2012 3rd International Conference on Information Security and Artificial Intelligence (ISAI 2012)
IPCSIT vol. 56 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V56.33

An Optimal Algorithm for Weighted Rectilinear Facility Location Problem

Xiaodong Wang[1 +], Lei Wang[2], and Yingjie Wu[3]

[1]College of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362000, PR China

[2]College of Computing, Georgia Institute of Technology, Atlanta GA 30332,USA

[3]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350002, PR China

Abstract. In this paper, we consider the weighted rectilinear min-sum facility problem to minimize the sum of weighted rectilinear distance between the given points and a new added point. We present a simple optimal algorithm for the weighted rectilinear min-sum facility location problem using linear time and space. The new algorithm is the first optimal algorithm for the weighted rectilinear min-sum facility problem and the techniques used in this paper are interesting themselves.

Keywords: facility location; binary search; algorithms

1. Introduction

Facility location problems have been widely studied, last but not least due to their importance in practical applications. Facility location problems typically involve a finite set of sites at which facilities can be located, and a finite set of clients, which demand requests to be supplied from facilities.

In many industrial applications concerning capacity planning, different commodities are manufactured and delivered directly from the factories, at discounted bulk delivery rates, or via warehouses or distribution centers to satisfy the customer demands.

A simple facility location problem is the Fermat-Weber problem, in which a single facility is to be placed, with the only optimization criterion being the minimization of the sum of distances from a given set of point sites.

This generalizes the median, which has the property of minimizing the sum of distances for one-dimensional data, and provides a central tendency in higher dimensions. It is also known as 1-median.

More complex problems considered in this discipline include the placement of multiple facilities, constraints on the locations of facilities, and more complex optimization criteria.

In this paper, we consider the following weighted rectilinear min-sum facility location problem:

Given a set $S = \{p_0, p_1, \ldots, p_{n-1}\}$ of n points in R^d with positive weights $w_0, w_1, \ldots, w_{n-1}$, find a point $y^* = (y_0^*, y_1^*, \ldots, y_{d-1}^*) \in R^d$ which minimizes the sum of weighted rectilinear distance (with l_1 norm) between the points in S and y^*.

A possible application for this problem is finding a location for a service center in a town in which all the streets are parallel to the axes. The location of the service center should minimize the sum of the weighted rectilinear distance between n customer locations given in S, and the service center.

When deciding where to place a facility that serves geographically scattered client sites - whether the facility is a delivery center, a distribution center, a transportation hub, a fleet dispatch location, etc - a typical

+ Corresponding author.
E-mail address: wangxiaodong@qztc.edu.cn

objective is to minimize the sum of the distances from the facility's location to the client sites. Furthermore, we may want to give the distance to certain sites more "weight" in the calculation, for example if a site requires several deliveries daily as opposed to one delivery daily to other sites, resulting in higher fuel and other costs. Or, perhaps we want to ensure particularly good service to a certain site. In these cases, these higher-weighted client sites may have greater relative influence on the final location.

This problem arises in the area of facility location in operations research known as weighted rectilinear 1-median problem, proposed by Hakimi [8, 9], and many variants of it have been studied (see, for example,[1, 3, 4, 6, 7, 10, 12]).

The weighted rectilinear min-sum facility location problem in R^d can be formulated as

$$\min_{y\in R^d} g(y)=\sum_{i=0}^{n-1}\sum_{j=0}^{d-1} w_i \mid p_{ij}-y_j \mid \tag{1}$$

where $p_i=(p_{i,0},p_{i,1},\ldots,p_{i,d-1})\in R^d, i=0,1,\ldots,n-1$, are the n given points in the set S.

The l_1 metric is separable, in the sense that the weighted distance between two points is the sum of their weighted distances of every coordinates. Therefore we can solve the problem for their coordinates separately. We regard the first coordinate part. The problem is reduced to the one dimensional case:

$$\min_{x\in R} f(x)=\sum_{i=0}^{n-1} w_i \mid x_i - x \mid \tag{2}$$

where $x_i, i=0,1,\ldots,n-1$, are the n real numbers.

The complexity of the problem is unknown [11]. We are unaware of any previous linear time algorithms for this problem.

Organization of the paper. In the following sections we solve the weighted rectilinear min-sum facility location problem. In section 2 we give a linear time solution algorithm for the problem. Computational experiments of the presented algorithm are performed in section 3. Some concluding remarks are in section 4.

2. The Binary Search Algorithm

It is not difficult to see that the function $f(x)$ in formula (2) is a piecewise linear function. Without loss of generality, let us assume that

$$-\infty = x_{-1} < x_0,\ldots,\le x_{n-1} < x_n = +\infty \tag{3}$$

The piecewise linear function $f(x)$ can be expressed as follows.

$$f(x)=f_j(x)=k_jx+b_j,\ x\in (x_{j-1},x_j] \tag{4}$$

where

$$k_j=\sum_{i=0}^{j-1} w_i - \sum_{i=j}^{n-1} w_i,\ b_j=\sum_{i=j}^{n-1} w_ix_i - \sum_{i=0}^{j-1} w_ix_i \tag{5}$$

The values of k_j and b_j can be computed iteratively as follows.

$$k_0=-\sum_{i=0}^{n-1} w_i,\ k_{j+1}=k_j+2w_j,\ 0\le j<n \tag{6}$$

$$b_0=\sum_{i=0}^{n-1} w_ix_i,\ b_{j+1}=b_j-2w_jx_j,\ 0\le j<n \tag{7}$$

$$k_0=-k_n,\ b_0=-b_n \tag{8}$$

It is easy to verify that at the position x_j we have

$$f(x_j)=f_j(x_j)=f_{j+1}(x_j),\ 0\le j<n \tag{9}$$

It follows that function $f(x)$ has a global minimum if $k_0<0$ and function $f(x)$ has a global maximum if $k_0>0$. The function $f(x)$ has both a global minimum and a global maximum when $k_0=0$.

If function $f(x)$ has a global minimum, then

$$\min_{x\in R}\left\{\sum_{i=0}^{n-1} w_i |x_i - x|\right\} = \min_{x\in R} f(x) = \min\{f(x_0), f(x_1), \ldots, f(x_{n-1})\} \tag{10}$$

If function has a global maximum, then

$$\max_{x\in R}\left\{\sum_{i=0}^{n-1} w_i |x_i - x|\right\} = \max_{x\in R} f(x)$$

$$= \max\{f(x_0), f(x_1), \ldots, f(x_{n-1})\} \tag{11}$$

Therefore

$$\min_{x\in R} f(x) = \min_{0\le i<n}\{k_i x_i + b_i\} \tag{12}$$

$$\max_{x\in R} f(x) = \max_{0\le i<n}\{k_i x_i + b_i\} \tag{13}$$

Theorem 1 *If all the weights $w_0, w_1, \ldots, w_{n-1}$ are positive, then the function $f(x)$ in formula (2) achieves its global minimum $f(x_j) = k_j x_j + b_j$ at the position x_j, if $k_j < 0$ and $k_{j+1} \ge 0$, where $k_j, 0 \le j < n$, are computed by (6).*

Proof. Since all the weights are positive, we have $k_0 = -\sum_{i=0}^{n-1} w_i < 0$. Therefore function $f(x)$ has a global minimum. It is easy to see from the formula (6) for computing k_j that $0 > k_0 < k_1 < \cdots < k_{n-1} > 0$. The function $f(x)$ is a convex piecewise linear function. Therefore, a local minimum of $f(x)$ is also a global minimum of $f(x)$.

A global minimum occurs when the function $f(x)$ changes its sign of its slop k_j. That is, the function $f(x)$ achieves its global minimum $f(x_j) = k_j x_j + b_j$ at the position x_j, when $k_j < 0$ and $k_{j+1} \ge 0$. ■

According to the Theorem 1, we can design a binary search algorithm for finding the global minimum of the function $f(x)$ as follows.

```
Binary-Search(x, w, first, last, k)
--------------------------------------
while last − first > 1 do
    mid ← (first + last)/2
    Select(x, w, first, mid, last)
    m ← k + Σ_{i=first}^{mid−1} w_i
    if m < 0 then
        m ← k
        first ← mid
    else
        last ← mid
    end if
end while
return first
```

The inputs of the above algorithm Binary-Search are n real number $x_0, x_1, \cdots, x_{n-1}$ and the associated n positive weights $w_0, w_1, \cdots, w_{n-1}$.

The variables $first$, mid and $last$ are used to represent *begin*, *middle* and *end* positions of the search interval respectively.

The variable k is actually $k_{first}/2$ computed by formula (6) for the current search interval $[first, last]$. Similarly the variable m is actually $k_{mid}/2$ computed by formula (6) for the middle position of the current search interval $[first, last]$.

The variables $first$ and $last$ are initialized with number 0 and $n-1$ respectively. The initial value of the variable k is $k_0/2$.

The algorithm is guaranteed that the global minimum of function $f(x)$ is located in the interval $[first, last]$. In the above algorithm Binary-Search, the median of the sequence $x_{first}, x_{first+1}, \cdots, x_{last}$ is chosen by the algorithm Select and the sequence $x_{first}, x_{first+1}, \cdots, x_{last}$ is partitioned such that the median is located in the position mid.

For all indices $i, first \le i < mid, x_i \le x_{mid}$. For all indices $j, mid < j \le last, x_j \ge x_{mid}$. Then the value of $m = k_{mid}/2$ is computed in row 4 according to formula (6).

There are two cases to be distinguished.

In the case of $m < 0$, the global minimum of function $f(x)$ is located in the interval $[mid, last]$ by Theorem 1. The interval $[first, mid-1]$ is discarded and the search interval is reduced to $[mid, last]$.

In the case of $m \ge 0$, the global minimum of function $f(x)$ is located in the interval $[first, mid]$ by Theorem 1. The interval $[mid+1, last]$ is discarded and the search interval is reduced to $[first, mid]$. In either case the size of search interval is reduced to half.

Let the time complexity of the algorithm Binary-Search is $T(n)$ in the worst case. The algorithm Select can be completed in linear time in the worst case due to Blum, Floyd, Pratt, Rivest and Tarjan [2] (see alternatively [5]). Remaining computation in the while loop of the algorithm can be completed in linear time obviously. Therefore the time complexity of the algorithm $T(n)$ is determined by the following formulas:

$$T(n) = \begin{cases} T(n/2) + O(n) & , n > 2 \\ 1 & , n \le 2 \end{cases} \tag{14}$$

The solution for the above recurrence is $T(n) = O(n)$. We have proved

Theorem 2 *The weighted rectilinear min-sum facility location problem is solvable in linear time.*

3. Computational Experiments

We use the following concrete function $f(x)$ of (2) as an example to demonstrate the linear time algorithm for the weighted rectilinear min-sum facility location problem described above.

$$\min_{x \in R} f(x) = \sum_{i=1}^{n} i\,|i - x| \tag{15}$$

where $x_i = i, w_i = i, i = 1, 2, \ldots, n$.

For fixed $n = 10$, the piecewise linear function function $f(x)$ can be shown in Table 1.

For this example, the search range of the algorithm Binary-Search is $[1,10]$. The algorithm first checks the middle point $x_5 = 5$ of the search range. The left slope of the piecewise linear function $f(x)$ at this point is $k_4 = -35$. The global minimum of $f(x)$ is located in the right of the point $x_5 = 5$.

The new search range of the algorithm becomes $[5,10]$ and the length of the search range is halved. The algorithm then checks the middle point $x_7 = 7$ of the new search range. The left slope of the piecewise linear function $f(x)$ at the point is $k_6 = -13$. The global minimum of $f(x)$ is again located in the right of the checking point. The new search range of the algorithm becomes $[7,10]$.

The next checking point is $x_8 = 8$, where the left slope of $f(x)$ is $k_7 = 1$. At this point we know that the piecewise linear function $f(x)$ has a negative left slop $k_6 = -13$ and a positive right slop $k_7 = 1$ at the point $x_7 = 7$. Therefore the global minimum of $f(x)$ is achieved at the point $x_7 = 7$. The algorithm Binary-Search terminated and the location of the global minimum is returned.

We have also implemented our new algorithm for the function $f(x)$ in various sizes and compared with the randomly generated function $g(x)$. The computational experiment results are shown in Table 2. It is not difficult to observe from Table 2 that the time complexity of the presented algorithm is indeed linear.

4. Concluding Remarks

We proposed a linear time algorithm for the weighted rectilinear min-sum facility location problem in the case of all the weights positive.

Since the input size of the problem is $O(n)$, our $O(n)$ time algorithm for computing the weighted rectilinear min-sum facility location is obviously optimal.

5. Acknowledgements

The authors would like to thank the referees for their comments.

This work was supported by the Natural Science Foundation of Fujian under Grant No.2009J01295 and the Haixi Project of Fujian under Grant No.A099.

6. References

[1] Jens Bleiholder and Felix Naumann, Data Fusion, ACM Computing Surveys, 2008, 41(1), pp.1-41.

[2] M. Blum, R. Floyd, V. Pratt, R. Rivest, R. Tarjan, Time bounds for selection, J. Comput. System Sci., 1973, 7, pp.448-461.

[3] Natashia Boland, Patricia Dominguez-Marin, Stefan Nickel, Justo Puerto, Exact procedures for solving the discrete ordered median problem, Computers & Operations Research, 2006, 33, pp.3270-3300.

[4] Marek Chrobak, Claire Kenyon, Neal Young, The reverse greedy algorithm for the metric k-median problem, Information Processing Letters, 2006, 97, pp.68-72.

[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest: Introduction to Algorithms, 2nd edition, MIT Press, 1990, pp.183-192.

[6] Tammy Drezner, Zvi Drezner, Carlton H. Scott, Location of a facility minimizing nuisance to or from a planar network, Computers & Operations Research, 2009, 36, pp.135-148.

[7] Reza Zanjirani Farahani, Zvi Drezner and Nasrin Asgari, Single facility location and relocation problem with time dependent weights and discrete planning horizon, Ann Oper Res, 2009, 167, pp.353-368.

[8] S.L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph, Oper. Res., 1964, 12, pp.450-469.

[9] S.L. Hakimi, Optimum distribution of switching centers in a communication network and some related graph-theoretic problems, Oper. Res., 1965, 13, pp.462-475.

[10] Abraham P. Punnen, Y.P. Aneja, On k-sum optimization, Operations Research Letters, 1996, 18, pp.233-236.

[11] K. Rosen, Location theory, In S. Louis Hakimi editor, Handbook of Discrete and Combinatorial Mathematics, CRC Press, 2000, pp.993-994.

[12] Arie Tamir. Sorting weighted distances with applications to objective function evaluations in single facility location problems. Operations Research Letters, 2004, 32(3), pp 249-257.

Table 1: The key points of $f(x)$

number i	x_i	w_i	k_i	b_i	$f(x_i)$
1	1.00	1.00	-53.00	383.00	330.00
2	2	2	-49	375	277
3	3	3	-43	357	228
4	4	4	-35	325	185
5	5	5	-25	275	150
6	6	6	-13	203	125
7	7	7	1	105	112
8	8	8	17	-23	113
9	9	9	35	-185	130
10	10	10	55	-385	165

Table 2: Binary-Search tests: times in seconds, Genuine Intel 1.60GHz, 0.99Gb RAM

n	1000	5000	10000	50000	100000	500000	1000000	5000000	10000000
$f(x)$	0.01	0.05	0.09	0.47	0.94	4.66	9.27	46.31	92.5
$g(x)$	0.03	0.11	0.23	1.16	1.77	9.05	17.17	101.42	188.55