a p-equivalent instance. Given a permutation matrix $\Psi \in perm(C)$ and a difference matrix Δ^{D}, it is trivial to construct the cost matrix D of a PMP instance equivalent to a PMP instance with cost matrix C.

Remark 2 Note that we reduce the problem of finding an equivalent matrix D with the minimum number of columns to the given matrix C to the following well known Dilworth's decomposition theorem (see e.g. Theorem 14.2 in Schrijver 2003):

> "The set of terms T_a with positive coefficients in a pseudo-Boolean polynomial are subsets of partially ordered set T, and hence, the minimum number of chains covering T_a (nothing else as the minimum number of aggregated columns of C) is equal to the maximum size of an antichain (the maximum number of non-embedded terms)."

Example 4 The maximum size of antichain found for instance in Example 1 is three, and the corresponding Hammer-Beresnev polynomial $\mathcal{B}_{C,p=2}(\mathbf{y})$, aggregated matrix D and one of its permutation matrix Π_D are $\mathcal{B}_{C,p=2}(\mathbf{y}) = [33 + 7y_1 + 4y_1y_4] + [0 + 2y_2 + 2y_1y_2] + [0 + 2y_4 + 8y_3y_4]$, with matrices

$$D = \begin{bmatrix} 33 & 2 & 10 \\ 44 & 0 & 10 \\ 44 & 4 & 2 \\ 40 & 4 & 0 \end{bmatrix} \quad \text{and} \quad \Pi_D = \begin{bmatrix} 1 & 2 & 4 \\ 4 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 4 & 2 \end{bmatrix}. \tag{27}$$

6 Summary and directions for future research

Conventionally, the number of entries in the cost matrix corresponding to a PMP instance indicates the size of the representation of the instance.It is often possible to represent the instance in a more compact manner. In Sect. 3 in this paper, we presented a representation of the PMP through a pseudo-Boolean polynomial called the truncated Hammer-Beresnev polynomial which achieves this compactification. This compactification is mainly achieved through the reduction of similar monomials in the polynomial, and the truncation of the polynomial to degree $m - p$ for a p median problem with m candidate facilities. Computations presented in the section show that the compactification is significant for some benchmark problem instances.

In Sect. 4 we presented a preprocessing procedure for p-median problems based on its truncated Hammer-Beresnev polynomial representation. This representation allows us to perform a p-truncation operation on the cost matrix representing the problem instance, which in turn allows us to ignore certain plants in our search for an optimal solution to a given PMP instance. Additionally, it allows us to explain why certain p-median problem instances are more difficult to solve than others.

In Sect. 5 we showed how to construct PMP instances that have the same optimal solutions as a given PMP instance. We call these instances p-equivalent. Construction of p-equivalent instances is practically useful; given a PMP instance, we can search for a p-equivalent PMP instance belonging to a known polynomially solvable class of PMP instances, solve the p-equivalent instance easily, thereby coming up with an optimal solution to the original instance. It also allows us to use data correcting algorithms (see, e.g., Goldengorin et al. 2003b to generate good quality solutions in reasonable time).

Most of the above-mentioned results and properties of the PMP derived from its pseudo-Boolean polynomial based formulation are much more difficult to discover from conventional mathematical programming formulation of the PMP (see e.g., Avella et al. 2007).

The paper leads to three interesting courses of future research on p-median problems. The first is to evaluate the use of the concept of p-equivalence described here to extend the set of polynomially solvable PMP instances. The second is to undertake a thorough computational study of the efficacy of the pseudo-Boolean formulation of the PMP. The third is to design new exact and heuristic algorithms for solving large-scale instances of PMP based on the truncated Hammer-Beresnev polynomial representation and the preprocessing scheme developed in Sect. 4.

Acknowledgements Professor Rainer Burkard, Professor Dmitrii Pasechnik, and one of reviewers have attracted our attention to the modern algebraic notation for polynomials. Professor Alex Belenky and another reviewer have suggested some improvements in the introduction of the pseudo-Boolean formulation of PMP. Both reviewers have provided us with many helpful comments. We are very grateful to all of the above mentioned colleagues.

References

AlBdaiwi BF, Goldengorin B, Sierksma G (2009) Equivalent instances of the simple plant location problem. Comput Math Appl 57:812–820

Avella P, Sforya A (1999) Logical reduction tests for the p-median problem. Ann Oper Res 86:105–115

Avella P, Sassano A, Vasil'ev I (2007) Computational study of large-scale p-median problems. Math Program Ser A 109:89–114

Beresnev VL (1973) On a problem of mathematical standardization theory. Upr Sist 11:43–54 (in Russian)

Boros E, Hammer PL (2002) Pseudo-Boolean optimization. Discrete Appl Math 123:155–225

Briant O, Naddef D (2004) The optimal diversity management problem. Oper Res 52:515–526

Burkard RE, Kliny B, Rudolf R (1996) Perspective of Monge properties. Discrete Appl Math 70:95–161

Burkard RE, Krarup J (1998) A linear algorithm for the pos/neg-weighted 1-median problem on a cactus. Computing 60:193–215

Goldengorin B, Ghosh D, Sierksma G (2003a) Branch and peg algorithms for the simple plant location problem. Comput Oper Res 30:967–981

Goldengorin B, Tijssen GA, Ghosh D, Sierksma G (2003b) Solving the simple plant location problems using a data correcting approach. J Glob Optim 25:377–406

Hakimi SL (1964) Optimum locations of switching centers and the absolute centers and medians of a graph. Oper Res 12:450–459

Hakimi SL (1965) Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Oper Res 13:462–475

Hammer PL (1968) Plant location—a pseudo-Boolean approach. Isr J Technol 6:330–332

Kariv O, Hakimi L (1979) An algorithmic approach to network location problems, part II: The p-medians. SIAM J Appl Math 37:539–560

Khumawala BM (1972) An efficient branch-and-bound algorithm for the warehouse location problem. Manag Sci 18:718–731

Krarup J, Vajda S (1997) On Torricelli's geometrical solution to a problem of Fermat. IMA J Math Appl Bus Ind 8:215–224

Mirkin B (2005) Clustering for data mining: a data recovery approach. Chapman & Hall/CRC computer science. Chapman & Hall/CRC, London

Mladenovic N, Brimberg J, Hansen P, Moreno-Peréy JA (2007) The p-median problem: a survey of metaheuristic approaches. Eur J Oper Res 179:927–939

Reese J (2006) Solution methods for the p-median problem: an annotated bibliography. Networks 48:125–142

Revelle CS, Eiselt HA, Daskin MS (2008) A bibliography for some fundamental problem categories in discrete location science. Eur J Oper Res 184:817–848

Schrijver A (2003) Combinatorial optimization. Polyhedra and efficiency. Springer, Berlin

Weber A (1909) Über den Standort der Industrien, Erster Teil: Reine Theorie des Standortes. Mohr, Tübingen