Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 179 (2007) 1097–1108

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

The directional *p*-median problem: Definition, complexity, and algorithms

Laura E. Jackson *, George N. Rouskas, Matthias F.M. Stallmann

North Carolina State University, Department of Computer Science, Raleigh NC 27695 – 7534, United States

Received 15 December 2003; accepted 15 June 2005
Available online 7 July 2006

Abstract

An instance of a *p*-median problem gives *n* demand points. The objective is to locate *p* supply points in order to minimize the total distance of the demand points to their nearest supply point. *p*-Median is polynomially solvable in one dimension but NP-hard in two or more dimensions, when either the Euclidean or the rectilinear distance measure is used. In this paper, we treat the *p*-median problem under a new distance measure, the directional rectilinear distance, which requires the assigned supply point for a given demand point to lie above and to the right of it. In a previous work, we showed that the directional *p*-median problem is polynomially solvable in one dimension; we give here an improved solution through reformulating the problem as a special case of the constrained shortest path problem. We have previously proven that the problem is NP-complete in two or more dimensions; we present here an efficient heuristic to solve it. Compared to the robust Teitz and Bart heuristic, our heuristic enjoys substantial speedup while sacrificing little in terms of solution quality, making it an ideal choice for real-world applications with thousands of demand points.

Keywords: Directional *p*-median problem; Traffic quantization; Vector quantization

1. Introduction

The traditional *p*-median problem asks us to find, for a given set of *n* demand points, the set of *p* supply points that minimizes the total distance of each demand point to its nearest supply point. Hassin and Tamir [6] give a O(*np*) algorithm to solve the *p*-median problem on the real line, and Megiddo and Supowit [9] prove that rectilinear and Euclidean versions of the *p*-median problem in the plane are NP-complete. The choice of distance measure impacts the complexity of the problem as well as the approach needed to find a solution.

In this paper, we explore the *p*-median problem under a new distance measure, the directional rectilinear distance. On the real line, this restriction requires that the assigned supply point for a given demand point be located to the right of it, while in the plane, the assigned supply point for a given demand point must lie above and to the right of it. In general, the rectilinear *l*-directional, *k*-dimensional *p*-median problem forces a supply point to achieve or exceed the values of the first *l* coordinates of its assigned demand points. This variant of the

* Corresponding author.
E-mail address: laurajackson@nc.rr.com (L.E. Jackson).

0377-2217/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2005.06.080

p-median problem arises naturally in certain quantization applications, which we discuss shortly.

In a previous work [7] we showed that the directional p-median problem is polynomially solvable in one dimension; we give here an improved solution through reformulating the problem as a special case of the constrained shortest path problem. We have previously proven that the problem is NP-complete in two or more dimensions [8]; we present here an efficient heuristic to solve it. Compared to the robust Teitz and Bart heuristic, our heuristic enjoys substantial speedup while sacrificing little in terms of solution quality, making it an ideal choice for our target applications which may have thousands of demand points.

The rest of the paper is organized as follows. Section 2 summarizes previous results, defines the directional rectilinear distance metric, and describes applications of directional p-median. Section 3 considers the one-dimensional problem, directional p-median on the real line, and gives the constrained shortest path formulation. Section 4 presents a fast heuristic algorithm for solving the directional p-median in multiple dimensions, and Section 5 concludes the paper.

2. Problem definition

2.1. The traditional p-median problem

In k-dimensional space, $k \geqslant 1$, the *continuous* p-median problem allows supply points to be located anywhere in k-space. The *discrete* problem provides a list of *candidate points* from which supply points may be chosen. In two-dimensional space, let $d((x_i,y_i),(z_j,t_j))$ be the distance from point (x_i,y_i) to point $(z_j,0_j)$ according to some distance metric. The decision version of the continuous p-median problem in the plane may be formally stated as:

Problem 2.1 (*Continuous-PM2*). Given a set $X = \{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ of demand points in the plane, an integer p, and a bound B, does there exist a set

$S = \{(z_1,t_1),(z_2,t_2),\ldots,(z_p,t_p)\}$

of p supply points such that

$$\sum_{i=1}^{n} \min_{1\leqslant j\leqslant p}\{d((x_i,y_i),(z_j,t_j))\} \leqslant B?$$

Continuous-PM2 is NP-complete under either the Euclidean (d_e) or the rectilinear (d_r) distance measure [9], where d_e and d_r are defined as:

$$d_e((x_i,y_i),(z_j,t_j)) = \sqrt{(x_i - z_j)^2 + (y_i - t_j)^2}$$
$$d_r((x_i,y_i),(z_j,t_j)) = |x_i - z_j| + |y_i - t_j|$$

Under the rectilinear distance measure, it is well known that only demand points and *intersection points* need be considered as candidates for supply points. Intersection points are found by crossing the set $\{x_1,x_2,\ldots,x_n\}$ with the set $\{y_1,y_2,\ldots,y_n\}$, and subtracting the demand points, yielding at most $n^2 - n$ new points. Thus the continuous p-median problem under the rectilinear distance measure reduces to a discrete p-median problem. For a complete treatment of discrete location problems, the reader is referred to [4].

The discrete p-median problem in the plane can be formulated as the following integer program.

Problem 2.2 (*Discrete-PM2*)

$$\begin{aligned}
\text{Minimize}\quad & \sum_{i\in X}\sum_{j\in C} d_{ij} r_{ij} \\
\text{s.t.}\quad & \sum_{j\in C} r_{ij} = 1 \quad \forall i\in X,\ r_{ij} \leqslant s_j \quad \forall i\in X,\ j\in C, \\
& \sum_{j\in C} s_j = p,\ r_{ij}, s_j \in \{0,1\} \quad \forall i\in X,\ j\in C,
\end{aligned}$$

where $i \in X$ is the the set of demand points, $j \in C$ is the the set of candidate points, d_{ij} is the distance from point, i to point j, p is the number of supply points to be chosen,

$$r_{ij} = \begin{cases} 1 & \text{if point } i \text{ is assigned to candidate } j, \\ 0 & \text{otherwise,} \end{cases}$$
$$s_j = \begin{cases} 1 & \text{if candidate } j \text{ is chosen,} \\ 0 & \text{otherwise.} \end{cases}$$

The distance matrix $[d_{ij}]$ holds distances between demand and candidate points. Section 4.1 discusses distance matrix properties that affect the difficulty of finding a good quality solution [13].

2.2. The directional p-median problem

We now define the *directional* rectilinear distance measure. In general, an l-directional, k-dimensional rectilinear metric (with $l \leqslant k$) defines distance from point $(r_1,\ldots,r_k)$ to $(q_1,\ldots,q_k)$ to be ∞ if $r_i > q_i$ for at least one $i \in \{1,\ldots,l\}$ and $\sum_{1\leqslant i\leqslant k}|q_i - r_i|$ otherwise. Thus, in a directional p-median problem, a supply point must achieve or exceed the values of the first l coordinates of all its demand points. On the real line, this restriction requires that the nearest supply point for a given demand point be located to the right of it, while in the plane, the nearest supply point for a given demand point must lie above and to the right of it. In the plane, the 2-directional rectilinear distance is:

$$d_{\mathrm{dr}}((x_i,y_i),(x_j,y_j)) = \begin{cases} x_j - x_i + y_j - y_i & \text{if } x_j \geqslant x_i \text{ and } y_j \geqslant y_i, \\ \infty, & \text{otherwise.} \end{cases}$$

We define *directional intersection points* to be the subset of intersection points that lie above (at least) one demand point as well as to the right of (at least) one demand point, as illustrated in Fig. 1. Specifically, the point (x_j, y_j) is a directional intersection point if:

1. $(x_j, y_j) \notin X$, that is, (x_j, y_j) is not itself a demand point, and
2. there exist points $(x_i, y_i) \in X$ and $(x_k, y_k) \in X$ for which $x_j = x_i$ and $y_j = y_k$ and $x_j > x_k$ and $y_j > y_i$.

Analogous to the case of the (non-directional) rectilinear p-median problem, the continuous 2-directional rectilinear p-median problem also reduces to a discrete problem; specifically, we need only consider demand points and directional intersection points as candidates for supply points. Thus the number of candidate points c is at most $n + (n^2 - n)/2$.

2.3. Applications

The directional p-median problem arises naturally in problem domains where it is important to quantize a set of inputs taking values from a continuous set of values under the constraint that each input is mapped to a quantization level with an equal or higher value. We now consider two such applications.

Preemptive scheduling of periodic tasks on multiprocessor systems. In the slotted time model we consider, a periodic task ("demand point" in p-median terminology) is characterized by a rate x_i, $0 < x_i < 1$. Writing x_i as a fraction in lowest terms, $x_i = \frac{C_i}{D_i}$, then C_i is the computation time and D_i is the period; that is, C_i is the number of unit-length subtasks that must be processed every D_i time slots, starting at time 0. A processor may work on at most one task at a time, and a task may be processed by no more than one processor at a time. A schedule is feasible if each task i receives C_i units of processing every D_i time slots, starting at time 0. This model is nearly identical to the one considered in [3], which does not require the ratio $\frac{C_i}{D_i}$ to be in lowest terms.

A feasible schedule for this problem exists if and only if $\sum_{i=1}^{n} x_i \leqslant m$ [3], where m is the number of processors, and the fastest scheduling algorithm runs in time $O(m \log n)$ at each slot [2]. Therefore, this algorithm may not be appropriate for applications with a very large number n of tasks, such as a web server for a popular web site which might receive thousands of requests per minute, or a Grid serving very large task sets that are also highly dynamic in nature. For these applications it is essential to have a scheduling algorithm with a running time independent of the number of requests.

In [7,8], we proposed to quantize the set of n task rates into a small, fixed set $p \ll n$ of offered rates ("supply points"). Specifically, the system assigns a task with requested rate x_i to the next higher offered rate z_j, such that $x_i \leqslant z_j$. By doing so, each task is guaranteed to receive at least the required amount of computation time with each period, hence the system will meet the quality of service (QoS) requested by the user. For such a quantized system we devised a new scheduling algorithm [7,8] which runs in time $O(m)$ per slot, i.e., independent of the input size n; the new algorithm is also significantly simpler than the one in [2]. While quantization has the disadvantage of requiring more resources (e.g., processor time) than a continuous-rate system to accommodate a given set of tasks, the tradeoff is more than paid for by the resulting gains in speed and simplicity.

Control and management of packet-switched networks. A transmission link in a typical backbone network operates at data rates of 2.5–10 Gbps, and can carry hundreds of thousands of independent traffic flows (users). Each flow is typically characterized

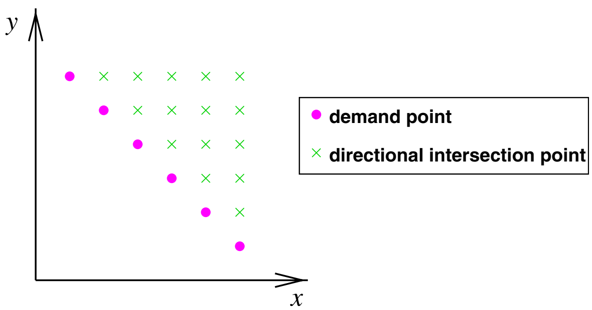

Fig. 1. A downward sloping line of demand points yields the most directional intersection points, $(n^2 - n)/2$.

by a number of traffic parameters, such as bandwidth and burst size, as well as QoS parameters, such as a delay bound. Accommodating such a large number of flows poses significant scalability problems for a number of network control and management functions. The service provider may prefer to group (quantize) similar requests into a single service level, such that any given user receives *at least* the amount requested of each resource. This quantization problem is equivalent to a directional p-median problem in multiple dimensions. The operations of traffic engineering, packet scheduling, network management, traffic policing, and billing are greatly simplified in such a quantized network. Performance analysis is also more tractable, since systems with continuous rates give rise to analytical models with infinite dimensions, and such models are usually approximated by finite-dimensional ones.

3. The directional p-median problem on the real line

In [7], we considered two variants of the one-dimensional directional p-median problem. In DPM1, the input is a finite set of demand points, while in SDPM1, the input is the probability density function representing the population of demand points. We presented optimal solutions in each case. In this work we first formally define problem DPM1 and then show how to translate DPM1 into a constrained shortest path problem, allowing a faster solution than that given in [7].

Let X be a set of n demand points on the real line $\{x_1,\ldots,x_n\}$, such that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$. A set of supply points $S = \{z_1,\ldots,z_p\}$, $z_1 < z_2 < \cdots < z_p$, $1 \leqslant p \leqslant n$, is a *feasible solution* for X if and only if $x_n \leqslant z_p$. For notational convenience, we assume $z_0 = 0$. Associated with a feasible solution is an *implied mapping* from $X \to S$, where $x_i \to z_j$ if and only if $z_{j-1} < x_i \leqslant z_j$. Fig. 2 shows a sample mapping from a set of 13 demand points onto a solution set of 6 supply points.

Letting X_j be the set of demands mapped to supply point z_j and $n_j = |X_j|$, problem DPM1 is:

Problem 3.1 (*DPM*1). Given a set X of n demand points, $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$, find a feasible set S of p supply points, $z_1 < z_2 < \cdots < z_p$, $1 \leqslant p \leqslant n$, which minimizes the following objective function:

$$g(z_1,\ldots,z_p) = \sum_{j=1}^{p} \sum_{x_i \in X_j} (z_j - x_i) \qquad (1)$$

The objective function $g(z_1,\ldots,z_p)$ is simply the sum of the distances from each demand point to its nearest (under the new directional distance measure) supply point. The minimum (optimal) value of g is called g^*, and a feasible set S at which g^* is obtained is an *optimal* solution set for X.

We have the following lemma; its proof, which can be found in [7], is straightforward.

Lemma 3.1. *Let X be a set of n demand points such that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$. There exists an optimal solution set $S = \{z_1,\ldots,z_p\}$, $z_1 < z_2 < \cdots < z_p$, of X, for which $z_j \in X$, for each $j = 1,\ldots,p$.*

3.1. Graph representation of DPM1: A faster algorithm

In [7] we present an $O(n^2p)$ algorithm that optimally solves DPM1. We can improve the solution time to $O(n\sqrt{p\log n})$ by restating DPM1 as a constrained shortest path problem (Garey and Johnson's problem ND30) (see also [15]). Let $G = (V,E)$ be a weighted, complete, directed acyclic graph (DAG), with vertex set $V = \{0,1,\ldots,n\}$ and arc weights $w(i,j)$ for arc (i,j) from vertex i to j, $0 \leqslant i < j$. Solving DPM1 is equivalent to finding a minimum weight p-link path from vertex 0 to n in G. Further, we show that the arc weights in the DPM1 graph representation obey the concave Monge property, allowing a solution in time

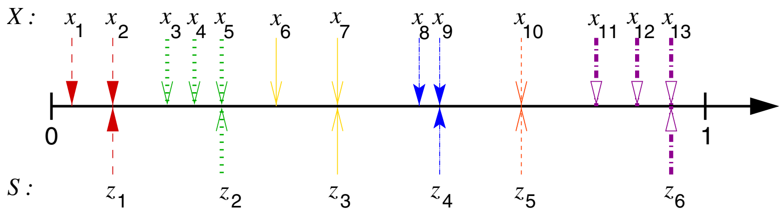

Fig. 2. Sample mapping of task densities to service levels.

$O(n\sqrt{p \log n})$ [1]. These results are stated in the following two lemmas.

Lemma 3.2. *Solving an instance of DPM1 with a set X of n demand points is equivalent to finding a minimum weight p-link path in a DAG.*

Proof. Given an instance of DPM1, we construct a DAG as follows: Demand x_i gives rise to node i, and we create a dummy node 0. Arc weight $w(i,k)$ represents the cost of mapping demand points $i+1, i+2, \ldots, k$, to point k:

$$w(i,k) = \begin{cases} 0, & k = i+1, \\ (k-i-1)x_k - \sum_{j=i+1}^{k-1} x_j, & k > i+1. \end{cases}$$

The objective is to find the minimum weight path from vertex 0 to n that has exactly p arcs. A path $t = (i_0, i_1), (i_1, i_2), \ldots, (i_{r-1}, i_r)$ is a (0-n)-path if $i_0 = 0$ and $i_r = n$. The weight of path t is:

$$w(t) = w(i_0, i_1) + w(i_1, i_2) + \cdots + w(i_{r-1}, i_r)$$

Any p-link path $t = (i_0, i_1), (i_1, i_2), \ldots, (i_{p-1}, i_p)$ with $i_0 = 0$ and $i_p = n$ is a feasible solution for DPM1, with the following interpretation: the demand points corresponding to nodes $i_1, i_2, \ldots, i_p$ are designated as supply points. □

As an example, Fig. 3 shows the graph for a DPM1 instance with $n = 5$. Suppose $p = 3$, and let (0, 2, 4, 5) be a 3-link path in Fig. 3. Then the corresponding feasible solution for DPM1 is $z_1 = x_2$, $z_2 = x_4$, and $z_3 = x_5$. The sum of the arc weights for this path equals the objective function value for the implied mapping for the corresponding solution $S = \{z_1, z_2, z_3\}$, namely:

$$\begin{aligned} w(t) &= w(0, 2) + w(2, 4) + w(4, 5) \\ &= (x_2 - x_1) + (x_4 - x_3) + 0 \end{aligned}$$

Lemma 3.3. *The arc weights for the graph representation of DPM1 obey the concave Monge condition.*

Proof. A weighted, complete DAG G satisfies the concave Monge condition if

$$w(i,j) + w(i+1, j+1) \leqslant w(i, j+1) + w(i+1, j) \tag{2}$$

holds for all $0 < i+1 < j < n$. We evaluate the left-hand side (LHS) and right-hand side (RHS) of Eq. (2), and then show that RHS − LHS ⩾ 0.

$$\begin{aligned} \text{LHS} &= w(i,j) + w(i+1, j+1) \\ &= (j-i-1)x_j - \sum_{m=i+1}^{j-1} x_m \\ &\quad + (j-i-1)x_{j+1} - \sum_{m=i+2}^{j} x_m \\ \text{RHS} &= w(i, j+1) + w(i+1, j) \\ &= (j-i)x_{j+1} - \sum_{m=i+1}^{j} x_m \\ &\quad + (j-i-2)x_j - \sum_{m=i+2}^{j-1} x_m \end{aligned}$$

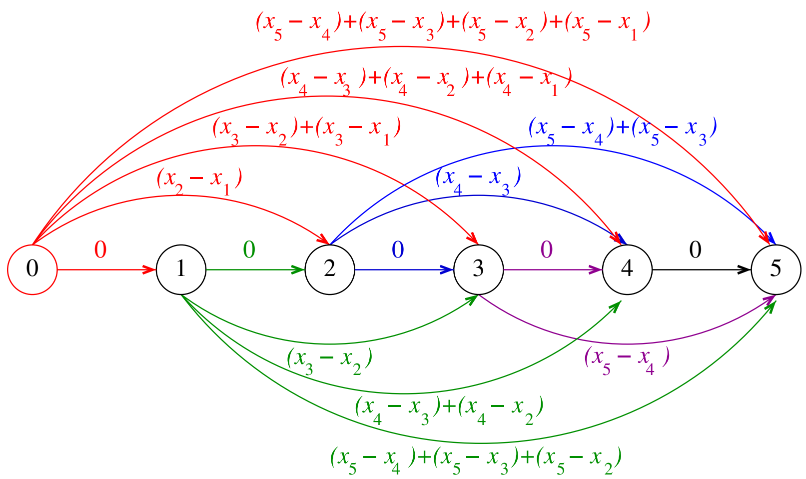

Fig. 3. Graph representation of an instance of DPM1 with $n = 5$.

$$
\begin{aligned}
&\text{RHS} - \text{LHS} \\
&\quad = (j-i-(j-i-1))x_{j+1} \\
&\qquad + (j-i-2-(j-i-1))x_j \\
&\qquad + \left(\sum_{m=i+1}^{j-1} x_m - \sum_{m=i+1}^{j} x_m + \sum_{m=i+2}^{j} x_m - \sum_{m=i+2}^{j-1} x_m\right) \\
&\quad = x_{j+1} - x_j + \left(-x_j + x_j\right) \\
&\quad = x_{j+1} - x_j \\
&\quad \geqslant 0
\end{aligned}
$$

The last step above follows from $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$. □

Due to Lemmas 3.2 and 3.3, DPM1 can be solved in time $O(n\sqrt{p \log n})$ using the algorithm in [1].

4. The directional *p*-median problem in two or more dimensions

Formally, we define the directional *p*-median problem in two dimensions (DPM2) to be:

Problem 4.1 (*DPM*2). Given a set $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ of points in the plane, an integer p, and a bound B, does there exist a set

$S = \{(z_1, t_1), (z_2, t_2), \ldots, (z_p, t_p)\}$

of p points such that

$$g(S) = \sum_{i=1}^{n} \min_{1 \leqslant j \leqslant p} \{d_{\text{dr}}((x_i, y_i), (z_j, t_j))\} \leqslant B? \tag{3}$$

In [8], we prove that DPM2 is NP-complete; we omit the proof due to its length and complexity. It follows that the directional rectilinear *p*-median problem in three or more dimensions is also NP-complete.

We now present an efficient heuristic algorithm for DPM2 that produces good quality results for a variety of input distributions. The new heuristic, which we refer to as the TB restricted (TBr) heuristic, uses as building blocks the vertex substitution heuristic of Teitz and Bart (TB) [14], described in Section 4.2, and the Heuristic Concentration approach (HC) from Rosing and ReVelle [12], described in Section 4.3. Compared to the robust TB, our heuristic enjoys substantial speedup while sacrificing little in terms of solution quality, making it an ideal choice for our target applications that have thousands of demand points. In Section 4.5, we describe a simulation study that evaluates our new heuristic under a variety of input conditions. TBr may also be easily extended to the directional rectilinear *p*-median problem in 3 or more dimensions.

4.1. Effect of distance characteristics on computational effort

Recall that the $n \times c$ distance matrix $[d_{ij}]$ holds the distances from demand points to candidate points (c is the number of candidate points). Different distance metrics produce distance matrices with different characteristics that influence the performance of a heuristic. Two such characteristics are symmetry and the ability to satisfy the triangle inequality [13]. The lack of symmetry has a minor impact on performance, but failure to satisfy the triangle inequality is a much graver crime, making optimal or even good quality solutions hard to find. (Non-directional) rectilinear and Euclidean distance matrices both are symmetric and obey the triangle inequality. A randomly generated distance matrix in general will neither be symmetric nor obey the triangle inequality. As for our directional rectilinear distance metric, the outlook is positive as regards the more critical attribute: the distance matrix obeys the triangle inequality but fails to be symmetric (recall that if $d_{dr}((x_i, y_i), (x_j, y_j))$ is finite and >0, then $d_{dr}((x_j, y_j), (x_i, y_i))$ is infinite).

ReVelle labels certain Discrete-PM2 problems as *integer friendly*, meaning that "either integer termination of linear programming formulations are frequent, or little branch and bound is needed to resolve the problem in integers" [10]. The conclusions of [13] are that the characteristic of obeying the triangle inequality has greatest impact on the integer friendliness of an instance of Discrete-PM2. The question of why this is the case remains open.

4.2. Teitz and Bart vertex substitution heuristic for p-median

The Teitz and Bart [14] vertex substitution heuristic (TB) for Discrete-PM2 is well-known and much studied. A study comparing TB to exact methods showed that TB rarely becomes trapped in local minima [11]. Although created for the non-directional *p*-median problem, TB works similarly for the directional problem.

TB begins with an initial solution of p supply points which are numbered arbitrarily from 1 to p. Assigning each demand point to its nearest supply point, the heuristic evaluates the objective function for this solution. The heuristic next replaces the first supply point with the candidate point (not already in the solution) that causes the greatest decrease in the objective function. The heuristic then repeats this process with each of the remaining supply points in turn. At each step, the heuristic seeks the best candidate to replace the supply point being considered for removal, given that all other supply points in the solution are fixed. The first major iteration ends when the heuristic has tried removing each of the p solution points, and the final solution becomes the initial solution for the second major iteration. TB terminates when a major iteration results in no changes to the solution, usually within only a few ($\leqslant 5$) iterations.

Each major iteration of TB runs in time $O(np(c-p))$, where c is the number of candidate points in an instance of DPM2. There are p existing supply points and $c-p$ alternate candidate points, for a total of $p(c-p)$ pairs to be considered; for each pair we need to examine each of the n demand points to find its closest supply point. Recall from Section 2.2 that the c candidate points in DPM2 include all n demand points as well as all directional intersection points, i.e., $c=(n^2+n)/2$ possible candidate points in the worst case. Therefore, the worst-case running time of each major iteration of TB is $O(n^3p)$.

The performance of TB depends on the starting solution. A common approach is to generate a number of random starting solutions as input for multiple TB runs, and then to choose the best solution from among the local optima that are found.

4.3. Heuristic concentration

Rosing and ReVelle present a two-stage metaheuristic called Heuristic Concentration (HC) [12]. Although they demonstrate how HC works by applying it to Discrete-PM2, HC can be applied to many different combinatorial problems. HC attempts to glean information from the many local minima obtained from repeated runs of a heuristic. For example, the (local minima) solutions resulting from separate TB runs may have differing objective function values, as well as different supply points in the solution set. HC takes advantage of the fact that there is frequently a great deal of overlap in the solution sets corresponding to these local minima. First, HC builds a concentration set (CS) by taking the union of the several local minima solutions. The CS has a high likelihood of containing the supply points that make up the optimal solution set. The second stage locates the best solution from among the members of the CS; if the size of the CS is sufficiently small, an integer linear program can be used to optimally select the best solution from the CS.

4.4. A new heuristic for DPM2

The motivation for designing a new algorithm for DPM2 is to find a very fast heuristic that does not sacrifice much in solution quality yet can tackle efficiently large instances of the problem where the number n of demand points is on the order of thousands or even hundreds of thousands. We refer to our new algorithm as the TB restricted (TBr) heuristic. TBr follows the two-stage approach of heuristic concentration. In the first stage, TBr builds the concentration set by running the one-dimensional algorithm for DPM1 (from Section 3.1) twice, once on the x-values and once on the y-values of the n demand points, to obtain the p best xs and the p best ys (when each dimension is considered *independently* of the other). Crossing these two sets yields p^2 points that form the CS. In the second stage, TBr randomly generates a number m of initial solutions from among all possible candidate points. Each initial solution serves as input into a separate run of the TB heuristic[1], which only chooses points for exchange from the CS. The final TBr solution is the best outcome from the m TB runs. Note that the final TBr solution may include a point that is *not* in the CS: each initial solution is drawn from *all* candidate points, and then TB looks only in the CS for possible replacement points.

The first stage of TBr builds the CS in time $O(n\sqrt{p\log n})$, the time required to solve DPM1. Each major iteration of TBr then runs in time $O(p(p^2-p)n)$ or $O(np^3)$. In contrast, each major iteration of TB runs in time $O(p(c-p)n)$ or $O(n^3p)$. Thus the overall complexity of TBr represents a substantial improvement over TB, especially for applications in which $n \gg p$.

[1] We have chosen to use TB at the second stage rather than the Densham and Rushton Global/Regional Interchange Algorithm (GRIA) [5]. On the traditional (non-directional) p-median problem, GRIA improves the runtime of TB. However, GRIA is not appropriate for a directional p-median problem.

4.5. Evaluation of the TBr heuristic

Simulation set-up and input parameters. To investigate the performance of TBr, we designed a simulation study using a variety of demand sets X. To generate points in the plane, we chose one discrete probability density function (pdf) for the x-values and one for the y-values, from among the possible pdf's defined in Table 1: EquallyLikely (E), Bimodal (B), and Quadrimodal (Q). We selected these density functions as representatives from the larger set of functions we considered in the comprehensive experimental study presented in [8].

An input combination is denoted by a two-letter combination – EE, EB, BB, or QQ – in which the first (respectively, second) letter represents the pdf used for the x's (respectively, y's). Fig. 4 shows scatter plots of demand sets of size $n = 1000$ generated from the four input combinations. From each input combination, we generated fifty demand sets with $n = 100$ and another fifty with $n = 200$. Each demand set was generated starting from a unique seed for a Lehmer random number generator with

Table 1
Probability density functions for input distributions

Distribution	$f(x)$	Domain
EquallyLikely	1/1000	[1, 1000]
Bimodal	1/4000 16/4000	[1, 250], [351, 650], [751, 1000] [251, 350], [651, 750]
Quadrimodal	5/24,000	[1, 95], [106, 145], [156, 445], [456, 595], [606, 1000]
	480/24,000	[96, 105], [146, 155], [446, 455], [596, 605]

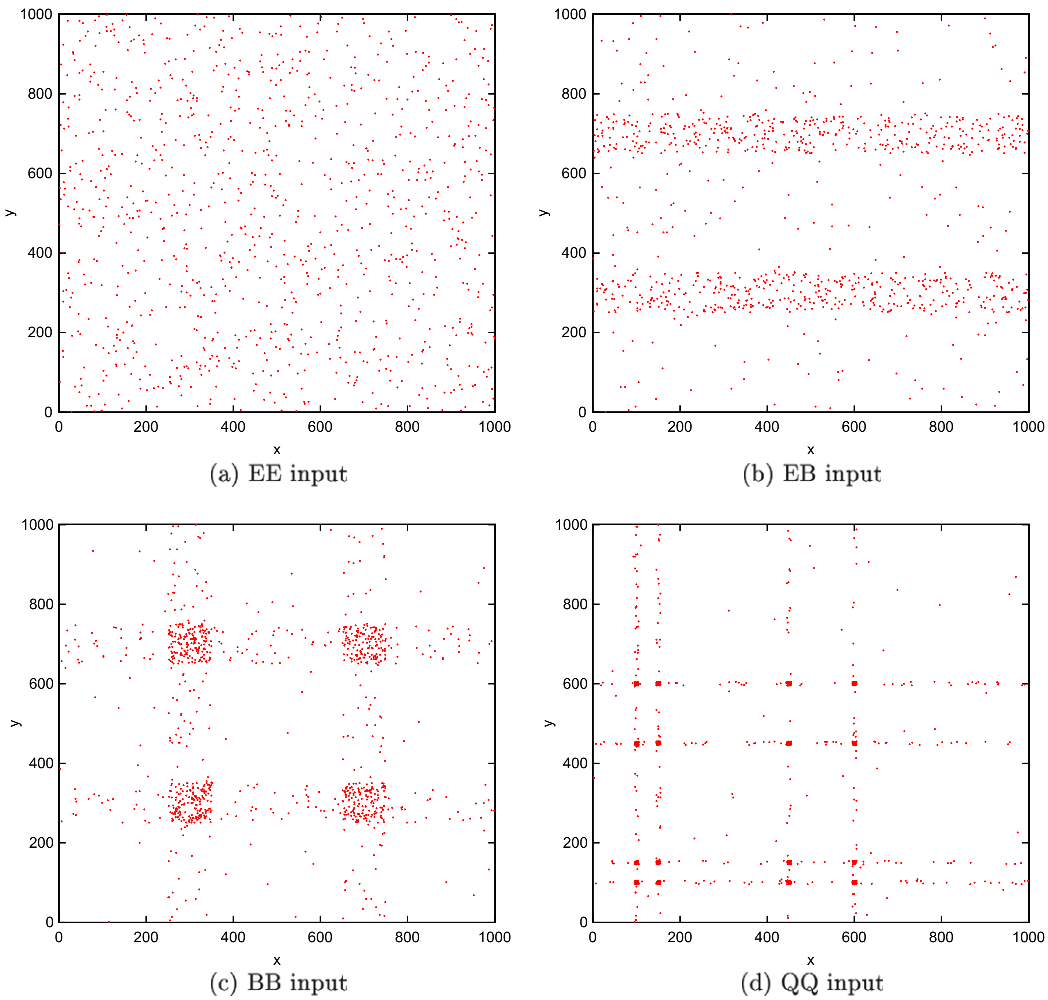

Fig. 4. Scatter plots of $n = 1000$ for the four input distribution combinations.

modulus $2^{31}-1$ and multiplier 48,271. Each demand set then served as input to the TB and TBr heuristics. The TB result for each demand set is the best of 100 runs of the TB heuristic.

We define the *normalized effectiveness* as a measure of the quality of a heuristic algorithm for DPM2:

$$\text{normalized effectiveness} = 1 + \frac{g(S)}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i} \geqslant 1 \qquad (4)$$

where $g(S)$ is the objective function from expression (3) evaluated at the set S of supply points returned by the algorithm. This definition is motivated by

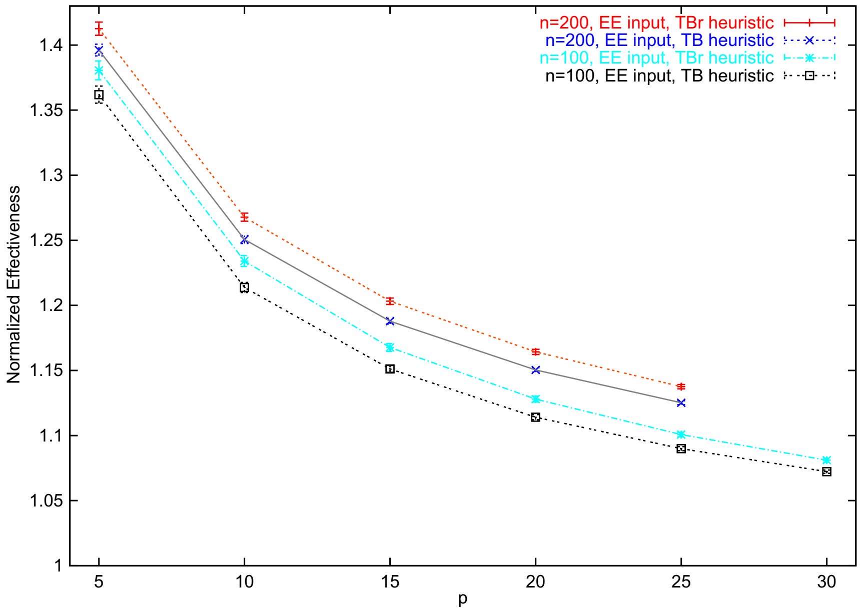

Fig. 5. Normalized effectiveness vs. p, TB and TBr heuristics, instances of size $n = 100$ and $n = 200$, EE input.

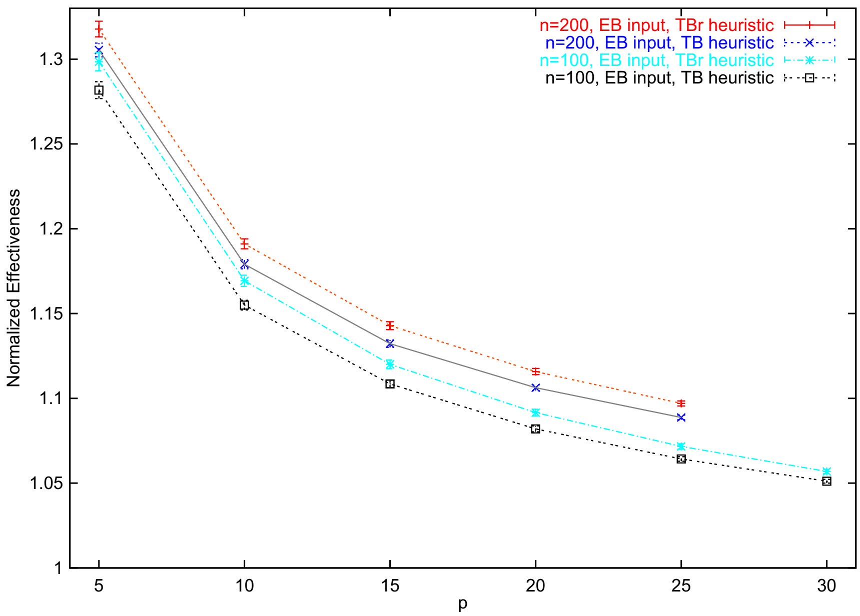

Fig. 6. Normalized effectiveness vs. p, TB and TBr heuristics, instances of size $n = 100$ and $n = 200$, EB input.

the observation that if $p = n$, the set of supply points is the same as the set of demand points; in this case, the objective function $g(S) = 0$, and the normalized effectiveness is equal to 1. Whenever $p < n$, we have $g(S) > 0$ (assuming that the n demand points are distinct), and the normalized effectiveness is greater than 1. Therefore, one is a legitimate lower bound on the normalized effectiveness. Since we cannot compute the optimal value, we can characterize the relative quality of the solutions produced by two different heuristics by observing which of the normalized effectiveness values is closer to one. In

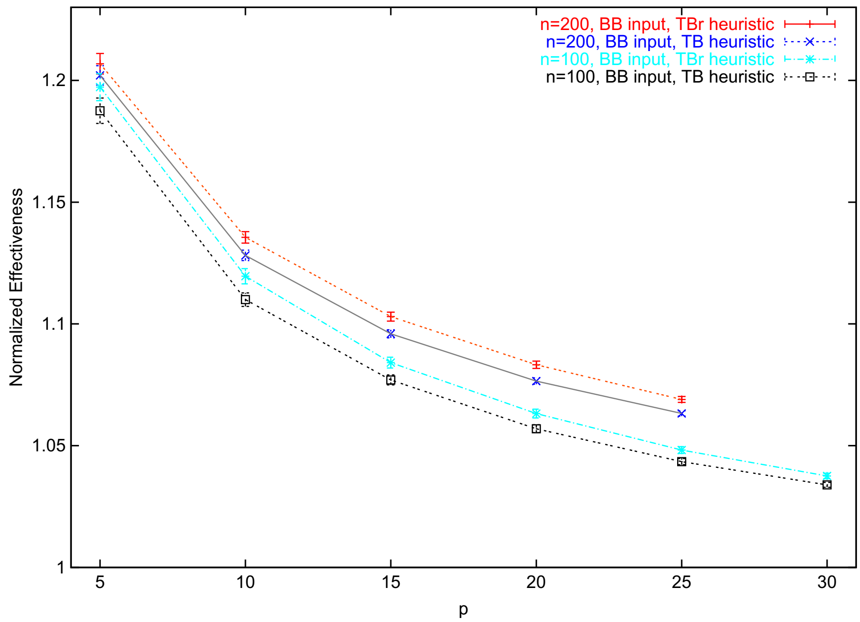

Fig. 7. Normalized effectiveness vs. p, TB and TBr heuristics, instances of size $n = 100$ and $n = 200$, BB input.

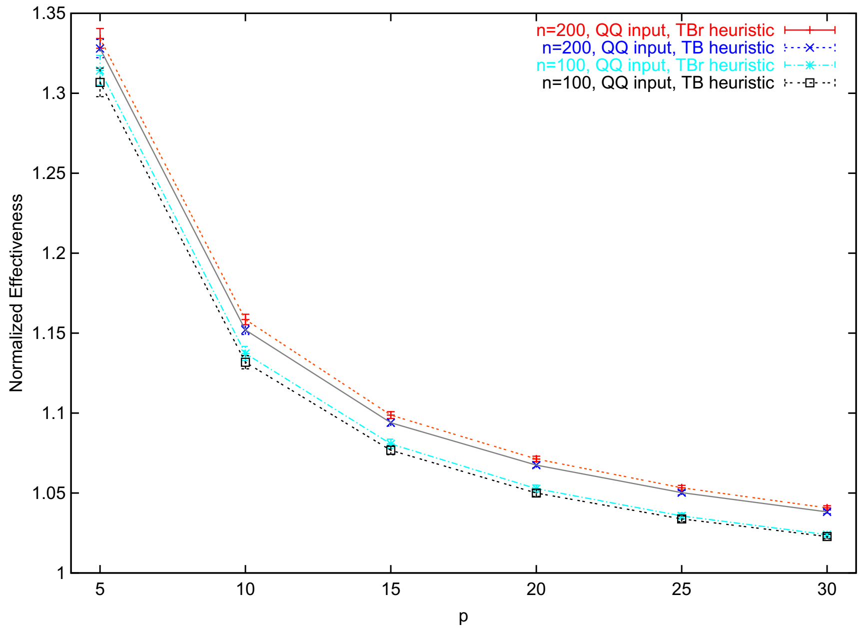

Fig. 8. Normalized effectiveness vs. p, TB and TBr heuristics, instances of size $n = 100$ and $n = 200$, QQ input.

other words, a better algorithm produces solutions with a lower value of the normalized effectiveness. We also note that the normalization allows us to compare results among problem instances with very different sets of demand points. For problem instances sets of size $n = 100$, we calculated the normalized effectiveness for $p = 5, 10, 15, 20, 25, 30$. For problem instances of size $n = 200$, we calculated the normalized effectiveness for $p = 5, 10, 15, 20, 25$. In addition, we calculated the normalized effectiveness for $p = 30$ for QQ instances of size $n = 200$.

Simulation results. The four graphs in Figs. 5–8 correspond to the four input combinations, EE, EB, BB, and QQ. For a given input, the graph shows the normalized effectiveness plotted against p for both the TB and the TBr heuristics, for problem instances of size $n = 100$ and $n = 200$. Each point is the mean of 50 problem instances, with error bars designating a 95% confidence interval. We note that, for each input, the $n = 200$ curve lies above the $n = 100$ curve. Holding p fixed, it is reasonable to expect a higher normalized effectiveness value for a demand set of size $n = 200$ as compared to $n = 100$. We also note that the TBr curve lies slightly above the TB curve, illustrating the quality of the TBr solution. We pay little penalty in solution quality for using TBr instead of TB, yet we realize great gains in speed. Finally, we note that the relative performance of the two heuristics is not affected significantly as the number p of supply points increases; in other words, the solutions returned by TBr are close to those returned by TB across the range of p values we considered in these experiments.

From the figures we observe that, for all values of p, the curves for input EE are the highest, followed by EB, BB, and QQ. We can attribute this result to the nature of the input. Namely, in the sample scatter plots for each input shown in Fig. 4, the level of "order" increases as we move from EE to EB to BB to QQ. (Or, said another way, the level of randomness decreases as we move from EE to EB to BB to QQ.) The EE input appears to be the worst case scenario, with points scattered evenly over the plane. In contrast, the other inputs possess natural clusters where points fall with higher density: EB has two horizontal strips, BB has four squares, and QQ has 16 squares. The more natural clustering that exists in the input, the easier it is for an algorithm to select appropriate supply points.

Finally, in Fig. 9 we present results of the TBr heuristic on instances of size $n = 1000$. Since the number of candidate points is $O(n^2/2)$, then moving from an instance of size $n = 100$ to one with $n = 1000$ represents a 100-fold increase in problem size. Due to the high complexity of the TB heuristic, we were not able to obtain results for $n > 200$ within a reasonable amount of time, (e.g., a few hours).

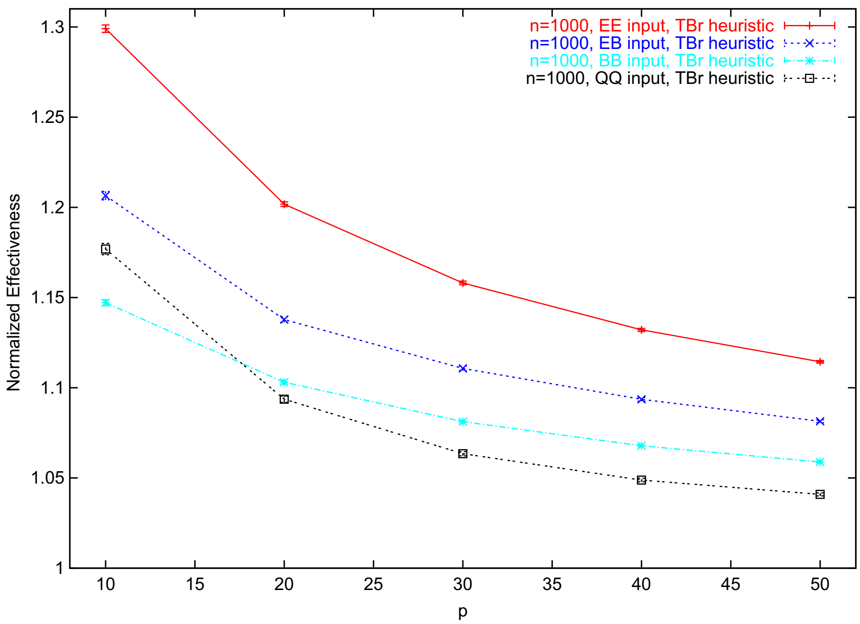

Fig. 9. Normalized effectiveness vs. p, TBr heuristic, instances of size $n = 1000$.

Each point in the graphs is the mean of 25 problem instances, and for each instance, TBr takes the best solution of 10 runs; error bars designate a 95% confidence interval. We observe that the curves corresponding to the four input combinations (EE, ED, BB, and QQ) exhibit the same relative behavior we described above. We also note that the curve for a particular input combination lies just above the corresponding curve for $n = 200$ in Figs. 5–8, indicating that the incremental penalty as the problem size increases is relatively small.

5. Concluding remarks

We have explored the p-median problem under a new distance measure: the rectilinear l-directional, k-dimensional p-median problem forces a supply point to achieve or exceed the values of the first l coordinates of its assigned demand points. We have shown that the one-dimensional directional p-median problem can be solved in time $O(n\sqrt{p \log n})$ through a constrained shortest path reformulation. For the NP-complete 2-dimensional problem, we presented a new heuristic that builds upon the Teitz and Bart (TB) heuristic and the Heuristic Concentration metaheuristic from Rosing and ReVelle. Our heuristic is a faster alternative for applications with very large demand sets. Although TB consistently returns solutions of slightly better quality, our heuristic tracks TB's performance closely and enjoys a significant speedup, running in time $O(np^3)$ as compared to TB's time of $O(n^3p)$.

References

[1] A. Aggarwal, B. Schieber, T. Tokuyama, Finding a minimum weight k-link path in graphs with Monge property and applications, Discrete Computational Geometry 12 (1994) 263–280.

[2] J.H. Anderson, A. Srinivasan, Mixed Pfair/ERfair scheduling of asynchronous periodic tasks, Journal of Computer and System Sciences 68 (1) (2004) 157–204.

[3] S.K. Baruah, N.K. Cohen, C.G. Plaxton, D.A. Varvel, Proportionate fairness: a notion of fairness in resource allocation, Algorithmica 15 (6) (1996) 600–625.

[4] M. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, John Wiley and Sons, New York, 1995.

[5] P.J. Densham, G. Rushton, A more efficient heuristic for solving large p-median problems, Papers in Regional Science: The Journal of the RSAI 71 (3) (1992) 307–329.

[6] R. Hassin, A. Tamir, Improved complexity bounds for location problems on the real line, Operations Research Letters 10 (1991) 395–402.

[7] L. Jackson, G.N. Rouskas, Optimal quantization of periodic task requests on multiple identical processors, IEEE Transactions on Parallel and Distributed Systems 14 (7) (2003) 795–806.

[8] Laura E. Jackson, The Directional p-Median problem with applications to traffic quantization and multiprocessor scheduling. PhD thesis, North Carolina State University, Raleigh, NC, December 2003.

[9] N. Megiddo, K.J. Supowit, On the complexity of some common geometric location problems, SIAM Journal on Computing 13 (1) (1984) 182–196.

[10] C.S. ReVelle, Facility siting and integer-friendly programming, European Journal of Operational Research 65 (2) (1993) 47–158.

[11] K.E. Rosing, E.L. Hillsman, H. Rosing-Vogelaar, A note comparing optimal and heuristic solutions to the p-median problem, Geographical Analysis 11 (1979) 86–89.

[12] K.E. Rosing, C.S. ReVelle, Heuristic concentration: Two stage solution construction, European Journal of Operational Research 97 (1) (1997) 75–86.

[13] D.A. Schilling, K.E. Rosing, C.S. ReVelle, Network distance characteristics that affect computational effort in p-median location problems, European Journal of Operational Research 127 (3) (2000) 525–536.

[14] M.B. Teitz, P. Bart, Heuristic methods for estimating the generalized vertex median of a weighted graph, Operations Research 16 (1968) 955–961.

[15] H.C. Joksch, The shortest route problem with constraints, Journal of Mathematical Analysis and Applications 14 (1966) 191–197.