Comput Optim Appl (2013) 56:723–734
DOI 10.1007/s10589-013-9574-3

Solving the median problem with continuous demand on a network

Rafael Blanquero · Emilio Carrizosa

Received: 5 November 2012 / Published online: 28 June 2013

Abstract Where to locate one or several facilities on a network so as to minimize the expected users-closest facility transportation cost is a problem well studied in the OR literature under the name of median problem.

In the median problem users are usually identified with nodes of the network. In many situations, however, such assumption is unrealistic, since users should be better considered to be distributed also along the edges of the transportation network. In this paper we address the median problem with demand distributed along edges and nodes. This leads to a global-optimization problem, which can be solved to optimality by means of a branch-and-bound with DC bounds. Our computational experience shows that the problem is solved in short time even for large instances.

Keywords Network location · Median problem · Continuous demand · DC functions · Global optimization

1 Introduction

Location problems on networks have attracted the interest of researchers and practitioners since the 60s of last century. Under the usual assumption, the demand is concentrated at the nodes of the network, and the facilities can be located either at the nodes or along the edges of the transportation network.

Research supported by grants from the Spanish Ministry of Education, Culture and Sport MTM2009-14039-C06-06, Junta de Andalucía TIC-6064, FQM-329, in part financed by the European Regional Development Fund (ERDF).

R. Blanquero (✉) · E. Carrizosa
Universidad de Sevilla, Seville, Spain
e-mail: rblanquero@us.es

E. Carrizosa
e-mail: ecarrizosa@us.es

Assuming that the demand is concentrated at the nodes is not realistic when modeling so-called *networks spatial phenomena*, e.g. [18], that is, phenomena which do happen in points along edges of the network, such as, for instance, traffic accidents, or close to such edges, as happens in urban settings, where edges model the city streets, close to the buildings where demand happens. See [18, 19] for further discussion on the advantages on continuous network models against traditional approaches (discrete or planar location models).

For this reason, several researchers have addressed location problems on networks under the assumption that demand is not only concentrated on nodes, but also is (continuously) distributed along the edges of the network.

Most papers consider the case in which, for each edge of the network, demand is uniformly distributed. Nkansah and David [17] addresses the problem of minimizing the expected distance from the users to one facility, [6, 8] consider the two-facility case on very particular network topologies (trees), whereas [15] addresses the minimization of the variance of distances from the users to the facility. Assuming uniform demands on edges should be seen as a first step towards gaining realism of the model, while maintaining tractability. Indeed, the resulting objective functions admits a rather simple form, as a piecewise polynomial function in one variable, whose optimization is reduced to inspecting all critical points, namely, extreme points and points at which the derivative of the polynomial function vanishes.

Assuming more general distributions for the demand has been advocated by several authors. Okabe et al. [18] suggests the use of general density distributions, from which random samples are generated, yielding a discrete approximation to the problem, which is the one which is later analyzed. Statistical kernel methods have also been recently proposed to model the demand, [20, 23], though, as far as the authors know, no optimization has been carried out, excepting, as said above, discretization via simulation.

In this paper we consider a single-facility location problem, namely, the 1-median problem: the point minimizing the expected distance from the users is sought. With respect to the state-of-the-art, we give a further step towards realism, by assuming arbitrary distributions for the demand along edges. Contrary to the planar 1-median problem, known to be convex, [7], the 1-median problem on networks with continuous demand poses nontrivial challenges: we have no longer a simple expression for the objective, and its optimization calls for the use of global-optimization techniques. In particular, it is shown that the objective function is DC, [10, 11, 21, 22] i.e., it can be written as the difference of two convex functions, and thus its optimization can be addressed via branch-and-bound methods customized for DC functions, [1, 4, 5].

The remainder of the paper is organized as follows. In Sect. 2 the 1-median problem with demand distributed on edges and on nodes of the network is formally introduced. Properties of the objective function are discussed in Sect. 3, where it is shown, in particular, how the function can be expressed as the difference of two convex functions on each edge of the network. These properties will be the cornerstone for a branch-and-bound algorithm, as described in Sect. 4. Our numerical experience is reported in Sect. 5, showing that our algorithm enables us to solve problems on large networks in reasonable time.

2 Problem formulation

Let $N=(A,E)$ be a connected and undirected network, with node set $A=\{a_1,\dots,a_n\}$ and edge set E, with $|E|=m$. Let us denote by l_{ij} the length of each edge $e_{ij}=[a_i,a_j]\in E$ and let $d(x,y)$ be the distance between two points $x,y\in N$, obtained as the shortest path from x to y. In particular, the distance $d_{ij}=d(a_i,a_j)$ between each pair of nodes $\{a_i,a_j\}$ can be worked out by using standard algorithms, [2]. Note that for every pair of nodes a_i,a_j with $[a_i,a_j]\in E$, it follows that $d_{ij}\le l_{ij}$, and equality holds if and only if the edge $[a_i,a_j]$ is the shortest path joining a_i and a_j.

Given a node $a_k\in A$ and a point $x\in[a_i,a_j]$, obtained after covering a distance l_x on the edge $[a_i,a_j]$, the distance $d(x,a_k)$ from a_k to x is, as a function of x, a piecewise linear concave function given by:

$$d(x,a_k)=\min\{r_{ij}^k(x),s_{ij}^k(x)\} \tag{1}$$

where

$$r_{ij}^k(x)=d(a_i,a_k)+l_x \qquad s_{ij}^k(x)=d(a_j,a_k)+(l_{ij}-l_x) \tag{2}$$

We assume that the demand not only occurs at nodes but also along the edges of the network. More precisely, the demand of a node $a\in A$ will be denoted by $\omega_a\ge 0$, the total demand of a given edge $e\in E$ is $p_e\ge 0$, and it is distributed along e according to a random variable with cumulative distribution function (cdf) F_e.

Under the previous assumptions, the median problem with continuous demand can be written as follows:

$$\min_{x\in N} H(x):=\sum_{a\in A} w_a d(x,a)+\sum_{e\in E} p_e\int_{y\in e} d(x,y)dF_e(y) \tag{3}$$

Observe that we are making no assumption on the type of distribution followed for the demand. In case the demand on the edges is continuously distributed along e, i.e., when the cdf F_e has a pdf f_e, (3) can be rewritten as

$$\min_{x\in N} H(x):=\sum_{a\in A} w_a d(x,a)+\sum_{e\in E} p_e\int_{y\in e} d(x,y)f_e(y)dy. \tag{4}$$

3 Properties

It has been noticed in [12] that the objective function of (3) is neither convex nor concave as a rule. Indeed, the objective function can exhibit local optima which are not globally optimal, as the following example shows. So the use of global optimization techniques is required if one seeks the optimal solution.

Example 1 Let us consider the network $N=(A,E)$ with $A=\{a_1,a_2,a_3\}$ and $E=\{[a_1,a_2],[a_1,a_3],[a_2,a_3]\}$. Arc lengths, node demands and arc demands are the

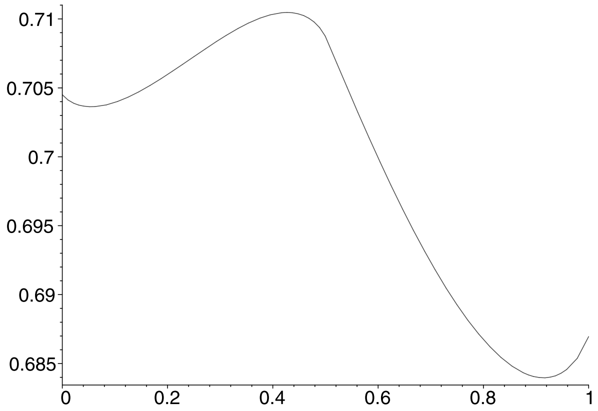

Fig. 1 Objective function $H_{12}(x)$ in Example 1

following:

$$\begin{array}{lll} l_{12}=1 & l_{13}=1 & l_{23}=1 \\ w_1=0 & w_2=0 & w_3=0 \\ p_{12}=0.35 & p_{13}=0.30 & p_{23}=0.35 \end{array}$$

We also assume that the demand along each arc e is distributed according to a beta distribution, i.e., the probability density function f_e has the form

$$f_e(x)=\frac{\Gamma(\alpha_e+\beta_e)}{\Gamma(\alpha_e)\Gamma(\beta_e)}x^{\alpha_e-1}(1-x)^{\beta_e-1} \quad x\in[0,1].$$

The parameters α_e, β_e of these probability distributions in the three edges are as follows:

Arc e	α_e	β_e
$[a_1,a_2]$	0.6	0.5
$[a_1,a_3]$	0.4	0.8
$[a_2,a_3]$	0.5	0.5

Under these assumptions, the objective function of Problem (3) restricted to the edge $[a_1,a_2]$ takes the form

$$H_{12}(x)=0.35\int_0^1 |x-y|f_{12}(y)dy+0.30\int_0^1 \min\{x+y,3-x-y\}f_{13}(y)dy$$
$$+0.35\int_0^1 \min\{2+x-y,1-x+y\}f_{23}(y)dy \tag{5}$$

Multimodality of the function is clearly seen in Fig. 1.

The special structure of the objective function of Problem (3) will be exploited in order to design a deterministic global optimization algorithm that allows us to find an optimal solution to the problem. More precisely, we will show that $H(x)$ in (3) belongs to the broad class of DC functions, [10, 11, 21]. This key property will allow us to solve Problem (3) by branch-and-bound algorithms, as the one described in Sect. 4, since lower and upper bounds can easily be obtained for DC functions as soon as a DC decomposition is available.

Definition 2 Let $\Omega \subset \mathbb{R}^n$ be a convex set. A function $h : \Omega \to \mathbb{R}$ is called DC in Ω if there exist two convex functions $h^+ : \Omega \to \mathbb{R}$, $h^- : \Omega \to \mathbb{R}$ such that

$$h(x) = h^+(x) - h^-(x) \quad \forall x \in \Omega \tag{6}$$

A pair (h^+, h^-) satisfying (6) is called a DC decomposition of h in Ω.

An interesting property of the class of DC functions is that it is closed under the most common operations in optimization, [3, 10, 11, 21, 22]. In particular, if $h_1, \dots, h_r$ are DC functions and $\lambda_i \in \mathbb{R}$, $i = 1, \dots, r$, then $\sum_{i=1}^r \lambda_i h_i$, $\max_{i=1,\dots,r} h_i$, $\min_{i=1,\dots,r} h_i$ and $\|(h_1, \dots, h_r)\|$ are also DC and their DC decompositions can be easily obtained from the DC decompositions of h_i, $i = 1, \dots, r$.

The following result shows that the objective function $H(x)$ in (3) is DC on each arc of the network.

Proposition 3 *Given $\bar{e} \in E$, the function $H_{\bar{e}} : \bar{e} \mapsto \mathbb{R}$ defined as*

$$H_{\bar{e}}(x) := \sum_{a \in A} w_a d(x, a) + \sum_{e \in E} p_e \int_{y \in e} d(x, y) dF_e(y)$$

is DC. A DC decomposition of $H_{\bar{e}}$ on $\bar{e}$ is given by the pair $(H_{\bar{e}}^+, H_{\bar{e}}^-)$, with

$$\begin{aligned} H_{\bar{e}}^+(x) &= p_{\bar{e}} \int_{y \in \bar{e}} |x - y| dF_{\bar{e}}(y) \\ H_{\bar{e}}^-(x) &= H_{\bar{e}}^+(x) - H_{\bar{e}}(x) \end{aligned} \tag{7}$$

Proof Let us rewrite $H_{\bar{e}}(x)$ as $H_{\bar{e}}(x) = h_1(x) + h_2(x) + h_3(x)$ where

$$h_1(x) = \sum_{a \in A} w_a d(x, a) \tag{8}$$

$$h_2(x) = \sum_{e \in E, e \neq \bar{e}} p_e \int_{y \in e} d(x, y) dF_e(y) \tag{9}$$

$$h_3(x) = p_{\bar{e}} \int_{y \in \bar{e}} d(x, y) dF_{\bar{e}}(y) \tag{10}$$

The function h_1 is piecewise linear and concave (see [13] for instance), and h_2 is also concave, see [12]. In what follows it is shown that h_3 is DC, and a DC decomposition is given. Given $x, y \in \bar{e} = [a_i, a_j]$, the distance $d(x, y)$ between x and y is obtained as the minimum of the lengths of the following paths:

1. the subedge of $\bar{e}$ with x, y as end points,

2. the subedge of $\bar{e}$ joining x and a_i, the shortest path joining a_i and a_j, and then the subedge of $\bar{e}$ joining a_j and y,
3. the subedge of $\bar{e}$ joining x and a_j, the shortest path joining a_j and a_i, and then the subedge of $\bar{e}$ joining a_i and y.

In other words $d(x, y)$ can be expressed as

$$d(x, y) = \min\{|x - y|, x + d_{ij} + (l_{ij} - y), (l_{ij} - x) + d_{ij} + y\} \tag{11}$$

$$= \min\{|x - y|, l_{ij} + d_{ij} - |x - y|\} \tag{12}$$

$$= |x - y| - \max\{0, 2|x - y| - (l_{ij} + d_{ij})\} \tag{13}$$

Observe that the last expression gives a DC decomposition of d on $\bar{e}$. Hence, h_3 can be written as

$$h_3(x) = p_{\bar{e}} \int_{y \in \bar{e}} |x - y| dF_{\bar{e}}(y) - p_{\bar{e}} \int_{y \in \bar{e}} \max\{0, 2|x - y| - (l_{ij} + d_{ij})\} dF_{\bar{e}}(y), \tag{14}$$

which yields a DC decomposition of h_3. Taking into account that $H_{\bar{e}} = h_1 + h_2 + h_3$, with h_1, h_2 concave and h_3 decomposed as a difference of convex functions in (14), it follows that (7) gives a DC decomposition of $H_{\bar{e}}$, as asserted. □

We end this section with further properties of the objective function H_e under some assumption on the edge e. These results extend previous well-known results for particular topologies, e.g. for networks which are chains or trees.

Corollary 4 *Let $\bar{e} \in E$ be an edge such that $p_{\bar{e}} = 0$. Then $H_{\bar{e}}$ is concave on $\bar{e}$.*

Proof If $p_{\bar{e}} = 0$, then, by Proposition 3, $(0, -H_{\bar{e}})$ is a valid DC decomposition for $H_{\bar{e}}$. □

Proposition 5 *Let $\bar{e} \in E$ be an edge such that $E \setminus \{\bar{e}\}$ is a disconnected network. Then $H_{\bar{e}}$ is convex on $\bar{e}$.*

4 The algorithm

Problem (3) will be solved by using a standard branch and bound method [14, 16] which will find out the optimal solution within a relative accuracy of $\varepsilon > 0$. The bounds of the objective function required to applying the algorithm will be worked out by taking into account its DC structure. A brief description of such an algorithm is shown next.

- Phase 1: Initialization

  1. Fix the required accuracy $\varepsilon > 0$.
  2. Set $UB = +\infty$ (upper bound initialization).
  3. Compute the all-pairs distance matrix.
  4. Set the list Λ of remaining segments as empty.

5. For each edge $e \in E$ do:
   (a) Consider e as a segment with its nodes as the segment vertices.
   (b) Evaluate the objective function at the segment midpoint. If this value is lower than UB, then update UB and store in x_{UB} the midpoint as incumbent.
   (c) Calculate a lower bound for the segment e, $LB(e)$.
   (d) If $LB(e) < UB/(1+\varepsilon)$, then insert e into Λ.

- Phase 2: Branch and Bound process
  Repeat as long as no stop was reached:

1. Select from Λ the minimum lower bound segment e_{min} and remove if from Λ.
2. If $LB(e_{min}) \geq UB/(1+\varepsilon)$, then stop the algorithm with x_{UB} as optimal solution and UB as optimal objective value.
3. Split e_{min} by its midpoint into two smaller segments, e^1_{min} and e^2_{min}.
4. Evaluate the objective function at the midpoint of the two small segments. If any of these values is lower than UB, then update UB.
5. Compute a lower bound of the objective function on each small segment.
6. If $LB(e^i_{min}) < UB/(1+\varepsilon)$ for $i = 1$ or $i = 2$, then insert e^i_{min} into Λ.
7. If UB has been updated in this iteration, then discard all segments from Λ whose lower bound is greater than UB.

The algorithm uses a data structure Λ where all the segments (bits of edge) that can contain an optimal solution are stored. The loop in Phase 1 establishes the initial composition of the data structure by selecting the edges whose lower bound is not greater that the global upper bound of the optimal objective value. At the same time, that upper bound is improved by evaluating each edge at its middle point and replacing the bound with the objective value when this is smaller.

Phase 2 of the algorithm consists of an undefined loop where the segment with the worst lower bound is processed; the algorithm finishes when the difference between that lower bound and the global upper bound is smaller than the tolerance ε chosen in Phase 1. If the stopping rule is not fulfilled, the selected segment is split into two equal segments which are processed in the same way that the edges in Phase 1. Eventually, if a change in the upper bound took place during an iteration, all the segments in the data structure that cannot contain an optimal solution are removed.

The computation of the objective function's lower bound on each segment requires more attention and is going to be detailed next.

4.1 Constructing lower bounds

Given an edge $\bar{e} = [a_i, a_j] \in E$, Proposition 3 provides a DC decomposition for $H_{\bar{e}}$ –the restriction of the objective H to e– and this fact can be exploited in order to obtain the lower bounds required in the previous algorithm. Starting from the DC representation (7), a concave underestimate $L_{\bar{e}}(x)$ of $H_{\bar{e}}$ is obtained by replacing $H^+_{\bar{e}}$ with an affine underestimate built in the usual way,

$$L_{\bar{e}}(x) = H^+_{\bar{e}}(x_0) + \xi(x - x_0) - H^-_{\bar{e}}(x)$$

where ξ is any point in $\partial H_{\bar{e}}^{+}(x_0)$, the subdifferential of $H_{\bar{e}}^{+}$ at $x_0 \in \bar{e}$. Since

$$\partial H_{\bar{e}}^{+}(x_0) = \int_{(0,x_0)} dF_{\bar{e}}(y) + [-1, 1] \int_{\{x_0\}} dF_{\bar{e}}(y) - \int_{(x_0,l_{\bar{e}})} dF_{\bar{e}}(y),$$

one has

$$2F_{\bar{e}}(x_0) - 1 \in \partial H_{\bar{e}}^{+}(x_0),$$

and thus, a concave underestimate $L_{\bar{e}}(x)$ is given by

$$L_{\bar{e}}(x) = H_{\bar{e}}^{+}(x_0) + \big(2F_{\bar{e}}(x_0) - 1\big)(x - x_0) - H_{\bar{e}}^{-}(x)$$

Due to the concavity of $L_{\bar{e}}$, it is enough to evaluate this function at the extreme points of $\bar{e}$ to obtain its minimum on the segment, which is also a lower bound for $H_{\bar{e}}^{+}$. Hence,

$$LB(\bar{e}) = \min\big\{L_{\bar{e}}(a_i), L_{\bar{e}}(a_j)\big\}$$

5 Computational results

The effectiveness of the proposed algorithm was investigated with the aid of numerical cases.

The algorithm described in Sect. 4 was coded in Fortran and compiled using Intel©Fortran Compiler XE 12.0. Executions were carried out on an Intel Core i7 computer with 8.00 Gb of RAM memory at 2.8 GHz, running Windows 7. The solutions were found to a relative accuracy of 10^{-3} and the integrals were calculated by means of the functions qdags and qdagp available at the IMSL Fortran Numerical Library.

We experimented with a set of 43 test networks obtained from [9, 24]. The number of nodes of these test problems ranges from 150 to 1000, and the number of edges from 296 to 3083. Each problem was solved 10 times over each network using randomly generated parameters: the demands of nodes were obtained from a Uniform distribution on [0, 1], as it is also the case of the overall demand of each edge. Regarding the demand along each edge, it was assumed to be distributed following a Beta distribution with parameters randomly generated on the interval [0.1, 5], which provides a wide range of density functions with very different shapes.

After the resolution of each set of 10 instances, some statistical measures (minimum, maximum, average and standard deviation) were calculated for the following indicators of the algorithm performance:

- Number of iterations of the Phase 2 of the algorithm.
- Maximum size of the data structure used for storage reached during the algorithm execution.
- CPU time.

Table 1 shows the computational results, where the number of nodes $|A|$ and edges $|E|$ of the graphs are reported as well as the above-mentioned computational measures.

Table 1 Computational results

Network	$\|A\|$	$\|E\|$	Iterations			B&B list			Time		
			min	max	mean ± std	min	max	mean ± std	min	max	mean ± std
KROB200G	200	386	6	18	11.60 ± 4.03	11	18	13.60 ± 2.46	27.89	32.76	30.14 ± 1.63
KROB150G	150	296	13	21	17.20 ± 2.70	4	4	4.00 ± 0.00	17.61	20.89	19.19 ± 1.17
KROA200G	200	392	11	17	13.70 ± 1.89	8	13	9.60 ± 1.58	30.58	34.60	32.15 ± 1.37
UR137	980	1744	10	16	12.20 ± 2.04	11	11	11.00 ± 0.00	494.23	543.32	511.20 ± 13.80
KROA150G	150	297	11	22	16.90 ± 4.09	6	16	13.20 ± 3.16	17.46	20.01	18.55 ± 0.80
PR152G	152	296	2	7	4.00 ± 1.56	5	6	5.10 ± 0.32	14.77	18.47	16.34 ± 1.03
RAT195G	195	336	4	11	8.10 ± 2.18	3	6	3.30 ± 0.95	17.89	20.31	18.67 ± 0.89
TS225G	225	306	9	15	12.10 ± 1.73	9	11	9.40 ± 0.70	8.94	10.30	9.85 ± 0.46
UR532	298	597	4	44	12.00 ± 13.26	3	28	10.10 ± 7.78	69.69	82.91	73.39 ± 4.14
UR542	343	862	9	17	13.90 ± 2.77	8	18	11.90 ± 3.48	169.28	182.12	175.00 ± 3.98
UR552	388	1135	15	30	22.90 ± 4.72	14	14	14.00 ± 0.00	333.14	347.49	339.86 ± 5.33
UR562	416	1403	23	37	28.20 ± 3.97	27	30	28.60 ± 0.84	532.68	569.67	551.77 ± 12.75
UR732	452	915	1	15	7.90 ± 5.00	6	11	8.00 ± 2.21	164.72	179.26	171.37 ± 4.72
UR535	458	812	0	4	1.60 ± 1.78	3	3	3.00 ± 0.00	102.79	121.29	113.57 ± 5.78
UR545	476	1104	7	20	14.00 ± 3.80	15	24	19.60 ± 2.67	264.00	285.97	275.21 ± 9.58
UR555	490	1305	8	22	16.20 ± 4.59	25	30	26.90 ± 1.79	401.20	429.67	415.26 ± 10.39
UR537	493	868	8	14	11.00 ± 2.05	10	13	12.10 ± 1.45	118.01	134.64	124.53 ± 5.49
UR565	496	1513	15	27	21.90 ± 3.60	23	28	26.30 ± 1.64	573.02	631.32	604.81 ± 19.32
UR547	498	1112	11	19	15.00 ± 2.62	9	14	11.40 ± 1.35	259.46	278.79	269.54 ± 6.59
UR557	498	1310	9	21	16.30 ± 4.16	10	17	11.50 ± 2.42	404.76	425.90	413.57 ± 7.60
UR567	499	1426	13	20	16.10 ± 2.02	5	9	6.80 ± 1.99	510.09	540.01	519.53 ± 8.27
UR742	538	1325	9	20	15.10 ± 4.09	15	18	16.90 ± 0.74	393.67	425.37	406.78 ± 10.22
UR752	580	1735	30	54	43.20 ± 6.68	25	42	29.50 ± 5.60	787.07	825.64	799.86 ± 12.27

Table 1 (*Continued*)

Network	$\|A\|$	$\|E\|$	Iterations			B&B list			Time		
			min	max	mean ± std	min	max	mean ± std	min	max	mean ± std
UR762	593	2089	31	47	39.40 ± 5.13	21	22	21.80 ± 0.42	1179.76	1271.95	1229.23 ± 26.58
UR132	605	1122	12	27	21.40 ± 3.92	8	8	8.00 ± 0.00	224.33	239.21	233.02 ± 5.58
UR735	662	1200	10	15	12.00 ± 1.56	8	11	10.50 ± 1.08	234.36	255.16	247.08 ± 6.57
UR142	709	1815	5	12	8.40 ± 2.32	4	4	4.00 ± 0.00	748.35	805.64	779.01 ± 17.74
UR745	713	1616	6	16	10.80 ± 3.74	13	14	13.60 ± 0.52	536.82	591.37	561.00 ± 15.58
UR755	724	1966	12	21	15.70 ± 3.27	19	23	20.50 ± 1.84	926.85	972.39	945.26 ± 13.39
UR765	741	2278	6	18	11.90 ± 3.93	13	18	14.20 ± 1.40	1316.04	1366.79	1341.44 ± 16.73
UR737	744	1315	6	14	10.90 ± 2.42	4	4	4.00 ± 0.00	273.45	320.83	290.30 ± 12.75
UR747	745	1659	9	20	14.10 ± 3.31	11	13	11.90 ± 0.57	574.32	609.25	586.98 ± 12.07
UR757	748	1969	12	23	16.70 ± 3.30	19	21	19.90 ± 0.74	891.87	967.30	929.94 ± 21.33
UR767	749	2314	14	22	17.00 ± 2.58	24	53	36.60 ± 9.81	1369.49	1427.39	1390.21 ± 17.92
UR152	766	2390	17	33	24.70 ± 4.74	16	18	17.10 ± 0.88	1455.80	1564.55	1491.42 ± 30.88
UR162	802	2897	30	47	37.80 ± 5.33	23	34	31.00 ± 2.98	2321.47	2419.84	2366.32 ± 32.81
UR135	892	1619	0	7	3.40 ± 2.50	4	6	5.50 ± 0.85	430.73	465.54	447.52 ± 11.17
UR145	929	2117	12	25	20.40 ± 4.03	16	36	28.90 ± 5.72	953.42	1011.92	975.20 ± 18.78
UR155	975	2680	8	22	14.90 ± 3.78	10	10	10.00 ± 0.00	1701.16	1816.37	1755.81 ± 32.00
UR165	980	3068	12	18	15.10 ± 1.97	20	29	24.10 ± 2.47	2368.22	2453.96	2425.75 ± 27.64
UR147	996	2254	10	21	14.60 ± 3.20	5	9	6.90 ± 1.66	1037.24	1113.22	1081.61 ± 23.26
UR157	1000	2690	11	26	18.80 ± 5.35	34	38	36.70 ± 1.42	1671.54	1779.63	1726.31 ± 30.29
UR167	1000	3083	14	25	18.20 ± 3.26	17	34	25.70 ± 7.15	2408.45	2495.42	2451.24 ± 31.00

The number of iterations and the maximum size of the branch-and-bound list remain low in all the executions. However, the CPU time results are quite high mainly due to the computation of the integrals in Phase 1 (steps 5-b and 5-c) and Phase 2 (steps 4 and 5). One can see that the CPU time required to solve different instances of the same problem shows a great stability.

We summarize the findings of this paper. We have addressed the problem of locating one facility on a network with demand distributed on nodes and edges, following arbitrary distributions. The problem is shown to be multimodal, calling for the use of global optimization techniques. The objective function on each edge has been shown to be DC, and a DC decomposition is given. This enables us to obtain concave underestimates of the objective, which are used in a branch and bound procedure. Our numerical tests show that problems of large size are solved rather quickly. Extensions of our techniques to the multifacility case are now under study.

References

1. Bello, L., Blanquero, R., Carrizosa, E.: On minimax-regret Huff location models. Comput. Oper. Res. **38**, 90–97 (2011)
2. Bertsekas, D.P.: Network Optimization: Continuous and Discrete Models. Athenas Scientific, Belmont (1998)
3. Blanquero, R., Carrizosa, E.: Optimization of the norm of a vector-valued DC function and applications. J. Optim. Theory Appl. **107**, 245–260 (2000)
4. Blanquero, R., Carrizosa, E.: Continuous location problems and big triangle small triangle: constructing better bounds. J. Glob. Optim. **45**, 389–402 (2009)
5. Blanquero, R., Carrizosa, E., Hansen, P.: Locating objects in the plane using global optimization techniques. Math. Oper. Res. **34**, 837–858 (2009)
6. Brandeau, M.L., Chiu, S.S., Batta, R.: Locating the two-median of a tree network with continuous link demands. Ann. Oper. Res. **6**, 223–253 (1986)
7. Carrizosa, E., Conde, E., Muñoz-Márquez, M., Puerto, J.: The generalized Weber problem with expected distances. RAIRO. Rech. Opér. **29**, 35–57 (1995)
8. Cavalier, T.M., Sherali, H.D.: Network location problem with continuous link demands: p-medians on a chain and 2-medians on a tree. Eur. J. Oper. Res. **23**, 246–255 (1986)
9. Corberán, A., Sanchis, J.M.: A branch & cut algorithm for the windy general routing problem and special cases. Networks **49**, 245–257 (2007)
10. Horst, R., Thoai, N.V.: DC programming: overview. J. Optim. Theory Appl. **103**, 1–43 (1999)
11. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer, Berlin (1996)
12. Chiu, S.S.: The minisum location problem on an undirected network with continuous link demands. Comput. Oper. Res. **14**, 369–383 (1987)
13. Larson, R.C., Odini, A.R.: Urban Operations Research. Prentice Hall, Englewood Cliffs (1981)
14. Lawler, E.L., Wood, D.E.: Branch-and-bound methods: a survey. Oper. Res. **14**, 699–719 (1966)
15. López de los Mozos, M.C., Mesa, J.A.: The variance location problem on a network with continuously distributed demand. RAIRO. Rech. Opér. **34**, 154–181 (2000)
16. Mitten, L.G.: Branch and bound methods: general formulation and properties. Oper. Res. **18**, 24–34 (1970)
17. Nkansah, P.T., David, H.T.: Network median problems with continuously distributed demand. Transp. Sci. **20**, 213–219 (1986)
18. Okabe, A., Okunuki, K., Shiode, S.: SANET: a toolbox for spatial analysis on a network. Geogr. Anal. **38**, 57–66 (2006)
19. Okabe, A., Satoh, T., Furuta, T., Suzuki, A., Okano, K.: Generalized network Voronoi diagrams: concepts, computational methods, and applications. Int. J. Geogr. Inf. Sci. **22**, 965–994 (2008)
20. Okabe, A., Satoh, T., Sugihara, K.: A kernel density estimation method for networks, its computational method and a GIS-based tool. Int. J. Geogr. Inf. Sci. **23**, 7–32 (2009)
21. Tuy, H.: DC optimization: theory, methods and algorithms. In: Handbook of Global Optimization, Kluwer Academic, Dordrecht (1995)

22. Tuy, H., Al-Khayyal, F., Zhou, F.: A d.c. optimization method for single facility location problems. J. Glob. Optim. **7**, 209–227 (1995)
23. Xie, Z., Yan, J.: Kernel density estimation of traffic accidents in a network space. Comput. Environ. Urban Syst. **32**, 396–406 (2008)
24. Data instances for arc routing problems. Available at www.uv.es/corberan/instancias.htm