Knowledge-Based Systems 191 (2020) 105226

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Accelerated hyperbolic smoothing method for solving the multisource Fermat–Weber and k-Median problems[☆]

Vinicius Layter Xavier [a,*], Adilson Elias Xavier [b]

[a] Department of Statistics, Institute of Mathematics and Statistics, Rio de Janeiro State University, Brazil
[b] Department of Computer Sciences, Federal University of Juiz de Fora, Brazil

ARTICLE INFO

Article history:
Received 17 May 2019
Received in revised form 3 October 2019
Accepted 11 November 2019
Available online 19 November 2019

Keywords:
Multisource Fermat–Weber
Min-sum-distances clustering
k-Median
Facility location
Smoothing

ABSTRACT

This article deals with the Multisource Fermat–Weber and continuous *k*-Median problems. The first problem is the continuous location–allocation problem, defined in a planar region, an important problem in facility location subject. The continuous *k*-Median problem, defined in a multidimensional space, is also known as the minimum sum-of-distances clustering problem. Their mathematical modellings lead to a min-sum-min formulation which is a global optimization problem with a bi-level nature, nondifferentiable and with many minimizers. To overcome these severe difficulties, the Hyperbolic Smoothing methodology is proposed, in connection with a partition of locations in two groups: location in the frontier and location in gravitational regions, which drastically simplify the computational tasks. For the purpose of illustrating both the reliability and the efficiency of the method, we perform a set of computational experiments making use of the traditional instances described in the literature. Apart from consistently presenting similar or even better results when compared to related approaches, the novel technique was able to deal with instances never tackled before, with up to 1243088 cities.

1. Introduction

The Multisource Fermat–Weber problem has different names, such as the continuous location–allocation problem, or the Fermat–Weber problem, as discussed by Wesolowsky [1]. It is an important nondifferentiable and a nonconvex mathematical problem of type $min - sum - min$, with a large number of local minimizers, as presented by Rubinov [2]. So, it is a global optimization problem.

Although this paper considers a particular Multisource problem, it must be emphasized that the proposed methodology, Hyperbolic Smoothing, can be used in exactly the same way for solving different $min - sum - min$ problems, e.g., clustering problems [3,4]. The continuous location–allocation problem originated in the 17th century, when Fermat posed the question of, given three fixed points in a planar region, find a new point such that the sum of the distances from each fixed point to the new point is minimized. Alfred Weber, a century ago, introduced the same problem for a general number of points, also adding weights on each point to consider customer demand.

The Multisource Fermat–Weber problem locates an arbitrary number of facilities (medians) at *continuous* locations in the Euclidean plane, as presented by Koopmans and Beckmann [5]. Its formulation produces a mathematical problem of global optimization, which is both a non-differentiable and a nonconvex problem, with a large number of local minimizers. There is an expressive number of articles studying this problem, perhaps the most frequently studied in location sciences. The bibliography presents many strategies for solving it, as shown by the extensive survey by Brimberg et al. [6], where an interested reader can obtain an ample overview of the subject.

In a short survey of the field, a few works can be highlighted. Cooper [7] proposed an efficient heuristic to solve the problem based on a scheme of iterative switching of location and allocation steps until no more gains in the values of the object function are obtained. This simple scheme is used by many researchers to this day, and it is as well the inspiration for several variant procedures. Bongartz et al. [8] present a projective method which treats, in a simultaneous way, the intrinsic location and allocation aspects of the problem. However, methods based in mathematical programming techniques have had little success due to the combinatorial nature of the problem. For this reason, some of more successful methods for solving this problem are based on metaheuristics. We are concerned in focusing the references on

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105226.

* Corresponding author.
E-mail addresses: viniciuslx@ime.uerj.br (V.L. Xavier), adilson.xavier@gmail.com (A.E. Xavier).

https://doi.org/10.1016/j.knosys.2019.105226
0950-7051/

an important group of researchers, who have been treating successfully the problem for more than 20 years by using different metaheuristic alternatives both for getting starting points of good quality and for solving the problem [6,9–12]. Notwithstanding this great interest, the bibliography only presents computational results for solving instances with moderate sizes, in general, datasets with up to 1060 cities.

A related location problem is the discrete p-median problem, where the location of facilities is restricted to a given finite set of possible places. The *discrete* problem has also received the most relevant interest from the literature [13–16], which also presents different alternatives for solving it.

The core focus of this paper adopts a novel approach: the smoothing of the $min - sum - min$ problem engendered by the modelling of the Multisource Fermat–Weber problem. The process whereby this is achieved is an extension of a smoothing scheme, called Hyperbolic Smoothing (HS), applied with broad success for solving large nondifferentiable problems in general, as e.g. for the $min - -max$ [17]; for the covering of plane domains by circles [18]; for covering of solid bodies by spheres [19]; or for the continuous location of hubs [20]. A overview of successful applications of the HS approach for solving a select set of five important problems can be seen in [21]. The HS technique was developed through an adaptation of the hyperbolic penalty method originally introduced by Xavier [22,23].

The present paper describes also a procedure for simplifying the computational tasks. The basic idea is the partition of the set of locations into two non-overlapping parts. By using a conceptual presentation, the first set corresponds to the location of points relatively close to two or more centroids. This set of locations, named boundary band points, can be managed by using the previously presented smoothing approach. The second set corresponds to location points significantly closer to a single centroid, in comparison with others. This set of locations, named gravitational points, is managed in a direct and simple way, offering much faster performance.

The articulation of hyperbolic smoothing with the above related partitioning strategy creates a new and wonderful scenario. First, the mathematical programming problem produced by the hyperbolic smoothing approach presents a completely differentiable characteristic, thus its solution can be obtained by using the most powerful optimization methods based on first and second order derivatives. On the other hand, the proposed strategy of partitioning enables a super expressive simplification of the computational work, since it reduces the initial set of locations to an effective set that is only a small proportion of the original one. For problems of great dimensions, such reduction of the computational work can be superior to 99,9% of the total of locations. Moreover, the larger the problem, the higher the reduction factor.

As will be shown below, the novel proposed algorithm delivers computational results that are superior to other alternatives as per the following criteria: accuracy (the lowest value of the object function); speed (shorter processing time); consistency (less variability among the values of solutions obtained from different starting points); and scalability (capability of solving problems of huge dimensions), which permits finding good solutions to problems that could not be solved by the methods presently in use, up to 1243088 cities.

For the sake of completeness, we present first the Hyperbolic Smoothing Fermat–Weber Method (HSFWM) [24]. A step-by-step definition of the problem and the original hyperbolic smoothing approach are presented in Section 2. Next, we address the new Accelerated Hyperbolic Smoothing Fermat–Weber Method (AHSFWM), a fast version of the original method. The boundary and gravitational regions partition scheme and the derived algorithm AHSFWM are presented in Section 3. Computational results are reported in Section 4. Conclusions are drawn in Section 5.

2. The multisource Weber problem formulation

Preliminarily, Table 1 describes the mathematical notation for specifying the Multisource Weber problem formulation.

Let $S = \{s_1, \dots, s_m\}$ denote a set of m cities or locations in an Euclidean planar space $\Re^2$, with a corresponding set of demands $W = \{w_1, \dots, w_m\}$, to be attended by a given number q of facilities. To formulate the traditional Multisource Weber problem as a $min-sum-min$ problem, we proceed as follows. Let $x_i, i = 1, \dots, q$ be the locations of facilities or centroids, $x_i \in \Re^2$. Given a point $s_j \in S$, we initially calculate the Euclidian distance from s_j to the nearest centroid:

$$z_j = \min_{i=1,\dots,q} \|x_i - s_j\|_2. \tag{1}$$

The general Multisource Weber problem, where each location $s_j, j = 1, \dots, m$ and each centroid $x_i, i = 1, \dots, q$ are in an Euclidian space with n dimensions $\Re^n$, consists of the location of q facilities in order to minimize the total transportation cost:

$$\begin{aligned} & minimize \sum_{j=1}^{m} w_j z_j \\ & \text{subject to} \quad z_j = \min_{i=1,\dots,q} \|x_i - s_j\|_2, \qquad j = 1, \dots, m. \end{aligned} \tag{2}$$

The continuous k-Median problem, also named the minimum sum-of-distances clustering problem, is simply a particular variation of problem (2), where the demands assume unitary values, $w_j = 1, j = 1, \dots, m$:

$$\begin{aligned} & minimize \sum_{j=1}^{m} z_j \\ & \text{subject to} \quad z_j = \min_{i=1,\dots,q} \|x_i - s_j\|_2, \qquad j = 1, \dots, m. \end{aligned} \tag{3}$$

In order to obtain a completely differentiable formulation, we perform a series of transformations. First let us perform a relaxation of the equality constraints:

$$\begin{aligned} & minimize \sum_{j=1}^{m} w_j z_j \\ & \text{subject to} \quad z_j - \|x_i - s_j\|_2 \le 0, \qquad j = 1, \dots, m, \quad i = 1, \dots, q. \end{aligned} \tag{4}$$

Since the z_j variables are not bounded from below, in the intrinsic minimization procedure, $z_j \to -\infty$, $j = 1, \dots, m$. In order to obtain the desired equivalence, we must, therefore, modify problem (4). We do so by first letting $\varphi(y)$ denote $max\{0, y\}$ and then observing that, from the set of inequalities in (4), it follows that

$$\sum_{i=1}^{q} \varphi(z_j - \|x_i - s_j\|_2) = 0, \qquad j = 1, \dots, m. \tag{5}$$

Using (5) in place of the set of inequality constraints in (4), we would obtain an equivalent problem, maintaining the undesirable property that $z_j, j = 1, \dots, m$ still has no lower bound. Considering, however, that the objective function of problem (4) will force each $z_j, j = 1, \dots, m$, downward, we can think of bounding the latter variables from below by including a perturbation $\varepsilon > 0$ in (5). So, we obtain the following modified problem:

$$\begin{aligned} & minimize \sum_{j=1}^{m} w_j z_j \\ & \text{subject to} \quad \sum_{i=1}^{q} \varphi(z_j - \|x_i - s_j\|_2) \ge \varepsilon , \qquad j = 1, \dots, m. \end{aligned} \tag{6}$$

Since the feasible set of problem (2) is the limit of that of (6) when $\varepsilon \to 0_+$, we can then consider solving (2) by solving a

Table 1
Notation in multisource Weber formulation.

Notation	Description
$S = \{s_1, \ldots, s_m\}$	set of cities or locations
s_j	generic city or location j
m	number of locations
w_j	demand of location j
$X = \{x_1, \ldots, x_q\}$	set of centroids
x_i	generic centroid i
q	number of centroids
n	number of dimensions of the space
z_j	distance of location j to its nearest centroid
$\varphi(y)$	auxiliary function $= max\{0, y\}$
$\phi(y, \tau)$	hyperbolic smoothing function of $\varphi(y)$
τ	smoothing parameter associated to the function ϕ
θ	hyperbolic smoothing function of the Euclidean distance
γ	smoothing parameter associated to the function θ
ε	perturbation parameter
J_B	the set of boundary locations
J_G	the set of gravitational locations
J_i	the subset of gravitational locations associated to the cluster i

sequence of problems like (6) for a sequence of decreasing values for ε that approaches zero.

Analysing problem (6), the definition of function φ endows it with an extremely rigid nondifferentiable structure, which makes its computational solution very hard. In view of this, the numerical method we adopt for solving problem (1), takes a smoothing approach. From this perspective, let us define the function:

$$\phi(y, \tau) = \left(y + \sqrt{y^2 + \tau^2} \right) /2, \tag{7}$$

for $y \in \Re$ and parameter $\tau > 0$. By using function ϕ in the place of function φ, we obtain the problem:

$$minimize \ \sum_{j=1}^{m} w_j \, z_j \tag{8}$$

$$\text{subject to} \quad \sum_{i=1}^{q} \phi(z_j - \|x_i - s_j\|_2, \tau) \geq \varepsilon, \qquad j = 1, \ldots, m.$$

Now, the Euclidean distance $\|s_j - x_i\|_2$ is the single nondifferentiable component in problem (8). So, to obtain a completely differentiable problem, it is still necessary to smooth it. For this purpose, let us define the function:

$$\theta(\, s_j \,, \, x_i \,, \, \gamma \,) \ = \ \sqrt{\sum_{l=1}^{n} (s_j^l - x_i^l)^2 \ + \ \gamma^2} \tag{9}$$

where $\gamma > 0$.

By using function θ in place of the distance $\|x_i - s_j\|_2$, the following completely differentiable problem is now obtained:

$$minimize \ \sum_{j=1}^{m} w_j \, z_j, \tag{10}$$

$$\text{subject to} \quad \sum_{i=1}^{q} \phi(z_j - \theta(x_i, s_j, \gamma), \tau) \geq \varepsilon, \qquad j = 1, \ldots, m.$$

Above, the objective function minimization process of problem (10) will work for reducing the values $z_j \geq 0, j = 1, \ldots, m$, to the utmost. On the other hand, given any set of centroids $x_i, \ i = 1, \ldots, q$, each constraint is a monotonically increasing function in z_j. So, these constraints will certainly be active and problem (10) will finally be equivalent to the following problem:

$$minimize \ \sum_{j=1}^{m} w_j \, z_j, \tag{11}$$

$$\text{subject to} \quad h_j(z_j, x) \ = \ \sum_{i=1}^{q} \phi(z_j - \theta(x_i, s_j, \gamma), \tau) \ - \ \varepsilon \ = \ 0,$$

$$j = 1, \ldots, m,$$

where x denotes the set of centroids $x_i, \ i = 1, \ldots, q$.

The dimension of the variable domain space of problem (11) is $(\, nq \ + \ m\,)$. As, in general, the value of the parameter m, i.e., the cardinality of the set S of locations $s_j, j = 1, \ldots, m$ is large, problem (11) has a large number of variables. However, it has a separable structure, because each variable z_j appears only in one equality constraint. Therefore, as the partial derivative of $h(z_j, x)$ with respect to $z_j, \ j = 1, \ldots, m$ is not equal to zero, it is possible to use the Implicit Function Theorem to calculate each component $z_j, \ j = 1, \ldots, m$ as a function of the centroid variables $x_i, \ i = 1, \ldots, q$. This way, the unconstrained problem

$$minimize \ f(x) \ = \ \sum_{j=1}^{m} w_j \, z_j(x) \tag{12}$$

is obtained, where each $z_j(x)$ results from the calculation of the single zero of each equation below, since each term ϕ strictly increases together with variable z_j :

$$h_j(z_j, x) \ = \ \sum_{i=1}^{q} \phi(z_j - \theta(x_i, s_j, \gamma), \tau) \ - \ \varepsilon \ = \ 0, \qquad j = 1, \ldots, m. \tag{13}$$

Again, due to the Implicit Function Theorem, functions $z_j(x)$ have all derivatives with respect to the variables $x_i, \ i = 1, \ldots, q$, and therefore it is possible to calculate the gradient of the objective function of problem (12),

$$\nabla f(x) \ = \ \sum_{j=1}^{m} w_j \, \nabla z_j(x) \tag{14}$$

where

$$\nabla z_j(x) \ = \ - \ \nabla \, h_j(z_j, x) \ / \ \frac{\partial \, h_j(z_j, x)}{\partial \, z_j}. \tag{15}$$

In this way, it is easy to solve problem (12) by making use of any method based on first order derivative information. Finally, it must be emphasized that problem (12) is defined on a (nq)−dimensional space, so it is a small problem, since the number of facilities, q, is, in general, very small for real applications.

The solution of the original location problem can be obtained by using the HSMFW algorithm (HSMFW — Hyperbolic Smoothing approach to the Multisource Fermat–Weber problem) which solves an infinite sequence of optimization problems, where parameters ε, τ and γ are gradually reduced to zero, just as in other smoothing methods.

Algorithm 1: Simplified HSMFW Algorithm.

1: (Initialization Step).

- Choose initial values: x^0, γ^1, τ^1, ε^1.
- Choose reduction factors: $0 < \rho_1, \rho_2, \rho_3 < 1$.
- Set $k := 1$.

2: (Main Step) Repeat until a stopping criterion is met

- Solve the problem minimize $\sum_j w_j z_j(x)$ (12), by calculating each zero of the equation (13) by using the smoothing parameters: $\gamma = \gamma^k$, $\tau = \tau^k$ and $\varepsilon = \varepsilon^k$, and by calculating each term of the gradient by using expressions (14) and (15).
- Let x^k be the solution obtained.
- Perform the updating procedure:
  - Set $\gamma^{k+1} := \rho_1 \gamma^k$, $\tau^{k+1} := \rho_2 \tau^k$, $\varepsilon^{k+1} := \rho_3 \varepsilon^k$.
  - Set $k := k + 1$.

Just as in other smoothing methods, the solution to the multisource Fermat–Weber problem is obtained, in theory, by solving an infinite sequence of optimization problems. In the HSMFW algorithm, each problem that is minimized is unconstrained, completely differentiable and of low dimension.

It should be noticed that when the algorithm causes τ and γ to approach 0_+, the constraints of the sub-problems, as given in (10), tend to those of (6). In addition, the algorithm causes ε to approach 0_+, so, in a simultaneous movement, solving problem (6) gradually approaches the original location problem (2).

3. The accelerated hyperbolic smoothing multisource fermat Weber method

The authors introduced in [3] the fast Accelerated Hyperbolic Smoothing Clustering Method for solving the minimum sum-of-squares clustering (MSSC) problem. Below we describe a similar methodology applied specifically to the Multisource Fermat–Weber Problem.

The basic idea is the partition of the set of locations into two non-overlapping regions. By using a conceptual presentation, the first region J_B, the set of boundary locations, corresponds to the location points that are close to two or more centroids, within a specified $\delta > 0$ tolerance. The second region J_G, the set of gravitational locations, corresponds to the location points that are significantly close to a unique centroid in comparison with the other centroids. Considering this partition, the problem (12) can be expressed as:

$$\text{minimize } f(x) = f_B(x) + f_G(x) = \sum_{j \in J_B} w_j\, z_j(x) + \sum_{j \in J_G} w_j\, z_j(x). \quad (16)$$

The first part of the expression (16), associated with the boundary locations, can be approximated by using the smoothing approach, see (12) and (13). The second part of the expression (16) can be calculated by using a fast procedure:

$$f_G(x) = \sum_{i=1}^{q} \sum_{j \in J_i} w_j\, \left\| x_i - s_j \right\|, \quad (17)$$

where J_i represents the subset of gravitational locations associated to the cluster i. Moreover, as the Euclidean distance is nondifferentiable in (17), the following smooth approximation is adopted

$$f_G(x) = \sum_{i=1}^{q} \sum_{j \in J_i} w_j\, \theta(x_i, s_j, \gamma). \quad (18)$$

The gradient of the second smoothed part of the objective function is easily calculated by:

$$\nabla f_G(x) = \sum_{i=1}^{q} \sum_{j \in J_i} w_j\, (x_i - s_j)/\theta(x_i, s_j, \gamma), \quad (19)$$

where the vector $(x_i - s_j)$ must be in ^{nq}, so it has the first $(i-1)q$ and the last $l = iq + 1, \ldots, nq$ components equal zero.

The efficiency of the AHSMFW, which is the previous HSMFW Method connected with the Boundary and Gravitational Regions Partition Scheme, depends naturally on parameter δ, since it defines the crucial partition of the set of locations. A choice of a large value will imply a decrease in the number of gravitational location points and, therefore, the computational advantages given by formulation (17) will be reduced. In an extreme situation, the boundary set J_G can become empty, which implies a return to formulation (12) with the hard task of determining zeros of m Eqs. (13). Otherwise, a choice of a small value for it will imply a sharp and inadequate specification of the set J_B and frequent violation of the sufficient basic condition $\Delta x < \delta$, for the validity of a partition, where Δx is the maximum displacement of the centroids. In an extreme situation, the set J_B can become empty, which implies the complete isolation of each one of q gravitational regions. The algorithm does not take into consideration the occurrence of empty gravitational regions. This possibility can be overcome by simply splitting clusters with greater inertia.

Algorithm 2: Simplified Version of Accelerated HSMFW Algorithm.

1: (Initialization Step).

- Choose the starting point x^0.
- Choose values of smoothing parameters: γ^1, τ^1, ε^1.
- Choose reduction factors: $0 < \rho_1, \rho_2, \rho_3 < 1$.
- Specify the boundary band width: $\delta^1 > 0$.
- Set $k := 1$.

2: (Main Step) Repeat until a stopping criterion is met

- Use $\bar{x} = x^{k-1}$ and $\delta = \delta^k$ to determine the non-overlapping $Z_\delta(\bar{x})$ and $G_\delta(\bar{x})$ partitions and corresponding sets J_i, $i = 1, \ldots, q$.
- Solve problem (16) starting at the initial point x^{k-1}:
  - For the first part $\sum_{j \in J_B} w_j z_j(x)$, associated with the boundary band zone, calculate each zero of the equation (13) using the smoothing parameters: $\gamma = \gamma^k$, $\tau = \tau^k$ and $\varepsilon = \varepsilon^k$, and calculate each term of the gradient by using expressions (14) and (15).
  - For the second part, $\sum_{j \in J_G} w_j z_j(x)$, associated with gravitational regions, use expressions (18) and (19).
- Let x^k be the solution obtained.
- Perform the updating procedure:
  - Set $\gamma^{k+1} := \rho_1 \gamma^k$, $\tau^{k+1} := \rho_2 \tau^k$, $\varepsilon^{k+1} := \rho_3 \varepsilon^k$.
  - Redefine the boundary value: δ^{k+1}.
  - Set $k := k + 1$.

As a general strategy, within the first iterations, larger δ values are used, because of more expressive centroid displacements. The

δ values would be dynamically updated considering the sizes of these displacements. In any case, the inexpensive partition procedure is always performed before the hard optimization one. So, it is possible to identify a priori any deficiency in the resulting partition and correct it, by increasing or decreasing δ , according to the case diagnosed.

4. Computational results

The computational results presented below were obtained from a preliminary implementation of the AHSMFW algorithm. The numerical experiments have been carried out on an Intel Core i7-2620M Windows Notebook with 2.70 GHz and 8GB RAM, where the programs were compiled by an Intel Fortran Compiler. Only one thread was used in all experiments. The unconstrained minimization tasks were carried out by means of a Quasi-Newton algorithm employing the BFGS updating formula from the Harwell Library, available in:

http://www.cse.scitech.ac.uk/nag/hsl/

In order to show the performance of the proposed algorithm, we present results obtained by solving six very large instances, all taken from classical travelling salesman problem benchmark collections. We assume demands with unitary values: $w_j = 1$, $j = 1, \dots, m$.

- U1060 and Pla85900, which use points in the plane from the TSPLIB collection of challenge problems of Reinelt [25]; http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
- sra104815, ara238025, lra498378 and lrb744710 which are instances taken from the Waterloo TSP; http://www.math.uwaterloo.ca/tsp/
- APL1243088, which is a synthetic instance obtained by performing a simple concatenation of the two instances, without any transformation, lra498378 and lrb744710 taken from the Waterloo TSP.

The AHSMFW is a general framework that bears a broad numbers of implementations. In the initialization steps, the following choices were made for the reduction factors: $\rho_1 = 1/4$, $\rho_2 = 1/4$ and $\rho_3 = 1/4$. The specification of initial smoothing and perturbation parameters was automatically tuned to the problem data. So, the initial smoothing parameter τ of the function $\phi(y, \tau)$(7) was specified by $\tau^1 = \sigma / 10$, where σ^2 is the variance of the set of cities, or locations: $S = \{s_1, \dots, s_m\}$. The initial perturbation parameter ε used in the specification of problem (6) was specified by $\varepsilon^1 = 4\tau^1$ and the Euclidian distance smoothing parameter γ of the function (9) by $\gamma^1 = \tau^1 / 100$.

Table 2 presents a comparison between the computational results obtained by AHSMFW algorithm for solving the instance U1060 with the results published by Drezner et al. [12], where the programs were compiled by an Intel 11.1 Fortran Compiler with no parallel processing using double precision arithmetic and run on a desktop with an Intel 870/i7 2.93 GHz CPU Quad processor and 8GB RAM and where only one thread was used. The first column of Table 2 presents the specified number of facilities q. The second to fourth column presents the values obtained by [12], respectively: f_{Ref} , the best value of the object function by using a set of 100 random starting points, the perceptual deviation value D_{Ref} of the set of solutions related to this best value and the mean CPU time given in seconds $Time_{Ref}$. The percentage deviation in all 100 attempts is calculated by $D_{Ref} = \sum_{t=1}^{100} (f^t - f_{Ref}) / f_{Ref}$, where f^t is the value of the function obtained from attempt t. The fifth to seventh columns present the similar values obtained by AHSMFW, respectively: f_{HS}, D_{HS} and $Time_{HS}$. The next column shows the perceptual relative difference Δf (%) between the objective function values f_{HS} and f_{Ref}. The last column shows the Times Ratio, *TR*, between times $Time_{Ref}$ and $Time_{HS}$, which is a valid indicator since both computers have a similar performance.

Table 2
Results for the TSPLIB1060 instance.

q	f_{Ref}	D_{Ref}	$Time_{Ref}$	f_{HS}	D_{HS}	$Time_{HS}$	Δf (%)	TR
5	1851877	0	18.8	1855160	0.10	0.07	0.18	104
10	1249564	0	31.4	1252010	0.28	0.14	0.20	224
15	980131	0	51.2	982476	0.59	0.22	0,24	232
20	828685	0	63.9	831521	0.67	0.34	0,34	188
25	721988	0.003	84.4	724435	0.53	0.47	0,34	180
30	638212	0	95.0	641061	0.47	0.65	0,45	146
35	577496	0	109.4	580891	0.71	0.98	0,58	112
40	529660	0.024	142.8	532971	0.68	1.36	0,63	105
45	489483	0.056	170.2	491921	0.81	1.64	0,50	104
50	453109	0.027	191.8	455306	1.03	1.99	0,48	96
55	422638	0.024	217.1	425268	1.32	2.47	0,62	88
60	397674	0.016	230.1	400867	1.09	2.98	0,80	77
65	376630	0.029	269.7	380976	0.82	3.54	1,15	76
70	357335	0.032	308.6	362423	0.70	4.08	1,42	76
75	340123	0.001	332.2	344278	0.81	4.72	1,22	70
80	325971	0.004	365.1	329312	0.75	5.37	1,02	68
85	313446	0.044	413.4	316710	0.66	6.00	1,04	70
90	302479	0.021	468.7	304728	0.81	7.21	0,74	65
95	292282	0.045	532.9	294986	0.74	7.86	0,93	68
100	282536	0.097	598.6	285455	0.79	8.48	1,03	71

The results presented in Table 2 show a consistent performance of both algorithms, since columns D_{Ref} and D_{HS} present small values, but the referential algorithm [12] exhibits an extremely consistent performance. In the same manner, the referential algorithm obtains better values of the objective function in all cases. However, the proposed algorithm produces values close to these solutions, with the maximum percent relative difference of 1.42%. The disadvantage in accuracy, as well as in consistency, can be explained by the elaborate strategies used by Drezner et al. [12] in each attempt, which engender starting points with excellent quality. In contrast, the implementation of the algorithm ASHMFW makes use of a very naive strategy, where the starting points are generated as small disturbances around the point of the best solution ever produced in the iterative procedure. On the other hand, the comparison between the mean CPU times spent by the two algorithms demonstrates a categorical superiority of the proposed algorithm, since the last column Times Ratio (*TR*), shows large values, in the range 65 to 232.

Table 3 presents the computational results obtained by ASHMFW algorithm for solving the instance Pla85900. The first column of Table 3 presents the specified number of facilities q. The second column presents f_{HS}, the best objective function value produced by the old HSMFW, results previously presented by [24], else by the now introduced accelerated AHSMFW algorithm, both by using a set of 100 random starting points. It must be pointed out that both algorithms obtained almost the same best results, with a five decimal digit precision. The next four columns present the results obtained by the new AHSMFW: the number of occurrences of the best solution *Occur*.; the perceptual deviation value D_{HS} of the set of solutions related to this best value, the mean CPU time given in seconds $Time_{HS}$ and the speed-up on the CPU time produced by the new proposed algorithm AHSMFW, column $Speed-Up$, in relation to the previous HSMFW value originally presented also in [24].

The results in Table 3 show a consistent performance of the AHSMFW algorithm, since column D_{HS} present small values. Column *Occur*. displays expressive values, particularly for small cases, up to 10 facilities, when there are less local minima points, which indicates again a consistent performance of AHSMFW. Column $Time_{HS}$ indicates small CPU times in seconds, since this

Table 3
New results for the Pla85900 instance.

q	f_{HS}	$Occur.$	D_{HS}	$Time_{HS}$	$Speed - Up$
2	0.163630F11	100	0.00	1.64	15.4
3	0.127842F11	100	0.00	2.48	20.6
4	0.108069F11	100	0.00	2.88	25.9
5	0.984600F10	97	0.01	3.84	31.5
6	0.902578F10	75	0.09	4.24	36.9
8	0.778303F10	53	0.03	4.72	55.3
10	0.704186F10	28	0.26	5.09	74.9
15	0.576996F10	1	0.49	6.29	149.1
20	0.502042F10	1	0.44	7.18	235.4
30	0.412451F10	1	0.72	9.09	447.0
40	0.358763F10	1	0.62	11.11	735.3

Table 4
Results for the sra104815 instance.

q	f_{HS}	$Occur.$	D_{HS}	$Time_{HS}$
2	0.436271F8	100	0.00	1.77
3	0.336853F8	100	0.00	2.03
4	0.263551F8	100	0.00	2.48
5	0.237265F8	87	0.07	2.87
6	0.214195F8	78	0.18	3.11
8	0.178438F8	90	0.23	3.68
10	0.160078F8	16	0.32	4.80
12	0.145478F8	19	0.93	6.31
15	0.129872F8	7	0.66	6.31
20	0.110610F8	1	0.92	7.29
25	0.972848F7	1	1.26	8.85

is a large instance with 85900 locations. The most important analysis derived from Table 3 is the comparison between the mean CPU times spent by the two algorithms, given by the speed-up factor in the last column. For the Pla85900 instance, on cases from $q = 2$ to 40, the observed speed-up factors present a monotonic increasing from 15.4 to 735.3. This high speed of the ASHMFW algorithm can be attributed to the partition of the set of locations into two nonoverlapping parts. This approach offers a drastic simplification of computational tasks.

Tables 4–8 adopt the same notation, but they contain only the first five columns of Table 3 and exhibit only results produced by the new ASHMFW, since it was not possible to find any previous solutions for these instances.

Similarly, the results presented in Tables 4–8 corresponding to the five test problems sra104815, ara238025, lra498378, lrb744710 and APL1243088, show a consistent performance of the AHSMFW algorithm. However, it was impossible to find any previous record of solutions for these last instances by any algorithm. Indeed, the instance U1060 of the TSPLIB collection of Reinelt [25] is the largest that we found in the literature, see for example [6,8–10,26]. We emphasize that the instance APL1243088 is more than 1000 times greater. Therefore, the results shown in these five tables are a challenge for future research. Moreover, the results presented in Tables 4–8 show an efficient performance of the AHSMFW algorithm, since the mean computational CPU times were consistently small, despite the large sizes of the test problems solved.

5. Conclusions

In this paper, a new method for the solution of the Multisource Fermat–Weber problem is proposed by applying two key procedures. First, by using the Hyperbolic Smoothing technique, the problem was reformulated, in an approximation approach, as a completely differentiable constrained optimization problem. By applying the Implicit Function Theorem, the problem was further reformulated as a low dimension unconstrained optimization problem, in a low Euclidean space $\Re^{nq}$. Second, by performing the partition of the locations set in a boundary band zone and gravitational regions, it is possible to extremely simplify the computational tasks.

The good performance of the novel AHSMFW algorithm, according to different criteria – accuracy, efficiency and consistency – can be attributed to both procedures: the complete differentiability of the optimization problem defined in a low dimension space and the simplification of the computational calculations engendered by the partition scheme.

Apart from consistently presenting similar or even better results, based on the accuracy criterion, when compared to related

Table 5
Results for the ara238025 Instance.

q	f_{HS}	$Occur.$	D_{HS}	$Time_{HS}$
2	0.140516F9	100	0.00	6.15
3	0.106822F9	100	0.00	7.02
4	0.926876F8	99	0.01	7.17
5	0.802255F8	100	0.00	7.92
6	0.729352F8	89	0.13	8.08
8	0.629411F8	60	0.24	9.12
10	0.561957F8	14	0.53	10.30
12	0.512001F8	5	0.84	11.48
15	0.452994F8	2	1.03	13.60
20	0.386577F8	4	1.81	14.63
25	0.347061F8	1	1.96	14.68

Table 6
Results for the lra498378 instance.

q	f_{HS}	$Occur.$	D_{HS}	$Time_{HS}$
2	0.896006F9	100	0.00	9.92
3	0.748544F9	100	0.00	11.83
4	0.629745F9	100	0.00	12.47
5	0.530382F9	97	0.31	14.05
6	0.481371F9	57	1.06	14.15
8	0.416231F9	39	0.62	16.15
10	0.372064F9	1	1.75	18.17
12	0.326851F9	3	2.89	20.00
15	0.286375F9	1	2.69	22.14
20	0.242253F9	2	3.14	26.34
25	0.216916F9	1	2.83	30.88

Table 7
Results for the lrb744710 instance.

q	f_{HS}	$Occur.$	D_{HS}	$Time_{HS}$
2	0.689546F9	100	0.00	16.83
3	0.516142F9	100	0.00	17.34
4	0.448308F9	100	0.00	19.25
5	0.399651F9	98	0.05	20.98
6	0.362778F9	86	0.31	23.03
8	0.314863F9	30	0.29	27.03
10	0.282256F9	10	0.20	32.63
12	0.251326F9	41	0.43	33.41
15	0.223052F9	6	0.66	37.31
20	0.192454F9	3	0.74	44.15
25	0.171865F9	1	0.54	50.23

Table 8
Results for the APL1243088 (Lra 498378 + Lrb 744710) Instance.

q	f_{HS}	$Occur.$	D_{HS}	$Time_{HS}$
2	0.229685F10	100	0.00	33.13
3	0.181327F10	91	1.18	48.51
4	0.157197F10	100	0.00	57.67
5	0.139735F10	98	0.04	60.90
6	0.125196F10	22	2.65	46.91
8	0.103438F10	4	8.87	46.52
10	0.917266F09	3	10.08	51.55
12	0.832075F09	1	9.45	57.64
15	0.740971F09	1	9.42	63.55
20	0.637078F09	1	9.62	77.81
25	0.565842F09	1	9.16	87.62

approaches, the novel technique was able to get solutions faster and to deal large instances never tackled before, with up to 1243088 cities.

Notwithstanding these good records, a desirable future work is the development of a new procedure for determination of the starting point, insofar as the strategy currently used is very naive. The choice of good starting points should further improve the performance of the algorithm, in particular according to the criteria of accuracy and consistency.

As Brimberg et al. consigns in [26], the exploration of the relation between the discrete p-median model and the continuous location–allocation model has been proposed in many works. As a future work, in a first phase, we are planning to use this relation in a reverse direction, where the novel AHSMFW methodology would be applied as an auxiliary tool for solving the discrete p-median problem. On an initial phase, it would be used to obtain fast and good approximate continuous solution. In a second phase, it would take this solution as an auxiliary point for the refining procedure in order to calculate, ultimately, the optimal, or at least a deep solution for the discrete problem, by using an adequate algorithm, such as that proposed by [27].

Finally, it must be noticed that the Multisource Fermat–Weber problem is a global optimization problem with a very large number of local minima, so that the proposed algorithm can only produce local minima.

Acknowledgment

Adilson Elias Xavier was supported by Brazilian Council for Scientific and Technological Development, Research Productivity Grant.

References

[1] G.O. Wesolowsky, The Weber problem: History and perspectives, Comput. Oper. Res.
[2] A. Rubinov, Methods for global optimization of nonsmooth functions with applications, Appl. Comput. Math. 5 (1) (2006) 3–15.
[3] A.E. Xavier, V.L. Xavier, Solving the minimum sum-of-squares clustering problem by hyperbolic smoothing and partition into boundary and gravitational regions, Pattern Recognit. 44 (1) (2011) 70–77.
[4] A.M. Bagirov, B. Ordin, G. Ozturk, A.E. Xavier, An incremental clustering algorithm based on hyperbolic smoothing, Comput. Optim. Appl. 61 (1) (2015) 219–241.
[5] T.C. Koopmans, M. Beckmann, Assignment problems and the location of economic activities, Econometrica 25 (1) (1957) 53–76, URL http://www.jstor.org/stable/1907742.
[6] J. Brimberg, P. Hansen, N. Mladonovic, S. Salhi, A survey of solution methods for the continuous location allocation problem, Int. J. Oper. Res. 5 (1) (2008) 1–12.
[7] L. Cooper, Heuristic methods for location–allocation problems, SIAM Rev. 6 (1) (1964) 37–53.
[8] I. Bongartz, P.H. Calamai, A.R. Conn, A projection method forl p norm location–allocation problems, Math. Program. 66 (1–3) (1994) 283–312.
[9] J. Brimberg, P. Hansen, N. Mladenović, E.D. Taillard, Improvements and comparison of heuristics for solving the uncapacitated multisource weber problem, Oper. Res. 48 (3) (2000) 444–460.
[10] Z. Drezner, J. Brimberg, N. Mladenović, S. Salhi, Solving the planar p-median problem by variable neighborhood and concentric searches, J. Global Optim. 63 (3) (2015) 501–514.
[11] J. Brimberg, Z. Drezner, A new heuristic for solving the p-median problem in the plane, Comput. Oper. Res. 40 (1) (2013) 427–437.
[12] Z. Drezner, J. Brimberg, N. Mladenović, S. Salhi, New local searches for solving the multi-source weber problem, Ann. Oper. Res. 246 (1–2) (2016) 181–203.
[13] J. Reese, Solution methods for the p-median problem: An annotated bibliography, Networks 48 (3) 125–142. http://dx.doi.org/10.1002/net.20128.
[14] M.S. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, John Wiley & Sons, 2011.
[15] N. Mladenović, J. Brimberg, P. Hansen, J.A. Moreno-Pérez, The p-median problem: A survey of metaheuristic approaches, European J. Oper. Res. 179 (3) (2007) 927–939, http://dx.doi.org/10.1016/j.ejor.2005.05.034, URL http://www.sciencedirect.com/science/article/pii/S0377221706000750.
[16] M.G. Resende, R.F. Werneck, A fast swap-based local search procedure for location problems, Ann. Oper. Res. 150 (1) (2007) 205–230.
[17] A. Bagirov, A. Al Nuaimat, N. Sultanova, Hyperbolic smoothing function method for minimax problems, Optimization 62 (6) (2013) 759–782.
[18] A.E. Xavier, A.A.F. De Oliveira, Optimal covering of plane domains by circles via hyperbolic smoothing, J. Global Optim. 31 (3) (2005) 493–504.
[19] D.C. Lubke, V.L. Xavier, H.M. Venceslau, A.E. Xavier, Flying elephants method applied to the problem of covering solid bodies with spheres, Int. J. Metaheurist. 7 (1) (2018) 30–42.
[20] A.E. Xavier, C.M. Gesteira, V.L. Xavier, Solving the continuous multiple allocation p-hub median problem by the hyperbolic smoothing approach, Optimization 64 (12) (2015) 2631–2647.
[21] A.E. Xavier, V.L. Xavier, Flying elephants: a general method for solving non-differentiable problems, J. Heuristics 22 (4) (2016) 649–664.
[22] A.E. Xavier, Penalização hiperbólica: Um novo método para resolução de problemas de otimização (Ph.D. thesis), M. Sc. Thesis-COPPE-UFRJ, Rio de Janeiro, 1982.
[23] A.E. Xavier, Hyperbolic penalty: a new method for nonlinear programming with inequalities, Int. Trans. Oper. Res. 8 (6) (2001) 659–671.
[24] V.L. Xavier, F.M. França, A.E. Xavier, P.M. Lima, A hyperbolic smoothing approach to the multisource weber problem, J. Global Optim. 60 (1) (2014) 49–58.
[25] G. Reinelt, Tsplib—a traveling salesman problem library, ORSA J. Comput. 3 (4) (1991) 376–384.
[26] J. Brimberg, Z. Drezner, N. Mladenović, S. Salhi, A new local search for continuous location problems, European J. Oper. Res. 232 (2) (2014) 256–265.
[27] J. Brimberg, Z. Drezner, N. Mladenović, S. Salhi, Using injection points in reformulation local search for solving continuous location problems, Yugosl. J. Oper. Res. 27 (3) (2016) 291–300.