J Glob Optim (2013) 56:983–1005
DOI 10.1007/s10898-012-9893-4

A two-level evolutionary algorithm for solving the facility location and design (1|1)-centroid problem on the plane with variable demand

J. L. Redondo · A. G. Arrondo · J. Fernández · I. García · P. M. Ortigosa

Received: 12 January 2011 / Accepted: 12 March 2012 / Published online: 13 April 2012

Abstract In this work, the problem of a company or chain (the leader) that considers the reaction of a competitor chain (the follower) is studied. In particular, the leader wants to set up a single new facility in a planar market where similar facilities of the follower, and possibly of its own chain, are already present. The follower will react by locating another single facility after the leader locates its own facility. Both the location and the quality (representing design, quality of products, prices, etc.) of the new leader's facility have to be found. The aim is to maximize the profit obtained by the leader considering the future follower's entry. The demand is supposed to be concentrated at n demand points. Each demand point splits its buying power among the facilities proportionally to the attraction it feels for them. The attraction of a demand point for a facility depends on both the location and the quality of the facility. Usually, the demand is considered in the literature to be fixed or constant regardless the conditions of the market. In this paper, the demand varies depending

J. L. Redondo (✉)
Department of Computer Architecture and Technology,
University of Granada, Granada, Spain
e-mail: jlredondo@atc.ugr.es

A. G. Arrondo · J. Fernández
Department of Statistics and Operations Research,
University of Murcia, Murcia, Spain
e-mail: agarrondo@um.es

J. Fernández
e-mail: josefdez@um.es

I. García
Department of Computer Architecture,
University of Málaga, Malaga, Spain
e-mail: igarciaf@uma.es

P. M. Ortigosa
Department of Computer Architecture and Electronics,
University of Almería, ceiA3, Almeria, Spain
e-mail: ortigosa@ual.es

on the attraction for the facilities. Taking variable demand into consideration makes the model more realistic. However, it increases the complexity of the problem and, therefore, the computational effort needed to solve it. Three heuristic methods are proposed to cope with this hard-to-solve global optimization problem, namely, a grid search procedure, a multistart algorithm and a two-level evolutionary algorithm. The computational studies show that the evolutionary algorithm is both the most robust algorithm and the one that provides the best results.

Keywords Nonlinear bi-level programming problem · Centroid (or Stackelberg) problem · Continuous location · Competition · Variable demand · Evolutionary algorithm · Multistart heuristic · Grid search

1 Introduction

Location science deals with the location of one or more facilities in a way that optimizes a certain objective (minimization of transportation costs, minimization of social costs, maximization of market share, etc.). For an introduction to the topic see [1,2]. Depending on whether a single player or multiple players are considered in the market, we can distinguish between *non-competitive* and *competitive* location models. In the former, it is assumed that the decision maker, who plans the location of his facilities, faces an empty space without any similar or competing facilities. In the latter, similar facilities already exist in the region (as in most of the cases in reality) and the task is to add new ones in an optimal way. The existing facilities may belong to the decision maker's own chain or to a competitor's chain.

Many competitive location models are available in the literature (see for instance the survey papers [3–5]), which vary in the ingredients which form the model. For instance, we may want to locate just a single facility or more than one new facility. The demand (usually supposed to be concentrated in a discrete set of points, called *demand points*) can be either *inelastic* or *elastic*, depending on whether the goods are essential or inessential. The *patronizing behaviour* of the customers is usually thought to be either *deterministic*, when the full demand of the customer is served by the facility to which he/she is *attracted* most (leading to Hotelling-type models) or *probabilistic*, when the customer splits his/her demand among all the existing facilities (leading to Huff-type models). The *attraction (or utility) function* of a customer towards a given facility, which usually depends on the distance between the customer and the facility, as well as on other characteristics of the facility which determine its *quality*, is also a key factor to be specified. The *market share* captured by the facilities depends on all those factors.

Furthermore, when a competition takes place, it may be *static*, which means that the competitors are already in the market, the owner of the new facility knows their characteristics and no reaction is expected from them (see [5]), or *with foresight*, in which the competitors are assumed to react after the new facility enters (see [6]). The maximization of profit for each competing firm can then be seen as a location game, which has been studied since the work of Hotelling [7]. If the competitors can change their decisions, then we have a *dynamic model*, in which the existence of *equilibrium* situations is of major concern.

In this context, Hakimi [8] introduced the well known Stackelberg problems (also known as Simpson's problems in voting theory). The scenario considered in this kind of problems is that of a *duopoly*. A chain (the leader) wants to set up p new facilities in the market, where similar facilities of a competitor (the follower), and possibly of its own chain, are already present. The follower will react by locating r facilities after the leader locates its own facilities. Hakimi introduced the terms *medianoid* for the follower problem, and *centroid* for the leader problem. More precisely, an $(r|X_p)$ medianoid problem refers to the follower's problem of locating r new facilities in the presence of p leader's facilities located at a set of points X_p. An $(r|p)$ centroid problem refers to the leader's problem of locating p new facilities, knowing that the follower will react positioning r new facilities by solving an $(r|X_p)$ medianoid problem.

In this paper, a constrained (1|1) centroid problem in the plane with Huff patronizing behaviour, in which the quality of the facility is regarded as a third decision variable of the model, is considered. And for the first time in the literature on centroid problems, elastic demand is contemplated. This problem is a hard-to-solve global optimization problem, with many local maxima and in some instances with very different objective values at quite close feasible points. The literature on centroid problems is scarce (see [6] for a review on the topic until 1996), and to our knowledge, among the existing papers only five of them deal with continuous problems. This is mostly due to the complexity of that type of bi-level programming problems. Drezner [9] solved the (1|1) centroid problem for the Hotelling model and Euclidean distances exactly, through a geometric-based approach. Bhadury et al. [10] also considered the $(r|p)$ centroid problem for the Hotelling model with Euclidean distances, and gave an alternating heuristic to cope with it. Drezner and Drezner [11] considered the Huff model, and proposed three heuristic approaches for handling the (1|1) centroid problem (see also [12]). More recently, Redondo et al. [13] introduced four heuristics for handling a (1|1) centroid problem with Huff patronizing behaviour and with the quality of the new facility as a variable of the problem. In all those papers the demand was assumed to be fixed.

In this paper, a (1|1) centroid problem similar to that in [13] is considered, but in which the demand varies depending on the attraction for the facilities. Three procedures are introduced for handling it, namely, a grid search procedure, a multistart heuristic and an evolutionary algorithm. Additionally, a local search procedure, called SASS+WLMv, has been proposed to be used in both the multistart and the evolutionary algorithms.

It is important to mention that to solve a single centroid problem, many medianoid problems have to be solved, since the evaluation of the leader's objective function at a given point requires the resolution of a medianoid problem. In this sense, this problem can be considered a two-level optimization problem [14,15]. Of course, it is highly important to compute the leader's objective function value accurately, which means that the follower's problem has to be solved with precision. However, the medianoid problem is also a hard-to-solve global optimization problem (as most competitive location problems are). Recently, in [16], the $(1|X_1)$ medianoid problem with Huff patronizing behaviour, elastic demand and considering the quality as variable of the problem, has been studied and solved using the evolutionary algorithm UEGO (*Universal Evolutionary Global Optimizer*), initially described in [17], and an exact interval branch-and-bound method (iB&B) [18]. The computational studies showed that the heuristic algorithm UEGO was reliable, always finding the global optimum. The computational effort of both algorithms, in terms of computing time and memory requirements, varied depending on the size of the problem. For the problem at hand, both algorithms are considered as alternatives to deal with the medianoid problems.

The paper is organized as follows: In Sect. 2, the centroid problem is introduced. The procedures used to solve the corresponding medianoid problem are outlined in Sect. 3. It is in Sect. 4 where we describe the procedures for solving the centroid problem. Computational studies are presented in Sect. 5 and the paper ends with some conclusions and lines for future research in Sect. 6.

2 A Huff-like (1|1)-centroid problem with decisions in both location and quality

2.1 Fixed demand case

A chain, the *leader*, wants to locate a new single facility in a given area of the plane, where m facilities offering the same goods or product already exist. The first k (≥ 0) of those m facilities belong to the chain, and the other $m - k$ (>0) to a competitor chain, the *follower*. The leader knows that the follower, as a reaction, will subsequently position a new facility too. The demand, supposed to be inelastic, is concentrated at n demand points, whose locations p_i and purchasing power (or planed budget to buy goods) $\widehat{w}_i$ are known. The location f_j and quality of the existing facilities are also known.

The following notation will be used throughout this paper:

Indices

i	Index of demand points, $i = 1, \ldots, n$.
j	Index of existing facilities, $j = 1, \ldots, m$.

Variables

$z_1 = (x_1, y_1)$	Location of the new leader's facility.
α_1	Quality of the new leader's facility.
$nf_1 = (z_1, \alpha_1)$	Variables of the new leader's facility.
$z_2 = (x_2, y_2)$	Location of the new follower's facility.
α_2	Quality of the new follower's facility.
$nf_2 = (z_2, \alpha_2)$	Variables of the new follower's facility.

Data

p_i	Location of the ith demand point.
$\widehat{w}_i$	Demand (or purchasing power) at p_i.
f_j	Location of the jth existing facility.
ϵ_i	Minimum distance from p_i at which the new facilities can be located.
$d_{i,j}$	Distance between p_i and f_j.
$a_{i,j}$	Quality of f_j as perceived by p_i.
$g_i(\cdot)$	A non-negative non-decreasing function.
$u_{i,j}$	Attraction that p_i feels for f_j (or utility of f_j perceived by the People at p_i), $u_{i,j} = a_{i,j}/g_i(d_{i,j})$
γ_i	Weight for the quality of the new facilities as perceived by demand point p_i.
S_1	Location space where the leader will locate its new facility.
$\alpha_1^{\min}$	Minimum level of quality for the new leader's facility.
$\alpha_1^{\max}$	Maximum level of quality for the new leader's facility.
S_2	Location space where the follower will locate its new facility.
$\alpha_2^{\min}$	Minimum level of quality for the new follower's facility.
$\alpha_2^{\max}$	Maximum level of quality for the new follower's facility.

Miscellaneous

d_{i,z_l}	Distance between p_i and z_l, $l = 1, 2$.
u_{i,nf_l}	Attraction that p_i feels for nf_l, $l = 1, 2$, $u_{i,nf_l} = \gamma_i \alpha_l / g_i(d_{i,z_l})$.
$M_1(nf_1, nf_2)$	Market share obtained by the leader after the location of the new facilities.

$M_2(nf_1, nf_2)$	Market share obtained by the follower after the location of the new facilities.
$\Pi_1(nf_1, nf_2)$	Profit obtained by the leader after the location of the new facilities.
$\Pi_2(nf_1, nf_2)$	Profit obtained by the follower after the location of the new facilities.

We assume that $g_i(d_{i,j}) > 0\,\forall i, j$, and consider that the patronizing behaviour of customers is probabilistic, that is, demand points split their buying power among *all* the facilities proportionally to the attraction they feel for them. Using these assumptions, the market share attracted by the leader's chain after the location of the leader and the follower's new facilities is

$$M_1(nf_1, nf_2) = \sum_{i=1}^{n} \widehat{w}_i \frac{u_{i,nf_1} + \sum_{j=1}^{k} u_{i,j}}{u_{i,nf_1} + u_{i,nf_2} + \sum_{j=1}^{m} u_{i,j}}.$$

and the corresponding market share attracted by the follower's chain is

$$M_2(nf_1, nf_2) = \sum_{i=1}^{n} \widehat{w}_i \frac{u_{i,nf_2} + \sum_{j=k+1}^{m} u_{i,j}}{u_{i,nf_1} + u_{i,nf_2} + \sum_{j=1}^{m} u_{i,j}}.$$

Given nf_1, the problem for the follower is the $(1|nf_1)$ medianoid problem:

$$(FP(nf_1)) \quad \begin{cases} \max\ \Pi_2(nf_1, nf_2) = F_2(M_2(nf_1, nf_2)) - G_2(nf_2) \\ \text{s.t.}\quad z_2 \in S_2 \\ \qquad d_{i,z_2} \geq \epsilon_i, i = 1, \ldots, n \\ \qquad \alpha_2 \in [\alpha_2^{\min}, \alpha_2^{\max}] \end{cases}$$

whose objective is the maximization of the profit obtained by the follower (once the leader has set up its new facility at nf_1), to be understood as the difference between the revenues obtained from the captured market share minus the operating costs of the new facility (see [18]). F_2 is a strictly increasing function which transforms the market share into expected sales and G_2 is a function which gives the operating cost for the follower of a facility located at z_2 with quality α_2.

Let us denote with $nf_2^*(nf_1)$ an optimal solution for $(FP(nf_1))$. The problem for the leader is the $(1|1)$ centroid problem:

$$(LP) \begin{cases} \max\ \Pi_1(nf_1, nf_2^*(nf_1)) = F_1(M_1(nf_1, nf_2^*(nf_1))) - G_1(nf_1) \\ \text{s.t.}\quad z_1 \in S_1 \\ \qquad d_{i,z_1} \geq \epsilon_i, i = 1, \ldots, n \\ \qquad \alpha_1 \in [\alpha_1^{\min}, \alpha_1^{\max}] \end{cases}$$

where F_1 and G_1 are the corresponding expected sales and operating costs functions, respectively, for the leader's chain.

In our computational studies we made the following choices:

– Functions $F_l,\ l = 1, 2$, are linear, $F_l(M) = s_l \cdot M$, where s_l is the income per unit of goods sold.
– Usually, the operating costs of a new facility consist of the sum of the locational costs and the costs related to reaching a given level of quality. Therefore functions $G_l, l = 1, 2$, are assumed to be separable, in the form $G_l(nf_l) = G_l^a(z_l) + G_l^b(\alpha_l)$. In particular, we have considered $G_l^a(z_l) = \sum_{i=1}^{n} \Phi_l^i(d_{i,z_l})$, with $\Phi_l^i(d_{i,z_l}) = \widehat{w}_i/((d_{i,z_l})^{\phi_l^{i0}} + \phi_l^{i1})$, $\phi_l^{i0}, \phi_l^{i1} > 0$ and $G_l^b(\alpha_l) = \exp(\alpha/\xi_l^0 + \xi_l^1) - \exp(\xi_l^1)$, with $\xi_l^0 > 0$ and $\xi_l^1 \in \mathbb{R}$ as given values.

A more detailed explanation of the parameters and functions of the model, as well as other possible expressions for F_l and G_l, can be found in [18]. Of course, other functions might be more suitable depending on the real problem considered, and for each real application the most appropriate F_l and G_l functions should be discovered. In [19] the interested reader can find a pseudo-real application to the case of the location of supermakets in the Autonomous Region of Murcia, in Southern Spain. Although in that paper the demand was assumed to be fixed and no reaction from the competitor was expected, the parameters and functions have the same meaning as those in the present paper.

As we can see, the leader problem (LP) is much more difficult to solve than the follower problem $(FP(nf_1))$. Notice, for instance, that to evaluate its objective function Π_1 at a given point nf_1, we have to first solve the corresponding medianoid problem $(FP(nf_1))$ to obtain $nf_2^*(nf_1)$. Furthermore, in order to compute the objective value of Π_1 at nf_1 accurately, the follower problem $(FP(nf_1))$ has to be precisely solved since otherwise, the error of the approximate value can be considerable.

2.2 Variable demand case

In the previous model the demands $\widehat{w}_i$ at the demand points are assumed to be fixed. Now, let us make the more realistic assumption that the demand at p_i is affected by the perceived utility of the facilities, given by the vector $u_i = (u_{i,nf_1}, u_{i,nf_2}, u_{i,1}, \ldots, u_{i,m})$. Making the simplifying assumption that *the utility is additive*, then $U_i = u_{i,nf_1} + u_{i,nf_2} + \sum_{j=1}^{m} u_{i,j}$ represents the total utility perceived by a customer at p_i provided by all the facilities. Hence, it is natural to assume that the actual demand at p_i is a function of U_i.

If we denote by $w_i^{\max}$ the maximum possible demand at p_i, and by $w_i^{\min}$ the minimum possible demand at p_i, then the actual demand w_i at p_i is a function of the utility vector u_i only through the total utility U_i, i.e., $w_i(U_i) = w_i^{\min} + incr_i \cdot e_i(U_i)$, where $incr_i = w_i^{\max} - w_i^{\min}$. Here, $e_i(U_i)$ is a non-negative and non-decreasing function of U_i that cannot exceed 1 (notice that w_i cannot exceed $w_i^{\max}$). Function $e_i(U_i)$ can be interpreted as the share of the maximum possible increment that a customer decides to expend under a given location scenario. Although there are different possible expressions for this function (see [16]), for the current study a *linear expenditures* case is considered. In this model, $w_i^{\min} = 0$, so that $incr_i = w_i^{\max}$, and $e_i(U_i) = c_i U_i$, with c_i a given constant such that $c_i \leq 1/U_i^{\max}$, where $U_i^{\max}$ is the maximum utility that can be possibly perceived by a customer at i, see [20].

The corresponding Huff-like (1|1)-centroid problem with elastic demand and decisions in both location and quality to be solved is analogous to (LP), but with the following modifications:

1. In functions M_l, $l = 1, 2$, $\widehat{w}_i$ is changed to $w_i(U_i)$.
2. In the cost functions $G_l(nf_l) = G_l^a(z_l) + G_l^b(\alpha_l)$, $l = 1, 2$, for $\widehat{w}_i$ a *mean* value $Aver_{A_i}(w_i(U_i))$ is used (in the sense of the first mean value theorem for integration, see [16]). Notice that we *do not* replace $\widehat{w}_i$ by $w_i(U_i)$ in functions G_l. As pointed out in [16], this means that we assume, on the one hand, that the cost to obtain a given level of quality, as given by G_l^b, does not depend on the level of demand in the market. This can be realistic in many cases, especially when $incr_i$ is not too high. On the other hand, it also implies that the location cost does not depend on the level of demand either. This is especially true if the cost of buying or renting the place for the location is paid in advance, before opening the new facility. In this way, the scenario which determines the cost of the location is not affected by the 'variation' in the demand produced by the location of the new facility, but just by the average demand.

As shown in [16], the objective function of the follower's problem with fixed demand is multimodal, but it tends to be smoother than the objective function of the follower's problem with variable demand, which has many more local optima and whose landscape is much steeper. Of course, the complexity of the centroid problem is more greatly affected due to the variable demand assumption.

2.3 Examples

In order to show the difficulty of the problem at hand, and its differences with the fixed demand case, in this section, we solve a quasi-real example dealing with the location of supermarkets in an area around the city of Murcia, in the southeast of Spain. In particular, a working radius of 25 km around the city of Murcia was considered. In all, 632,558 people live in that area, and they form our set of customers. Although they are distributed over 71 population centers, with populations varying between 1,138 and 178,013, in this example, we have considered an aggregated version, in which only population centers with a city hall are taken into account. The 21 towns with a city hall form our reduced set of demand points, with the population obtained by aggregating all population centres in the town which they administratively depend on. The mean purchasing power of a town was considered proportional to its population. The position and population of the towns can be seen in Fig. 1, where grey circles represent the forbidden areas around the existing demand points, which are at the center of those circles (the greater the circle, the greater the purchasing power at the demand point). The location space $S_1 = S_2$ was taken as the smallest rectangle containing all demand points.

There are five supermarkets in the area: three from a first chain, 'E', and two from another chain, 'C'. Figure 1 shows the location of each supermarket as viewed from chain E's point of view: firms belonging to chain E are marked by a black triangle, and firms from the other chain are shown by a black square on the map. We set the quality parameters $a_{i,j}$ within the interval [3,4]. The optimization of quality for the new facilities was carried out in the interval $[\alpha_l^{\min}, \alpha_l^{\max}] = [0.5, 5], l = 1, 2$. The income per unit of goods sold has been set to $s_l = 32,\ l = 1, 2$. Due to the lack of real data from the chains (they consider those data sensitive for them and are not willing to make them public), the other parameters have been validated in an ad hoc way to obtain 'reasonable' results. The interested reader is referred to [19] for more details about the case study and the value of the parameters.

In Fig. 1a, we can see the optimal location and quality for the new leader's facility (represented by $*$) and the new follower's facility (represented by $+$), when chain E is the leader, assuming that the demand is fixed (as obtained by algorithm UEGO_cent.SASS [13]). The corresponding solutions when the demand is variable are shown in Fig. 1b (as obtained using TLUEGO_UE, see Sect. 4.3). In those figures, we have added two windows on the right and bottom of the map, allowing us to view all three 2-dimensional projections of the 3-dimensional solution set: the map itself shows the 2-dimensional spatial part, without the quality, the right pane shows the quality and vertical space part (quality increases from left (0.5) to right (5)), the bottom pane shows the quality and horizontal space part (quality increases from top (0.5) to bottom (5)). The numerical results are shown in Table 1.

As we can see, in the fixed demand case, the optimal location for the leader is near the city of Alcantarilla, with a quality of 0.5. At that point, the market share captured by the new leader's facility is $m_1 = 2.112$, which is 5.94 % of the total market share. Considering all its facilities, chain E gets 53.22 % of the market, and a profit $\Pi_1 = 593.352$. The location for the follower's facility is near the city of Molina, with a quality of 3.696, where it captures

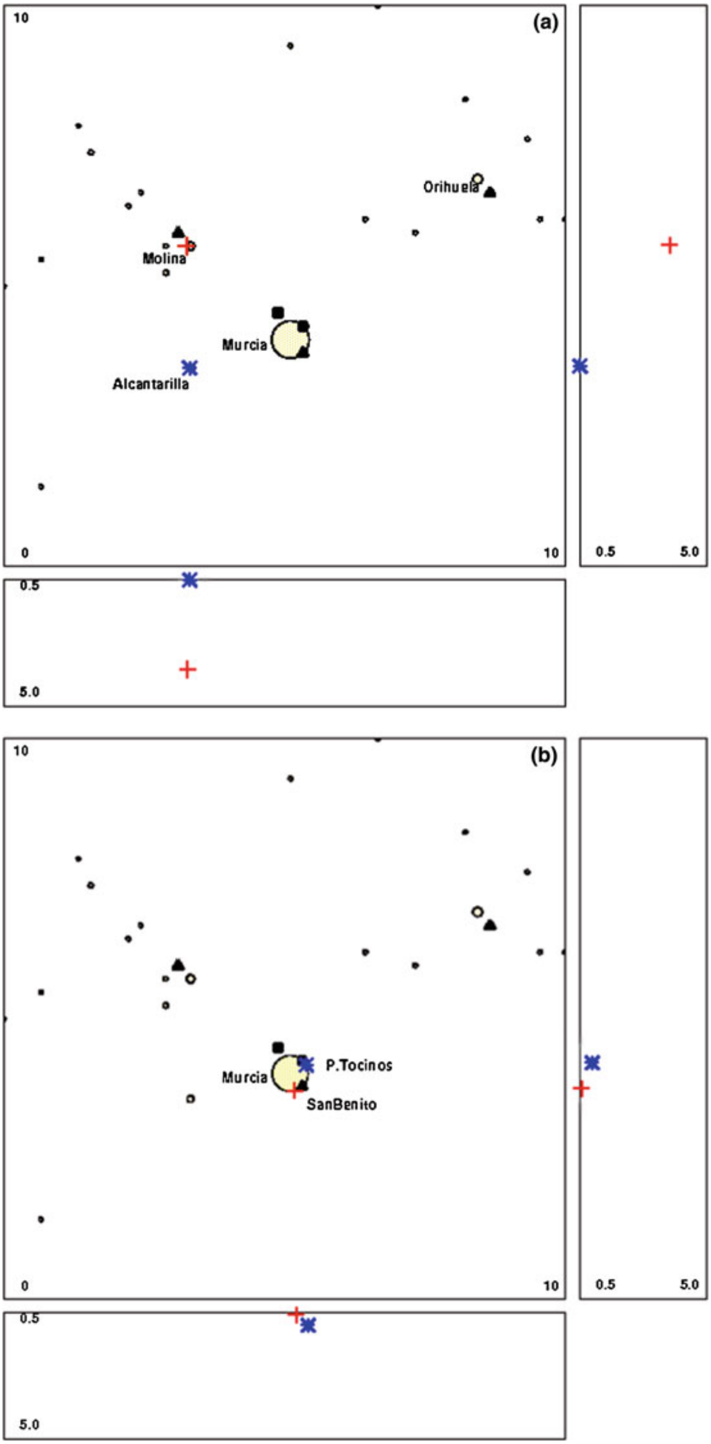

Fig. 1 Optimal location and quality for both leader and follower when chain E is the leader. **a** Fixed demand. **b** Variable demand. Leader's facility *star* (*blue*) and follower's facility *plus sign* (*red*). (Color figure online)

Table 1 Examples

Demand	nf_1	M_1	m_1	Π_1	nf_2	M_2	m_2	Π_2
Leader: chain E								
Fixed	(3.303, 6.433, 0.500)	18.915	2.112	593.352	(3.259, 4.285, 3.696)	16.625	7.123	461.776
Variable	(5.407, 5.798, 0.961)	2.807	0.419	73.454	(5.190, 6.276, 0.571)	3.618	0.249	101.563
Leader: chain C								
Fixed	(8.487, 3.026, 3.277)	15.961	6.247	442.122	(3.274, 6.441, 0.500)	19.579	2.187	614.652
Variable	(5.368, 6.166, 1.042)	3.822	0.453	106.320	(5.298, 6.228, 0.571)	2.6378	0.2489	70.227

20.04 % of the total market share. However, the leader's optimal location in the variable demand case is in the suburb of Puente Tocinos, in Murcia city, with a quality of 0.961. The market share captured by the facility is 0.419, which is 5.94 % of the total market share. The whole chain gets 43.68 % of the market and a profit $\Pi_1 = 73.454$. The location for the follower's facility is near the suburb of San Benito, in Murcia city, with a quality of 0.571, where it captures 3.875 % of the total market share.

If we now assume that chain C is the leader, then, in the fixed demand case, the optimal location for the leader is near the city of Orihuela, with a quality of 3.277, where the facility captures 17.57% of the total market share. The location for the follower's facility is near the city of Alcantarilla, with a quality of 0.5, where it captures 6.15 % of the total market share. The corresponding total market share captured by the chains and their profits can be seen in Table 1. However, the leader's optimal location in the variable demand case is near the suburb of San Benito, in Murcia city, with a quality of 1.042, and the location for the follower's facility is near the suburb of San Benito too, with a quality of 0.571.

These two examples show how important it is to take variable demand into consideration. As can be seen, the maximum profit for the chain is obtained at different locations and with different qualities, depending on whether variable (elastic) demand or fixed (inelastic) demand is considered. Also, the percentage of market share captured by the chains may change to the point that the chain getting more profit may be the opposite one.

3 Solving the medianoid problem

The medianoid problem associated with our centroid problem was studied in [16]. In that work, an exact interval branch-and-bound method (iB&B) and an evolutionary algorithm (UEGO), were proposed to deal with the problem.

The iB&B method considered in that work is described in [21]. Such a method is based on *Interval Analysis*. It uses boxes to define the search region and its branches, and *inclusion functions* to bound the objective function over a given box. Furthermore, it includes several new accelerating devices in order to solve difficult, highly nonlinear problems more efficiently. As a result, it determines an enclosure of all the globally optimal solutions within a pre-specified precision (for the medianoid problem with variable demand, a tolerance $\epsilon = 0.0001$ was considered in [16]). For practical purposes, iB&B is competitive for small sized problems. However, the computational time needed to solve a problem increases by a factor of 5 as the size of the problem increases by a factor of 2. The memory requirements increase accordingly. In fact, in [16], iB&B could only manage instances with, at most, 200 demand points.

UEGO is a general evolutionary algorithm designed to solve many kinds of multimodal global optimization problems (see [17] for a detailed description of the UEGO algorithm). It promotes the formation and maintenance of subpopulations (or individuals). The maximum number of subpopulations is given by the input parameter M (maximum population size). In this scenario, a subpopulation is a sphere defined by its center and a radius. The center is a solution, and the radius indicates its attraction area, which covers a region of the search space and hence, multiple solutions. The radius of the subpopulations is neither constant along the execution of UEGO nor the same for each subpopulation. This radius is a monotonous function that decreases as the index level (or cycles or generations) increases. The parameter L indicates the maximum number of levels in the algorithm. The radius of a subpopulation created at level i (with $i \in [1, L]$), is given by a decreasing exponential function which depends on the initial domain landscape (the radius at the first level, r_1) and the radius of the smallest subpopulation, r_L. Besides M, L and r_L, UEGO has another input parameter, N, which refers to the maximum number of function evaluations allowed for the whole optimization process. However, it is important to mention that UEGO may terminate simply because it has executed all of its levels. The final number of function evaluations depends on the complexity of the problem.

UEGO could also be identified as a memetic algorithm [22] in the sense that it uses local optimization in the evolution process. UEGO performs a local maximization on each subpopulation at every generation, and the local maxima replace the caller individuals. For dealing with the medianoid problem, a Weiszfeld-like algorithm was used in [16]. We will call it WLMv in what follows. The WLMv algorithm is a simple steepest-descent type method which takes discrete steps along the search paths. The method sets the derivatives of the objective function to zero and the next iteration is obtained by implicitly solving the resulting equations. The method is stopped when either two consecutive iterations are closer than a given tolerance ($\epsilon = 0.0001$), or the maximum number of iterations is reached ($r_{\max} = 400$). This local optimizer allows the evolutionary algorithm UEGO to find the global optimum with reliability. In what follows, we will refer to UEGO_medv as the algorithm UEGO executed with WLMv for solving the medianoid problems. In [16] it was found that a good parameter setting for UEGO_medv was $N = 10^6$, $M = 350$, $L = 30$ and $r_L = 0.05$.

UEGO_medv proved to be reliable when solving the medianoid problem with elastic demand in [16]. It always found the optimal solution in all the problems with up to $n = 200$ demand points, for which the optimal solution was known. Furthermore, it was faster than the interval B&B method (it needs, in average, from 28.43 % less time when $n = 50$–80.32 % when $n = 200$), and it had much less memory requirements than iB&B.

It is notheworthy that many medianoid instances have to be solved when dealing with a single centroid problem. Then, a trade-off between guarantee in the quality of the final solution and required computing time has to be found. Looking for this equilibrium, both alternatives, i.e. iB&B and UEGO_medv, are considered and analyzed when solving the medianoid problems.

4 Solving the centroid problem

In this section, three heuristics devised to cope with the centroid problem are described. More precisely, a grid search procedure, a multistart method named MSH, and an evolutionary algorithm called TLUEGO, are presented. In the last two, a local optimizer is needed. A subsection will be dedicated to briefly explain such a local technique. Two variants have been designed for the local optimizer, which derive two versions for MSH and TLUEGO algorithms.

4.1 GS: a grid search procedure

The first method is a simple Grid Search procedure (GS) as in [13]. A grid of points that cover the leader's 3-dimensional searching region is generated. For each point of the grid we first check its feasibility. If it is feasible, then we evaluate the objective function for the leader, which implies that we first have to solve the corresponding medianoid problem to obtain an optimal solution for the follower. To this aim, the algorithm UEGO_medv is used. When all the feasible points of the grid have been evaluated, a second finer grid is constructed in the vicinity of the point of the first grid having the best objective value. In our first grid, the length of the step between two adjacent points was 0.1 unit in each coordinate, and in the second grid, 0.02 unit.

4.2 The local optimizer SASS+WLMv

As mentioned above, to solve the centroid problem, both the multistart and the evolutionary algorithms make use of a local procedure. Local optimizers usually assume that the configuration of the problem during the optimization process does not change. However, this is not the case for the centroid problem, since every time the leader's facility changes, so does the follower's facility. Thus, the value of the objective function of the leader's problem may change if the new configuration is taken into account. This means that the new follower's facility should be computed every time the leader's facility changes. However, since the number of function evaluations in any local optimizer is usually large, obtaining the exact new follower's facility at each new location of the leader's facility will make the process very time-consuming.

To deal with our counterpart, the centroid problem with fixed demand, a local procedure called SASS+WLM was introduced in [13]. The idea of such an algorithm is to apply the stochastic hill climber SASS (see [23]) to improve the leader's facility, and a Weiszfeld-like algorithm WLM to approximate the follower's. The leader optimization is focused on a sphere whose radius is determined by the input parameter σ_{ub}. The algorithm stops when a maximum number of iterations ($ic_{\max}$) is reached, or when a maximum number of consecutive failures at improving the objective function ($Maxfcnt$) occurs.

For the problem at hand, and after trying different strategies, a local procedure similar to SASS+WLM in [13] is proposed. The pseudocode of this new method is given in Algorithm 1. The main differences between the local algorithm used in this paper (that we will call SASS+WLMv) and the one in [13] are:

- The Weiszfeld-like algorithm used now for updating the follower's facility is WLMv (described in [16]), instead of WLM. Similar to what was considered for UEGO_medv (see Sect. 3), WLMv stops when either two consecutive iterations are closer than the tolerance $\epsilon = 0.0001$, or when a maximum number of $r_{\max} = 400$ iterations is reached.
- The WLMv algorithm is not as reliable as the corresponding method WLM for the fixed demand case. Then, a large maximum number of iterations $ic_{\max}$ in SASS could direct the leader towards overestimated solutions. To deal with this drawback, the parameter $ic_{\max}$ in SASS+WLMv is reduced to 15. Additionally, once the maximum number of iterations $ic_{\max}$ is reached, the medianoid problem is solved *optimally*. Otherwise the objective value for the leader could be completely wrong, overestimated. This can even happen if the solutions are very close to optimality in objective function value but are in significantly different locations, and even if the leader's problem is solved optimally given the non-optimal follower's solution. For the centroid problem with fixed demand, the medianoid problem was computed with UEGO_med so as to have a reliable approximation of the follower's facility. For the problem at hand, two alternatives have been taken into account to do so: iB&B or UEGO_medv, giving place to two versions of the local optimizer.

Notice that the algorithm iB&B gives as a solution a list of small 3-dimensional intervals where any optimizer point must lie. Then, when selecting this method in Step 9 of Algorithm 1 the solution nf_2^{opt} considered will be the best point evaluated by the algorithm iB&B.

Algorithm 1 Algorithm SASS+WLMv($nf_1, nf_2, ic_{\max}(=15), \sigma_{ub}$)

1: Initialize SASS parameters. Set $ic = 1, nf_1^{opt} = nf_1, \Pi_1^{opt} = \Pi_1(nf_1, nf_2)$.
2: **while** $ic \le ic_{\max}$
3: Update SASS parameters considering the previous successes at improving the objective function value of the leader.
4: Generate a location for the leader $nf_1^{(ic)}$ within the updated radius.
5: Solve the corresponding medianoid problem using WLMv and let $nf_2^{(ic)}$ denote the solution obtained.
6: **if** $\Pi_1(nf_1^{(ic)}, nf_2^{(ic)}) > \Pi_1^{opt}$
7: set $nf_1^{opt} = nf_1^{(ic)}$ and $\Pi_1^{opt} = \Pi_1(nf_1^{(ic)}, nf_2^{(ic)})$.
8: $ic = ic + 1$.
9: Compute the corresponding follower nf_2^{opt} for nf_1^{opt} using either iB&B or UEGO_medv.
10: **if** $\Pi_1(nf_1^{opt}, nf_2^{opt}) > \Pi_1(nf_1, nf_2)$
11: return (nf_1^{opt}, nf_2^{opt})
12: **else**
13: Return (nf_1, nf_2).

4.3 TLUEGO: a two-level evolutionary global optimization algorithm

The more robust algorithm designed to cope with the centroid problem has been the evolutionary algorithm called TLUEGO. This algorithm is similar to the algorithm UEGO_cent.SASS introduced in [13], which deals with the corresponding centroid problem with fixed demand. TLUEGO, as well as UEGO_cent.SASS, shares some concepts and ideas with UEGO_medv (see Sect. 3). In particular, the concept of subpopulation (including attraction radius), the use of a local optimizer and the set of input paramaters have been either adopted or adapted to cope with the centroid problems. The values of the input parameters used for UEGO_medv (see Sect. 3) have also been adopted for TLUEGO.

In the following, the general structure of TLUEGO is provided (see Algorithm 2). At the beginning, a single subpopulation (the root) exists, and as the algorithm evolves and applies genetic operators, new subpopulations can be created. For TLUEGO to work properly, it is very important to correctly evaluate the fitness of the new subpopulations after the creation procedure. To this aim, a reliable follower solution has to be computed, and to do so, two alternative algorithms are possible: iB&B or UEGO_medv, as was suggested in Sect. 3. At every generation, TLUEGO performs a local optimizer operation on each subpopulation. For the problem at hand, the algorithm SASS+WLMv is used. Notice that it is executed twice in order to have more chances for obtaining a better point. The value of σ_{ub} passed to SASS+WLMv is always (the two times it is called) the radius associated to the calling subpopulation. In this way, the scope of the local optimizer is exactly the area covered by the subpopulation. Notice that a subpopulation involves a 'cooling' technique which enables the search to focus on the promising regions of the space, starting off with a relatively large radius that decreases as the search proceeds. Then, exploration and exploitation of the search space are guaranteed. TLUEGO has been executed with the two variants of the local optimizer, i.e. considering iB&B and UEGO_medv when computing a reliable solution for the follower (Step 9 in Algorithm 1). It is important to highlight that TLUEGO performs two selection procedures during the optimization process. The first one is carried out after the new offspring is generated. It consists of the 'Fuse subpopulations' and the 'Shorten subpopulation list' procedures. The second one takes place after the optimization procedure, and only considers the Fuse subpopulations procedure. The reader is referred to [13] for a more detailed description of these procedures.

The inclusion of iB&B or UEGO_medv in TLUEGO derives two algorithms for solving the centroid problem, TLUEGO_BB and TLUEGO_UE, respectively.

Algorithm 2 Algorithm TLUEGO$(N,\ M,\ L,\ r_L)$

Set iteration counter $i = 1$.
Initialize a random leader location (center of initial subpopulation) $nf_1^{(i)}$ and compute the corresponding follower $nf_2^{(i)}$ using either iB&B or UEGO_medv.
$(nf_1^{(i)*}, nf_2^{(i)*}) = \text{SASS} + \text{WLMv}(nf_1^{(i)}, nf_2^{(i)}, ic_{\max}(=15), \sigma_{ub}(=r_i))$.
$(nf_1^{opt}, nf_2^{opt}) = \text{SASS} + \text{WLMv}(nf_1^{(i)*}, nf_2^{(i)*}, ic_{\max}(=15), \sigma_{ub}(=r_i))$.
for $i = 2$ until L
  Create new subpopulations
  Compute the corresponding follower for the new subpopulations using either iB&B or UEGO_medv, and evaluate the leaders' fitness values.
  Fuse subpopulations, and Shorten the subpopulation list.
  for each existing subpopulation $nf_1^{(i)}$ (with radius r_i) and its corresponding follower $nf_2^{(i)}$
    $(nf_1^{(i)*}, nf_2^{(i)*}) = \text{SASS} + \text{WLMv}(nf_1^{(i)}, nf_2^{(i)}, ic_{\max}(=15), \sigma_{ub}(=r_i))$.
    $(nf_1^{opt}, nf_2^{opt}) = \text{SASS} + \text{WLMv}(nf_1^{(i)*}, nf_2^{(i)*}, ic_{\max}(=15), \sigma_{ub} = r_i)$.
  Fuse subpopulations.
  $i = i + 1$.
Return the best leader facility and its objective value.

4.4 MSH: a multistart heuristic algorithm

The MSH algorithm consists of randomly generating *MaxStartPoints* feasible candidate solutions for the leader and applying a local optimizer to them in order to achieve an optimized leader solution. The final solution will be the one with the best objective function value.

For the case at hand, the considered local optimizer has been SASS+WLMv (see Algorithm 1). Notice that this method focuses the search on an area defined by the parameter σ_{ub}. In order to provide a balance between exploitation and exploration of the search space, this method has also been executed twice as in TLUEGO, but with different values for σ_{ub}. In the first call, a value of $\sigma_{ub} = 2.083895$ (the one corresponding to level 10 in TLUEGO) was considered. Such a value was chosen because in this way, the initial random candidate solutions in the multistart strategy can cover the whole searching space, and at the same time, they can focus on an area small enough so that the local procedure can find a good local optimum. In the second call, a value of $\sigma_{ub} = 0.162375$ (the one corresponding to level 23 in TLUEGO) was used, to improve the quality of the local optimum obtained with the first call. These σ_{ub} values were selected after some preliminary studies, in which eight problems of different sizes were solved trying different strategies for the heuristic algorithm.

As in TLUEGO, two versions of the MSH method, called MSH_BB and MSH_UE, are proposed. These differ in whether iB&B or UEGO_medv is used as a method of computing the follower nf_2^{opt} in Step 9 of the local optimizer SASS+WLMv (see Algorithm 1).

5 Computational studies

All the computational results in this paper have been carried out on a processor Xeon IV with 2.4GHz and 1 GByte RAM. The algorithms have been implemented in C++.

To study the performance of the algorithms, 24 different problems have been generated varying the number n of demand points, the number m of existing facilities and the number k

Table 2 Settings of the test problems

n	15			25			50		
m	2	5	10	2	5	10	2	5	10
k	0,1	0,1,2	0,2,4	0,1	0,1,2	0,2,4	0,1	0,1,2	0,2,4

of those facilities belonging to the leader's chain. The actual settings (n, m, k) employed are detailed in Table 2. For every setting the problem was generated by randomly choosing its parameters uniformly within pre-specified intervals as in [16]. In all the problems we have set $S_1 = S_2 = ([0, 10], [0, 10])$ and $\alpha_1, \alpha_2 \in [0.5, 5]$

Since most of the proposed algorithms are heuristics, each run may provide a different solution. Thus, to study their robustness, for every heuristic algorithm, each problem has been solved ten times and average values have been computed. Nevertheless, the heuristic GS has been run only once and the results obtained in that run (no average results) are given.

Tables 3, 4 and 5 show the results obtained by the algorithms when $n = 15$, $n = 25$ and $n = 50$, respectively. In the column labelled 'Av(T)', the average time in the ten runs (in seconds) is given; in the 'BestSol' column, the best solution (x_1, y_1, α_1) found in the ten runs is shown; the 'MaxDist' column gives the maximum Euclidean distance (considering the three variables of the problem) between any pair of solutions provided by the algorithm, which gives an idea of how far the solutions provided by the algorithm in different runs can be; in the next three columns, the minimum, the average and the maximum objective value in the ten runs are given. Finally, in the 'Dev' column, the standard deviation is provided. As can be seen in these tables, two versions of TLUEGO and MSH algorithms have been executed. It is worth mentioning that the number of times that MSH_BB (resp. MSH_UE) was allowed to repeat its basic local optimizer was chosen so that the CPU time employed by MSH_BB (resp. MSH_UE) was, on average (when considering all the problems with the same value of n), similar to the CPU time employed by TLUEGO_BB (resp. TLUEGO_UE) or a bit higher. In particular, for the problems with 15, 25 and 50 demands points, the number of starting points were 150, 200 and 250, respectively. On the bottom of these tables, we also show the average results for each algorithm.

The method considered to solve the medianoid problem does not seem to have too much influence on the quality of the final solution, i.e., TLUEGO and MSH behave similarly independent of whether iB&B or UEGO_medv is employed. This fact is corroborated by [16], where it was stated that UEGO_medv was able to obtain the global optimal solution for all the problems where iB&B could be executed (with $n \leq 200$). However, the computing time is highly affected by those methods. The iB&B technique is faster than UEGO_medv for small size problems, which helps to reduce the execution time of both TLUEGO and MSH. Namely, the use of iB&B reduces by 74.6 % the computing time of TLUEGO for problems with $n = 15$ (as compared to its counterpart executed with UEGO_medv). The corresponding reduction for the problems with $n = 25$ is 30.9 %. Similar reductions in computing time can be seen in MSH when iB&B is used instead of UEGO_medv. Nevertheless, for medium size problems (with $n = 50$ demand points), TLUEGO_UE and MSH_UE reduce the computing time as compared to TLUEGO_BB and MSH_BB, by 12.79 and 10.63 %, respectively. These results are also consistent with the ones showed in [16], where it was observed that the increase of requirements for iB&B with the size of the problem was greater than for UEGO_medv.

Focusing now on the strategies proposed to solve the current centroid problem, it can be stated that TLUEGO (in any of its versions) is the algorithm providing the best results. Its

Table 3 Results for the problems with $n = 15$

(n, m)	Algorithm	Av (T)	BestSol			Max	Objective function			
		(s)	x_1	y_1	α_1	Dist	Min	Av	Max	Dev
(2,0)	TLUEGO_BB	618	8.505	4.154	0.50	0.049	−4.630	−4.630	−4.630	0.000
	TLUEGO_UE	1,217	8.513	4.152	0.50	0.032	−4.630	−4.630	−4.630	0.000
	MSH_BB	619	8.558	4.138	0.50	0.258	−4.678	−4.653	−4.635	0.016
	MSH_UE	1,254	8.566	4.133	0.50	0.235	−4.744	−4.663	−4.640	0.041
	GS	379,932	8.540	4.140	0.50	–	–	−4.637	–	–
(2,1)	TLUEGO_BB	121	7.860	7.841	0.50	0.000	38.732	38.732	38.732	0.000
	TLUEGO_UE	977	7.860	7.841	0.50	0.001	38.731	38.732	38.732	0.001
	MSH_BB	142	7.865	7.841	0.50	0.111	38.607	38.667	38.727	0.043
	MSH_UE	1,247	7.876	7.837	0.50	0.276	38.468	38.608	38.698	0.097
	GS	400,226	7.860	7.840	0.50	–	–	38.730	–	–
(5,0)	TLUEGO_BB	264	5.731	8.062	0.88	0.058	−5.806	−5.805	−5.802	0.002
	TLUEGO_UE	1,039	5.731	8.062	0.85	0.017	−5.804	−5.803	−5.802	0.001
	MSH_BB	314	5.773	8.078	0.81	0.525	−6.394	−6.115	−5.978	0.148
	MSH_UE	1,188	5.729	8.064	0.91	3.315	−6.713	−6.095	−5.815	0.328
	GS	481,907	5.720	8.080	0.75	–	–	−5.928	–	–
(5,1)	TLUEGO_BB	393	1.328	0.000	0.50	0.007	10.574	10.575	10.575	0.000
	TLUEGO_UE	1,156	1.328	0.000	0.50	0.019	10.573	10.574	10.575	0.001
	MSH_BB	376	1.325	0.036	0.50	8.029	10.560	10.564	10.571	0.005
	MSH_UE	1,159	1.328	0.004	0.50	0.036	10.571	10.573	10.574	0.001
	GS	511,797	9.120	0.180	0.50	–	–	10.533	–	–
(5,2)	TLUEGO_BB	71	5.711	2.343	0.50	0.000	39.968	39.968	39.968	0.000
	TLUEGO_UE	691	5.714	2.341	0.50	0.003	39.968	39.969	39.970	0.001
	MSH_BB	109	5.711	2.345	0.50	0.162	39.852	39.910	39.965	0.042
	MSH_UE	1,058	5.714	2.342	0.50	0.116	39.883	39.930	39.964	0.031
	GS	391,193	5.720	2.340	0.50	–	–	39.952	–	–
(10,0)	TLUEGO_BB	140	0.000	1.854	0.50	0.000	−9.753	−9.753	−9.753	0.000
	TLUEGO_UE	726	0.000	1.854	0.50	0.001	−9.754	−9.753	−9.753	0.000
	MSH_BB	196	0.000	1.855	0.50	0.021	−9.758	−9.757	−9.756	0.001
	MSH_UE	984	0.000	1.854	0.50	0.017	−9.764	−9.758	−9.753	0.004
	GS	621,003	0.120	1.880	0.50	–	–	−9.789	–	–
(10,2)	TLUEGO_BB	139	7.206	10.000	0.50	0.003	16.414	16.415	16.415	0.000
	TLUEGO_UE	709	7.206	10.000	0.50	0.000	16.415	16.415	16.415	0.000
	MSH_BB	198	7.206	9.995	0.50	0.080	16.392	16.405	16.411	0.007
	MSH_UE	915	7.206	10.000	0.50	0.009	16.402	16.411	16.415	0.005
	GS	573,090	7.200	10.000	0.50	–	–	16.395	–	–
(10,4)	TLUEGO_BB	59	2.273	0.487	0.50	0.000	38.323	38.323	38.323	0.000
	TLUEGO_UE	613	2.273	0.487	0.50	0.000	38.323	38.323	38.323	0.000
	MSH_BB	110	2.271	0.486	0.50	0.129	38.220	38.282	38.320	0.042
	MSH_UE	921	2.260	0.483	0.50	0.121	38.218	38.263	38.307	0.032
	GS	563,554	2.260	0.480	0.50	–	–	38.304	–	–
Aver.	TLUEGO_BB	226	–	–	–	0.015	15.478	15.478	15.479	0.000

Table 3 continued

(n, m)	Algorithm	Av (T) (s)	BestSol x_1	y_1	α_1	Max Dist	Objective function Min	Av	Max	Dev
	TLUEGO_UE	891	–	–	–	0.009	15.478	15.478	15.479	0.001
	MSH_BB	258	–	–	–	1.164	15.350	15.413	15.453	0.038
	MSH_UE	1, 091	–	–	–	0.516	15.290	15.409	15.469	0.067
	GS	490,338	–	–	–	–	–	15.445	–	–

TLUEGO_BB ($epsilon = 0.0001$), TLUEGO_UE, MSH_BB and MSH_UE (with 150 starting points), and GS

Table 4 Results for the problems with $n = 25$

(n, m)	Algorithm	Av (T) (s)	BestSol x_1	y_1	α_1	Max Dist	Objective function Min	Av	Max	Dev
(2,0)	TLUEGO_BB	2,714	4.580	6.183	4.99	0.016	45.533	45.593	45.637	0.042
	TLUEGO_UE	2,991	4.580	6.184	4.99	0.030	45.472	45.541	45.592	0.056
	MSH_BB	2,169	4.584	6.189	4.92	2.210	22.210	34.739	44.373	8.146
	MSH_UE	2,652	4.581	6.185	4.32	4.964	24.210	32.102	40.052	5.405
	GS	876,818	4.600	6.200	5.00	–	–	42.191	–	–
(2,1)	TLUEGO_BB	1,324	7.066	7.225	4.93	0.110	61.041	61.049	61.055	0.006
	TLUEGO_UE	1,957	7.066	7.225	4.89	0.105	61.044	61.045	61.050	0.002
	MSH_BB	1,619	7.229	7.414	4.63	0.781	35.451	39.567	51.214	5.876
	MSH_UE	2,630	7.072	7.231	4.78	1.148	49.804	55.859	60.568	3.566
	GS	745,640	7.080	7.240	4.75	–	–	59.912	–	–
(5,0)	TLUEGO_BB	915	0.000	2.159	0.50	0.002	−10.847	−10.847	−10.846	0.000
	TLUEGO_UE	1,579	0.000	2.159	0.50	0.002	−10.847	−10.847	−10.846	0.000
	MSH_BB	1,503	0.000	2.158	0.50	0.064	−10.882	−10.867	−10.849	0.012
	MSH_UE	2,163	0.002	2.158	0.50	0.028	−10.862	−10.853	−10.847	0.005
	GS	860,059	0.020	2.140	0.50	–	–	−10.881	–	-
(5,1)	TLUEGO_BB	1,704	10.000	8.657	0.50	0.000	13.492	13.492	13.492	0.000
	TLUEGO_UE	1,495	10.000	8.657	0.50	0.000	13.492	13.492	13.492	0.000
	MSH_BB	2,072	10.000	8.658	0.50	0.039	13.469	13.481	13.489	0.008
	MSH_UE	2,322	9.999	8.657	0.50	0.002	13.480	13.488	13.492	0.005
	GS	844,241	9.300	6.600	0.50	–	–	13.430	–	–
(5,2)	TLUEGO_BB	1,957	6.152	2.006	2.47	0.085	46.736	46.739	46.741	0.002
	TLUEGO_UE	2,251	6.152	2.006	2.51	0.521	46.502	46.666	46.732	0.085
	MSH_BB	2,387	6.146	2.000	2.47	0.801	44.630	45.947	46.620	0.701
	MSH_UE	2,793	6.149	2.010	2.73	0.671	45.249	45.911	46.389	0.542
	GS	762,229	6.140	2.000	2.25	–	–	46.388	–	–
(10,0)	TLUEGO_BB	425	0.526	0.000	0.50	0.001	−10.283	−10.283	−10.283	0.000
	TLUEGO_UE	2,135	0.526	0.000	0.50	0.003	−10.283	−10.283	−10.283	0.000
	MSH_BB	651	0.526	0.000	0.50	0.016	−10.289	−10.285	−10.283	0.002

Table 4 cotinued

(n, m)	Algorithm	Av (T)	BestSol			Max	Objective function			
		(s)	x_1	y_1	α_1	Dist	Min	Av	Max	Dev
	MSH_UE	3,650	0.526	0.000	0.50	0.001	−10.284	−10.283	−10.283	0.001
	GS	1,237,873	0.520	0.000	0.50	–	–	−10.295	–	–
(10,2)	TLUEGO_BB	627	9.623	9.436	0.50	0.001	24.958	24.959	24.959	0.000
	TLUEGO_UE	1,692	9.651	9.440	0.50	0.027	24.960	24.962	24.965	0.002
	MSH_BB	959	9.633	9.437	0.50	0.377	24.856	24.896	24.953	0.035
	MSH_UE	2,285	9.615	9.438	0.50	0.281	24.854	24.911	24.948	0.035
	GS	963,392	9.880	9.460	0.50	–	–	24.852	–	–
(10,4)	TLUEGO_BB	942	0.486	4.980	2.94	0.544	62.201	62.334	62.414	0.094
	TLUEGO_UE	1,259	0.486	4.980	3.32	0.702	62.169	62.273	62.402	0.074
	MSH_BB	1,449	0.481	4.982	3.31	2.302	57.799	59.881	62.081	1.794
	MSH_UE	2,153	0.483	4.977	2.78	3.298	51.366	56.898	62.045	4.034
	GS	1,021,566	0.480	4.980	2.75	–	–	61.836	–	–
Aver.	TLUEGO_BB	1326	–	–	–	0.095	29.104	29.130	29.146	0.018
	TLUEGO_UE	1,920	–	–	–	0.174	29.064	29.106	29.138	0.027
	MSH_BB	1,601	–	–	–	0.824	22.156	24.670	27.700	2.072
	MSH_UE	2,581	–	–	–	1.299	23.477	26.004	28.296	1.699
	GS	913,977	–	–	–	–	–	28.429	–	–

TLUEGO_BB ($epsilon = 0.0001$), TLUEGO_UE, MSH_BB and MSH_UE (with 200 starting points), and GS

average objective function values (see column 'Av' in Tables 3, 4 and 5) are always higher than the ones given by both MSH and GS. It is also important to highlight that the minimum objective function value found by TLUEGO in the ten runs is always better than the average values obtained by both MSH and GS (see columns 'Min' and 'Av').

In addition to this, TLUEGO is also the more reliable algorithm, in the sense that it usually attains the same solution in all the runs, whereas MSH is more erratic, and may provide different solutions in each run (see the values of 'MaxDist' and 'Dev').

MSH has been designed to check whether a random search is enough to find the global optimum of the centroid problem introduced in this paper. But as the results show, it is necessary to direct the search through the whole space, as TLUEGO does. In only four problems (those with settings (15, 10, 0), (15, 10, 2), (25, 5, 1) and (25, 10, 0)) does MSH find the same best solution as TLUEGO (see columns 'BestSol'). Furthermore, in none of the 24 problems the best objective function value found by MSH in the ten runs is better than the corresponding average value of TLUEGO (see columns 'Max' and 'Av'). Comparing MSH to GS, only in 10 out of the 24 problems is the average value of MSH greater than the objective value obtained by GS.

GS is rather time-consuming. Moreover, there is no guarantee that GS can find a good approximation to the global optimum. If the objective function value decreases dramatically in a small neighbourhood around the global optimum and the grid is not dense enough, the second finer grid can focus around a local optimum. Something similar can happen when a local optimum exists whose objective value is close to the global optimum value and the grid is not fine enough. The risk of failure is even higher in the presence of constraints, as happens in our centroid problem, since it may occur that the global optimum is surrounded (in part)

Table 5 Results for the problems with $n = 50$

(n, m)	Algorithm	Av (T)	BestSol			Max	Objective function			
		(s)	x_1	y_1	α_1	Dist	Min	Av	Max	Dev
(2,0)	TLUEGO_BB	14,473	5.936	5.663	2.44	0.138	−9.998	−9.986	−9.972	0.010
	TLUEGO_UE	5,557	5.936	5.663	2.45	0.222	−9.996	−9.985	−9.971	0.010
	MSH_BB	17,699	6.051	5.644	2.72	8.323	−12.840	−12.642	−11.859	0.391
	MSH_UE	7,242	5.939	5.661	2.23	8.203	−12.836	−11.499	−10.250	1.094
	GS	3,003,349	5.940	5.660	2.25	–	–	−10.276	–	–
(2,1)	TLUEGO_BB	8,632	2.840	4.738	4.96	0.051	77.850	78.006	78.297	0.152
	TLUEGO_UE	6,377	2.840	4.738	5.00	0.043	78.297	78.516	78.654	0.125
	MSH_BB	8,080	2.839	4.726	4.98	1.729	28.929	51.910	76.424	19.328
	MSH_UE	7,180	2.852	4.737	4.94	3.679	21.223	63.114	75.007	20.981
	GS	3,003,750	2.840	4.720	5.00	–	–	75.285	–	–
(5,0)	TLUEGO_BB	8,295	2.643	1.097	4.98	0.000	27.917	27.917	27.917	0.000
	TLUEGO_UE	9,483	2.645	1.096	4.98	0.005	27.917	28.127	28.267	0.172
	MSH_BB	9,433	3.711	2.168	4.99	1.791	12.304	17.132	21.203	2.929
	MSH_UE	11,058	2.640	1.099	4.99	2.337	6.922	15.253	27.402	7.428
	GS	3,003,816	3.720	2.160	4.50	–	–	21.678	–	–
(5,1)	TLUEGO_BB	10,993	7.346	9.574	4.39	0.740	16.347	16.455	16.614	0.090
	TLUEGO_UE	7,153	7.346	9.574	4.61	0.634	16.422	16.561	16.666	0.093
	MSH_BB	12,956	7.042	9.555	4.74	2.520	−0.745	6.598	13.342	4.769
	MSH_UE	9,112	7.265	9.583	4.13	1.952	0.359	8.730	12.275	4.303
	GS	3,003,513	7.340	9.580	4.00	–	–	15.153	–	–
(5,2)	TLUEGO_BB	11,834	9.502	4.853	2.34	0.441	48.553	48.664	48.776	0.096
	TLUEGO_UE	16,579	9.502	4.853	2.37	0.393	48.478	48.706	48.779	0.115
	MSH_BB	15,717	9.497	4.851	2.03	5.111	46.868	47.116	48.106	0.495
	MSH_UE	20,315	10.000	0.000	0.50	0.000	47.698	47.698	47.698	0.000
	GS	3,004,494	9.980	0.300	0.50	–	–	47.459	–	–
(10,0)	TLUEGO_BB	7,825	1.008	7.438	5.00	0.073	31.359	31.580	31.734	0.180
	TLUEGO_UE	6,510	1.008	7.438	5.00	0.099	31.223	31.524	31.734	0.204
	MSH_BB	9,336	0.999	7.428	4.98	1.490	18.745	23.831	29.913	3.574
	MSH_UE	7,495	2.206	8.129	4.93	1.145	23.221	26.986	30.718	2.655
	GS	3,003,878	1.020	7.420	5.00	–	–	28.702	–	–
(10,2)	TLUEGO_BB	4,826	9.865	8.238	4.98	0.032	56.181	56.315	56.414	0.109
	TLUEGO_UE	6,532	9.865	8.238	4.99	0.021	56.225	56.390	56.521	0.095
	MSH_BB	5,199	9.545	7.674	4.84	1.288	43.832	47.754	54.446	3.663
	MSH_UE	6,660	9.866	8.238	4.38	3.113	36.785	46.307	52.232	5.220
	GS	3,003,775	9.880	8.240	5.00	–	–	50.801	–	–
(10,4)	TLUEGO_BB	8,884	7.675	3.264	4.22	0.013	70.718	70.729	70.738	0.009
	TLUEGO_UE	7,880	7.675	3.264	4.24	0.061	70.730	70.737	70.741	0.004
	MSH_BB	10,296	7.669	3.265	4.17	0.585	67.679	68.935	70.198	0.911
	MSH_UE	10,223	7.661	3.255	3.92	1.727	66.780	68.119	69.594	1.090
	GS	3,003,775	7.460	3.040	4.00	–	–	69.442	–	–
Aver.	TLUEGO_BB	9,470	–	–	–	0.186	39.866	39.960	40.065	0.081

Table 5 continued

(n, m)	Algorithm	Av (T)	BestSol			Max	Objective function			
		(s)	x_1	y_1	α_1	Dist	Min	Av	Max	Dev
	TLUEGO_UE	8,259	–	–	–	0.185	39.912	40.072	40.174	0.102
	MSH_BB	11,090	–	–	–	2.855	25.597	31.329	37.722	4.508
	MSH_UE	9,911	–	–	–	2.769	23.769	33.088	38.084	5.346
	GS	3,003,794	–	–	–	–	–	37.280	–	–

TLUEGO_BB ($epsilon = 0.0001$), TLUEGO_UE, MSH_BB and MSH_UE (with 250 starting points), and GS

Table 6 Average results considering all the problems ($n = 15, 25, 50$)

Algorithm	Av (T)	Max	Objective function			
	(s)	Dist	Min	Av	Max	Dev
TLUEGO_BB	3, 674	0.099	28.149	28.189	28.230	0.033
TLUEGO_UE	3, 690	0.123	28.151	28.219	28.264	0.043
MSH_BB	4, 316	1.614	21.034	23.804	26.958	2.206
MSH_UE	4, 528	1.528	20.845	24.834	27.283	2.371
GS	1,469,370	–	–	27.052	–	–

by infeasible areas, and the grid may not have a feasible point near the global optimum. Of course, the finer the grids, the higher the possibilities for the method to find the optimum, but one never knows how small the distance between two adjacent points in the grid should be, and regardless how small that distance is, it may still happen that the search does not reach the global optimum. In fact, we have used GS only as a safeguard to check the goodness of TLUEGO and MSH, and also because it allows us to study the difficulty of the problem at hand and to draw the graphs of the objective function projected in both the location and the quality spaces.

Finally, to have an overall view of the algorithms' behaviour, average values when considering all the problems are presented in Table 6. As can be observed, similar conclusions can be inferred, i.e. TLUEGO is both the algorithm giving the best and most robust results, and this using the least computational time. On average, MSH provides the worst objective function value (see column Av) and different runs may provide very different objective values. GS is much more time-consuming and is not able to find the global optimum. Regarding the quality of the solutions, it seems that TLUEGO_UE and MSH_UE obtain better average results than their corresponding counterparts executed with iB&B.

To illustrate the algorithms' behaviour, the best solution obtained by the different algorithms for the problem with setting (50, 5, 0) are depicted in Fig. 2, projected onto the two-dimensional spaces (see also Table 7). In that figure, the black squares (■) correspond to the locations of the existing follower's facilities, the green symbol + gives the best solution found by TLUEGO_BB in the ten runs, and the blue star ✳ the one obtained by TLUEGO_UE; the green and red signs × give the best solutions provided by MSH_BB and MSH_UE, respectively. Finally, the red plus sign + represents the solution obtained by GS algorithm. Light yellow ovals represent the forbidden areas around the existing demand points, which are at the centre of those ovals (the greater the oval, the greater the purchasing power at the demand

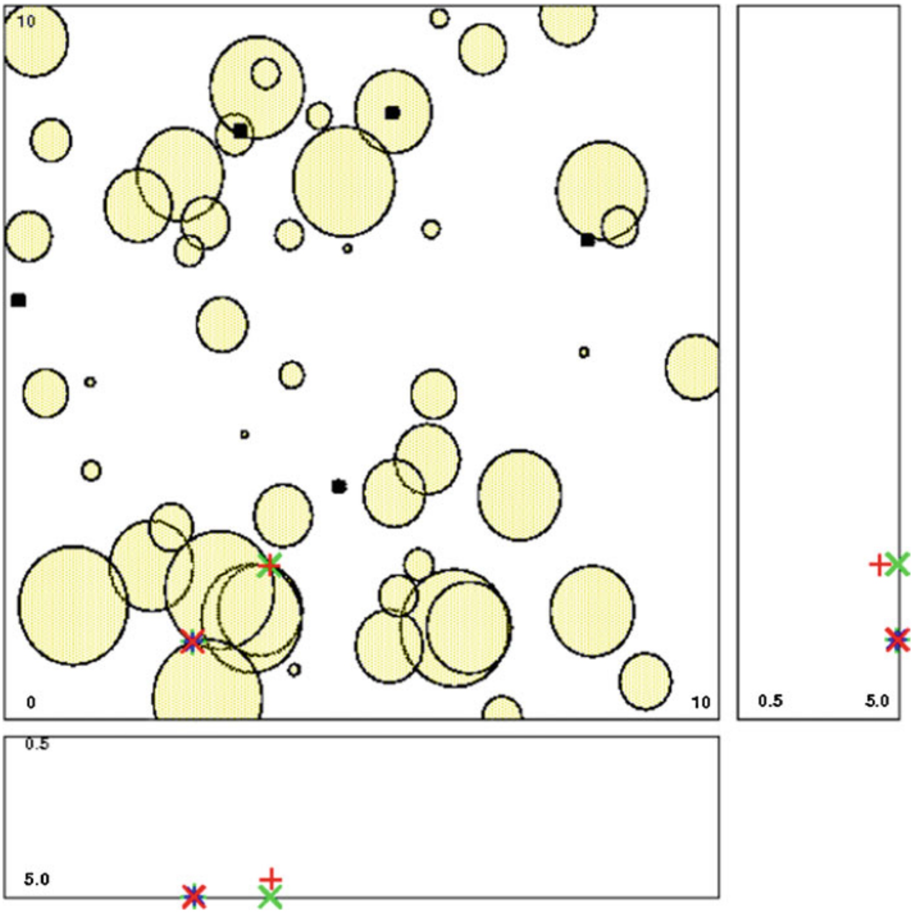

Fig. 2 Example with $n = 50, m = 5, k = 0$. TLUEGO_BB = *plus sign* (*green*), TLUEGO_UEGO = *star* (*blue*), MSH_BB = *multiple sign* (*green*) MSH_UEGO = *multiple sign* (*red*) and GS = *plus sign* (*red*). (Color figure online)

Table 7 Results for the problem with setting (50, 5, 0)

Algorithm	Av (T)	BestSol			Max	Objective function			
	(s)	x_1	y_1	α_1	Dist	Min	Av	Max	Dev
TLUEGO_BB	8,295	2.643	1.097	4.98	0.000	27.917	27.917	27.917	0.000
TLUEGO_UEGO	9,483	2.645	1.096	4.98	0.005	27.917	28.127	28.267	0.172
MSH_BB	9,433	3.711	2.168	4.99	1.791	12.304	17.132	21.203	2.929
MSH_UEGO	11,058	2.640	1.099	4.99	2.337	6.922	15.253	27.402	7.428
GS	3,003,816	3.720	2.160	4.50	–	–	21.678	–	–

point). Notice that in this problem $k = 0$, i.e., there are no existing facilities belonging to the leader's chain.

As can be seen, in this example the optimal solution is at the intersection of two forbidden regions, as found by both TLUEGO_BB and TLUEGO_UE. The solution provided by MSH_UE is around that area, but it focuses on a local maximum whose objective value is close to the optimal one (see 'Max' column in Table 7). Also notice that the solutions provided by GS and MSH_BB are quite close, but they yield very different objective values, showing that the objective function can locally be quite steep. Thus, this example clearly shows that GS is not a good strategy, since it does not allow a proper approximation to the optimal point. Additionally, MSH (in any of its versions) may get trapped in a local optimum, since the goodness of that algorithm depends on how close the starting points are with respect to the optima.

6 Conclusions

In this study a new (1|1)-centroid problem on the plane with variable demand has been introduced. Three heuristics have been proposed for handling the problem, namely, a grid search procedure, a multistart method and an evolutionary algorithm. The computational studies have shown that the evolutionary algorithm TLUEGO provides the best results and is more robust than the other strategies. However, the computational time employed by TLUEGO for solving a problem with 50 demand points is in average more than 2.5 hours. This clearly suggests that a parallelization of the algorithm is needed, especially if real problems, with more demand points, are to be solved.

Acknowledgments This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117, ECO2008-00667/ECON, ECO2011-24927), Junta de Andalucía (P08-TIC-3518, P10-TIC-6002, P11-TIC7176) and Fundación Séneca (The Agency of Science and Technology of the Region of Murcia, 00003/CS/10 and 15254/PI/10), in part financed by the European Regional Development Fund (ERDF). Juana López Redondo is a fellow of the Spanish 'Juan de la Cierva' contract program. We thank the 'Centro de Supercomputación' of the 'Fundación Parque Científico de Murcia' for having allowed us to use its facilities and for its technical support.

References

1. Drezner, Z.: Facility Location: A Survey of Applications and Methods. Springer Series in Operations Research and Financial Engineering, Berlin (1995)
2. Drezner, Z., Hamacher, H.W.: Facility location. Applications and Theory. Springer, Berlin (2002)
3. Eiselt, H.A., Laporte, G., Thisse, J.F.: Competitive location models: a framework and bibliography. Transport. Sci. **27**(1), 44–54 (1993)
4. Kilkenny, M., Thisse, J.F.: Economics of location: a selective survey. Comput. Oper. Res. **26**(14), 1369–1394 (1999)
5. Plastria, F.: Static competitive facility location: an overview of optimisation approaches. Eur. J. Oper. Res. **129**(3), 461–470 (2001)
6. Eiselt, H.A., Laporte, G.: Sequential location problems. Eur. J. Oper. Res. **96**(2), 217–231 (1997)
7. Hotelling, H.: Stability in competition. Econ. J. **39**, 41–57 (1929)
8. Hakimi, S.L.: On locating new facilities in a competitive environment. Eur. J. Oper. Res. **12**(1), 29–35 (1983)
9. Drezner, Z.: Competitive location strategies for two facilities. Reg. Sci. Urban Econ. **12**(4), 485–493 (1982)
10. Bhadury, J., Eiselt, H.A., Jaramillo, J.H.: An alternating heuristic for medianoid and centroid problems in the plane. Comput. Oper. Res. **30**(4), 553–565 (2003)
11. Drezner, T., Drezner, Z.: Facility location in anticipation of future competition. Locat. Sci. **6**(1), 155–173 (1998)
12. Drezner, T., Drezner, Z.: Retail facility location under changing market conditions. IMA J. Manag. Math. **13**(4), 283–302 (2002)

13. Redondo, J.L., Fernández, J., García, I., Ortigosa, P.M.: Heuristics for the facility location and design (1|1)-centroid problem on the plane. Comput. Optim. Appl. **45**(1), 111–141 (2010)
14. Pardalos, P.M., Chinchuluun, A., Huang, H.X.: Multilevel (hierarchical) optimization: complexity issues, optimality conditions, algorithms. In: Gao D.Y., Sherali H.D. (eds) Advances in Applied Mathematics and Global Optimization in Honor of Gilbert Strang, Advances in Mechanics and mathematics. vol. 17, pp. 197–222 (2009)
15. Pardalos, P.M., Migdalas, A., Värbrand, P.: Multilevel Optimization: Algorithms and Applications. Kluwer, Dordrecht (1998)
16. Redondo, J.L., Fernández, J., Arrondo, A.G., García, I., Ortigosa, P.M.: Fixed or variable demand? Does it matter when locating a facility?. Omega **40**(1), 9–20 (2012)
17. Jelásity, M., Ortigosa, P.M., García, I.: UEGO, an abstract clustering technique for multimodal global optimization. J. Heuristics **7**(3), 215–233 (2001)
18. Fernández, J., Pelegrín, B., Plastria, F., Tóth, B.: Solving a Huff-like competitive location and design model for profit maximization in the plane. Eur. J. Oper. Res. **179**(3), 1274–1287 (2007)
19. Tóth, B., Plastria, F., Fernández, J., Pelegrín, B.: On the impact of spatial pattern, aggregation, and model parameters in planar huff-like competitive location and design problems. OR Spectr. **31**(1), 601–627 (2009)
20. Berman, O., Krass, D.: Locating multiple competitive facilities: spatial interaction models with variable expenditures. Ann. Oper. Res. **111**(1), 197–225 (2002)
21. Tóth, B., Fernández, J.: Interval methods for single and bi-objective optimization problems—applied to competitive facility location problems. Lambert Academic Publishing, Saarbrücken (2010)
22. Speer, N., Spieth, C., Zell, A.: A memetic co-clustering algorithm for gene exression profiles and biological annotation. In: Proceedings of the IEEE 2004 Congress on Evolutionary Computation, CEC 2004, vol. 2, pp. 1631–1638. IEEE Press (2004)
23. Solis, F.J., Wets, R.J.B.: Minimization by random search techniques. Math. Oper. Res. **6**(1), 19–30 (1981)