

Solving a Huff-like Stackelberg location problem on networks

Boglárka G.-Tóth · Kristóf Kovács

Received: January 15, 2015/ Accepted: date

Abstract This work deals with a Huff-like Stackelberg problem where the leader wants to locate a facility so that its profit is maximal after the competitor (the follower) has built its facility. We assume that the follower makes a rational decision, maximizing its own profit. The inelastic demand is aggregated into the vertices of a graph, and facilities can be located along the edges. For this computationally hard problem we give a Branch and Bound algorithm using interval analysis and DC bounds. Our computational experience shows that the problem can be solved on medium sized networks in reasonable time.

Keywords Stackelberg problem, bilevel optimization, Branch and Bound, interval analysis, DC decomposition

1 Introduction

Stackelberg location problems are about making location decisions of two competing firms that want to build new facilities. In the well-known Stackelberg model the *leader* locates its facilities first. Once the locations of the new facilities of the leader are set, the *follower* locates its new facilities. Each firm has an objective function maximizing the market share or profit of its facilities which depends both on the locations and other features (qualities) of the new and the existing facilities. The qualities of all existing and new facilities are supposed to be known and fixed. Since both firms try to maximize their market share or profit, the leader has to take into account the possible reactions of the follower, leading to a bilevel optimization problem.

This work is funded by the Hungarian National Research, Development and Innovation Office – NKFIH, OTKA grant PD115554 and also by the project ICT COST Action TD1207 (EU).

Boglárka G.-Tóth · Kristóf Kovács

Dept. of Differential Equations, Budapest University of Technology and Economics, E-mail: {bog,kkovacs}@math.bme.hu

In this bilevel problem the inner problem is similar to the Huff-like single facility location problem in [2]. Blanquero et al. deal with the problem of locating a new facility maximizing the market share when probabilistic customer choice is assumed. The problem is given by a network with the nodes as demand points and facilities positioned along the edges. A B&B algorithm is proposed with bounds using interval analysis and DC decomposition.

Sáiz et.al. [13] solved a Huff-like Stackelberg problem on the plane, also using a B&B method. The attraction (the measure of how attractive a facility is for customers located at a particular demand point) and market share was defined in the same way as in [2] with Euclidean distance and the facilities allowed to be anywhere in the convex hull of the demand points. Compared to the same problem on a network, the planar space makes the problem inherently more difficult, although in this work the zero-sum property of the objectives was of considerable help in the solution. Other examples of papers on Huff-like Stackelberg problems are the heuristic methods of Drezner and Drezner in [4], and a bilevel approach in [8] by Küçükaydin et al. Heuristic approaches are applied to the probabilistic (1|1)-centroid problem on the plane by Redondo et al. in [10, 9, 11], where both the locations and the designs (qualities) of the facilities are to be determined simultaneously. The objective function is the net profit taking into account the operational costs (depending both on the location and the quality) of a new facility. They worked with a model similar to ours but proposed only heuristic solutions. In [12], a two-stage method is devised to solve a similar facility location and design problem, where competitors react by changing the qualities of the existing facilities, but do not locate a new facility for the follower.

In this paper, we aim at the solution of the Stackelberg problem on networks, maximizing the profit of the leader. Literature on similar problems on the plane discusses only heuristic solutions. In contrast, our objective is to design a reliable method that gives a solution with prescribed accuracy. The difficulty of the problem compared to those addressed in earlier works with deterministic methods lies in the fact that the location space is a network and costs appear in the objective function. The latter makes the problem significantly harder, since the objective function does not have the zero-sum property as in [13]. Moreover, the objective function is not necessarily continuous. We explore this property in Section 3.

The paper is organized as follows. In Section 2 we formulate the model of this bilevel Stackelberg problem. The structural properties of the problem are examined in Section 3. In Section 4 we describe the Branch and Bound method designed to solve the leader's problem, the bounds used in the solution and the pseudocode of the algorithm. Computational results are shown and analyzed in Section 5. Finally conclusions are drawn in Section 6.

2 Model

Now we introduce the problem formally. Given a network $N = \langle V, E \rangle$, and for each edge $e \in E$ its length l_e . If we denote the endpoints of e by a_i and a_j , $e = (a_i, a_j)$, then $x \in [0, l_e]$ denotes the point on edge e at distance x from a_i and $l_e - x$ from a_j .

The fixed demand is concentrated at the vertices of N , where each $a \in V$ has a buying power of ω_a . The function $d_a(x)$ gives the distance between the demand point a and the facility at x . Assuming that x is located on edge e with endpoints a_i and a_j , the distance is given as

$$d_a(x) = \min\{d(a_i, a) + x, d(a_j, a) + l_e - x\},$$

where $d(a_i, a)$ is the length of the shortest path from demand point a_i to a in the network.

Introduce the following notations:

Indices

- a index of demand points ($a = 1, \dots, |V|$)
- j index of existing facilities ($j = 1, \dots, k$ for the leader,
 $j = k + 1, \dots, m$ for the follower)
- l, f index of the leader and the follower
- t, o index of the two firms when their role can be interchanged

Variables

- x_l position of the leader's new facility
- x_f position of the follower's new facility

Data

- x_j position of facility j
- q_j quality of facility j
- q_l quality of the leader's new facility
- q_f quality of the follower's new facility
- ω_a buying power of demand point a

Miscellaneous

- $d_a(x)$ distance between the demand point a and the facility located at x
- $\varphi(\cdot)$ a positive non-decreasing function
- $q/\varphi(d_a(x))$ the attraction a feels for the facility at x with quality q
- $\psi(\cdot)$ a positive non-decreasing function

We will assume that both firms are already present on the market, owning a number of facilities. Let us introduce the following notations for the total attraction of demand point a for the existing facilities owned by the leader, the follower and both of them, respectively (we assume in this paper that the attraction is additive),

$$\beta_a^l = \sum_{j=1}^k q_j / \varphi(d_a(x_j)), \quad \beta_a^f = \sum_{j=k+1}^m q_j / \varphi(d_a(x_j)), \quad \beta_a = \beta_a^l + \beta_a^f.$$

By using the indices t for one of the firms and o for the other firm instead of l and f , the above notation becomes β_a^t and β_a^o .

The market share of firm t (with the new facility at x_t and fixed quality q_t) after the other firm locates at \hat{x}_o is

$$M_t(x_t, \hat{x}_o) = \sum_{a \in V} \omega_a \frac{q_t / \varphi(d_a(x_t)) + \beta_a^t}{q_t / \varphi(d_a(x_t)) + q_o / \varphi(d_a(\hat{x}_o)) + \beta_a^o}.$$

The function $\varphi(\cdot)$ is a positive nondecreasing function defined on nonnegative reals. The usual choice is $\varphi(d) = d^\lambda$, $\lambda > 0$, where $\lambda = 2$ gives the so-called gravitational model. We assume that $\varphi(d) = d^\lambda$, thus the market share of firm t at demand point a , denoted by $m_a(x_t)$ can be written as

$$m_a(x_t) = \frac{q_t / d_a^\lambda(x_t) + \beta_a^t}{q_t / d_a^\lambda(x_t) + q_o / d_a^\lambda(\hat{x}_o) + \beta_a^o} = \frac{1 + \beta_a^t d_a^\lambda(x_t) / q_t}{1 + (q_o / d_a^\lambda(\hat{x}_o) + \beta_a^o) d_a^\lambda(x_t) / q_t}, \quad (1)$$

where the other firm's location is fixed at \hat{x}_o . Denoting

$$\gamma_a = \frac{q_o / d_a^\lambda(\hat{x}_o) + \beta_a^o}{q_t} \quad \text{and} \quad \alpha_a = \frac{\beta_a^t}{\gamma_a q_t} = \frac{\beta_a^t}{q_o / d_a^\lambda(\hat{x}_o) + \beta_a^o},$$

the market share of firm t at demand point a can be further simplified to

$$m_a(x_t) = \alpha_a + (1 - \alpha_a) \frac{1}{1 + \gamma_a d_a^\lambda(x_t)},$$

where $0 \leq \alpha_a \leq 1$ by definition. Thus, the total market share of firm t can be written as

$$M_t(x_t, \hat{x}_o) = \sum_{a \in V} \omega_a m_a(x_t) = \sum_{a \in V} \omega_a \left(\alpha_a + (1 - \alpha_a) \frac{1}{1 + \gamma_a d_a^\lambda(x_t)} \right). \quad (2)$$

Both firms are assumed to have operational costs depending on their proximity to the demand points and their quality. Costs near highly populated areas are likely to be higher, also better quality generates higher costs. Such behavior of the operational cost $G(x, q)$ of a facility at x with quality q , can be expressed as

$$G(x, q) = \sum_{a \in V} \omega_a \frac{q}{\psi(d_a(x))}, \quad (3)$$

where $\psi(\cdot)$ is a similar function to $\varphi(\cdot)$, though the two need not be the same. To transform the market share into expected sales, a linear function is used, where c is the income per unit of goods sold that could be different for the two firms. It is important to mention that other functions may be more suitable for describing cost and income behaviour and have to be carefully adjusted to a real situation, though their choice does not make much difference in the methodology introduced in this article.

By the above assumptions the profits of the two firms are

$$F_l(x_l, x_f) = cM_l(x_l, x_f) - G(x_l, q_l), \quad F_f(x_l, x_f) = cM_f(x_l, x_f) - G(x_f, q_f),$$

without the costs of existing facilities which are supposed to be constant.

Using the previous functions we can formulate the Stackelberg problem as

$$\begin{aligned} \max_{x_l \in [0, l_e], e \in E} \quad & F_l(x_l, x_f^*) \\ \text{s.t.} \quad & X_f^* = \operatorname{argmax}_{x_f \in [0, l_e], e \in E} F_f(x_l, x_f) \\ & x_f^* = \operatorname{argmax}_{\hat{x}_f \in X_f^*} M_f(x_l, \hat{x}_f), \end{aligned}$$

where X_f^* is the set of optimal locations of the follower maximizing its profit for a given x_l . The follower's choice, in the case of multiple optima, is always the one that maximizes its market share, thus minimizes that of the competitor. Equivalently, it minimizes the competitor's profit since the leader's operational cost does not depend on the follower's location. This is called the pessimistic approach in bilevel programming [3].

3 Discontinuity of the leader's objective function

The objective functions of both the leader's and the follower's problem are nonlinear nonconvex functions. For the market shares of the firms we have $M_l(y, z) + M_f(y, z) = W \forall y, z$, where W is the total demand, $W = \sum_{a \in V} \omega_a$. This is called the zero-sum property.

According to the following assertion, the follower's problem as a function of the leader's location

$$g(x_l) = \max_{x_f \in [0, l_h]} F_f(x_l, x_f), \quad (4)$$

is continuous on an edge e , $x_l \in [0, l_e]$ (assuming the follower to be located on edge h).

Assertion 1 *Let $D = [a, b] \times [c, d] \subset \mathbb{R}^2$ be a nonempty set. If $f : D \rightarrow \mathbb{R}$ is continuous, then $g(x) = \max_{y \in [c, d]} f(x, y)$ is continuous.*

This is a special case of Theorem 4.3 in [3].

The objective function of the leader is usually not continuous if operational costs are taken into account. However, the following result holds.

Assertion 2 *The leader's objective function F_l is continuous if operational costs are not taken into account, i.e. $F_t(x_t, x_o) = M_t(x_t, x_o)$.*

Proof By assumption, the leader's problem on edge e , when the follower is on edge h , is

$$\begin{aligned} \max_{x_l \in [0, l_e]} \quad & M_l(x_l, x_f^*) \\ \text{s.t.} \quad & x_f^* = \operatorname{argmax}_{x_f \in [0, l_h]} M_f(x_l, x_f), \end{aligned}$$

where multiple optima of the follower do not influence the objective of the leader by the zero-sum property. Using the zero-sum property again, we can reformulate $M_l(x_l, x_f^*) = W - M_f(x_l, x_f^*)$, and so the problem is equivalent to

$$\begin{aligned} \max_{x_l \in [0, l_e]} \quad & W - M_f(x_l, x_f^*) \\ \text{s.t.} \quad & x_f^* = \operatorname{argmax}_{x_f \in [0, l_h]} M_f(x_l, x_f), \end{aligned}$$

which, in turn, using the notation introduced in (4) is

$$\max_{x_l \in [0, l_e]} W - g(x_l).$$

As we saw in Assertion 1, $g(x_l)$ is a continuous function, hence the objective function of the leader's problem, $W - g(x_l)$, is also continuous. \square

We construct a counterexample where the leader's profit function is not continuous when costs are present. Let \hat{x}_l be the location of the leader, where there are more than one global optimizers for the follower's problem, i.e. $\operatorname{argmax}_{x_f} F_f(\hat{x}_l, x_f) = \{x_f^{1*}, x_f^{2*}, \dots\}$.

Assume also that $G(x_f^{1*}, q_f) \neq G(x_f^{2*}, q_f)$. Thus, $M_f(\hat{x}_l, x_f^{1*}) \neq M_f(\hat{x}_l, x_f^{2*})$, and by the zero-sum property the market share of the leader differs on these locations of the follower, i.e. $M_l(\hat{x}_l, x_f^{1*}) \neq M_l(\hat{x}_l, x_f^{2*})$. Given that the cost of the leader depends only on its location,

$$F_l(\hat{x}_l, x_f^{1*}) = cM_l(\hat{x}_l, x_f^{1*}) - G(\hat{x}_l, q) \neq cM_l(\hat{x}_l, x_f^{2*}) - G(\hat{x}_l, q) = F_l(\hat{x}_l, x_f^{2*}),$$

i.e. the leader's profit on the different optimizers of the follower is not the same.

The follower's choice between x_f^{1*} and x_f^{2*} is the location that maximizes its market share. Assume that $M_l(\hat{x}_l, x_f^{1*}) = M_l(\hat{x}_l, x_f^{2*}) + K$, where $K > 0$, hence x_f^{1*} is the optimizer for the market share. Furthermore we can assume, that there is a position \tilde{x}_l inside the ε -neighborhood of \hat{x}_l , where the follower's optimizer becomes x_f^{2*} . We have $F_l(\tilde{x}_l, x_f^{2*}) - F_l(\hat{x}_l, x_f^{1*}) \leq K + \delta(\varepsilon)$, but also $F_l(\tilde{x}_l, x_f^{2*}) - F_l(\hat{x}_l, x_f^{1*}) \geq K - \delta(\varepsilon)$, because of the continuity of G . Since K is independent of ε , F_l has a discontinuity somewhere between \hat{x}_l and \tilde{x}_l .

A specific example is a network, shown on Figure 1, with three demand points a_1, a_2, a_3 and demand $\omega_{a_i} = 1$, $i = 1, 2, 3$. Length of the two edges $(a_1, a_2), (a_2, a_3)$ are 3 and 4, respectively, qualities are $q_l = q_f = 1$ and there are no existing facilities. The gravitational model was used to calculate the utility, that is, $\varphi(d) = d^2$, and the income per units of goods sold was $c = 1$. The function ψ used in the definition of the operational cost was set to $\psi(d) = d^2 + 0.5$.

The leader's profit on edge (a_2, a_3) at $\hat{x}_l = 3.0285$ is 0.6307, and at $\hat{x}_l + \epsilon = 3.0286$ is 0.3872. The follower's choice is the point $x_f^{1*} \approx 1.641$ on edge (a_2, a_3) and $x_f^{2*} \approx 1.639$ on edge (a_1, a_2) , see Figure 1. The follower's market share and profit functions for the location of the leader \hat{x}_l and the leader's profit

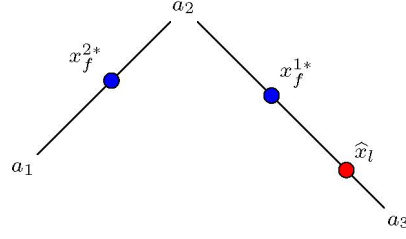


Fig. 1: The network for the counterexample with the leader's placement \hat{x}_l and follower's placements x_f^{1*} , x_f^{2*} , where $F_f(\hat{x}_l, x_f^{1*}) = F_f(\hat{x}_l, x_f^{2*})$ but $M_f(\hat{x}_l, x_f^{1*}) > M_f(\hat{x}_l, x_f^{2*})$.

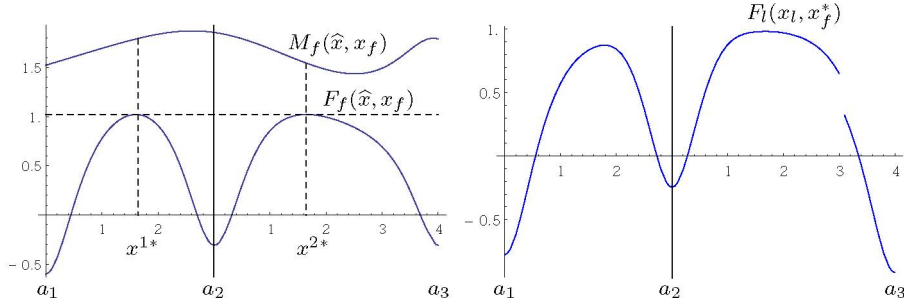


Fig. 2: The market share and profit functions of the follower for \hat{x}_l are shown on the left and the profit function of the leader with the optimal follower's placement $F_l(x_l, x_f^*)$, on the right.

function are shown in Figure 2. From the graphs the discontinuity of F_l can clearly be seen.

In the following section we propose an algorithm to solve both the follower's and the leader's problem.

4 Solution method

A Branch and Bound method is designed to solve the leader's problem, and consequently the follower's problem as well. The basic idea is that we simultaneously tighten the sets containing the global optimizer(s) of the leader's and follower's problem, respectively.

Since the search space is a network, we define subproblems of the leader as edges, or subsets of edges called segments or intervals, $X = [\underline{x}, \bar{x}] \subseteq [0, l_e]$, $e \in E$. For a given edge (or segment) of the leader the follower's possible position may lie on many edges of the graph, and until the leader is not enclosed tightly, the follower can only be bounded to a set of edges or segments. Thus, for every segment of the leader we store the segments of the follower that may contain

its global optimizer. Consequently, a partial solution or subproblem of the leader refers to a segment of the leader and the set of segments of the follower associated with it.

An inner B&B method is used to tighten the segments of the follower, and a main (outer) B&B method to tighten the segments of the leader. Therefore, we have to compute lower and upper bounds for the leader's (follower's) profit when the follower (leader) is constrained to an interval. For the calculation of the lower and upper bounds of a segment of the follower X_f , its single leader's segment X_l is taken into account. These lower and upper bounds are $LB(F_f(X_l, \hat{x}_f))$ and $UB(F_f(X_l, X_f))$, respectively, where \hat{x}_f is a feasible solution in the follower's segment. For the calculation of the bounds for a leader's segment X_l , every segment of the follower corresponding to it has to be considered, i.e. $LB(F_l(\hat{x}_l, \mathbb{X}_f))$ and $UB(F_l(X_l, \mathbb{X}_f))$, where \hat{x}_l is a feasible solution in the leader's segment and $\mathbb{X}_f \ni X_f$ the set of the corresponding segments of the follower.

In all cases we considered two estimations of the bounds: interval arithmetic and DC bounds combined with interval arithmetic.

4.1 Interval arithmetic bounds

Interval arithmetic is a means to obtain reliable results by putting bounds on rounding and measurement errors in the first place. We use it to obtain lower and upper bounds on the objective functions automatically. See [5] for details of interval analysis in global optimization.

Let us denote intervals with capital letters e.g. $X = [\underline{x}, \bar{x}]$, where $\underline{x} \leq \bar{x}$ are the lower and upper bounds of X , respectively. Using this notation, the distance between an interval X and demand point a can be given as

$$d_a^\lambda(X) = \left[\underline{d_a^\lambda(X)}, \overline{d_a^\lambda(X)} \right], \quad \underline{d_a^\lambda(X)} = \min \{ d_a^\lambda(\underline{x}), d_a^\lambda(\bar{x}) \},$$

$$\overline{d_a^\lambda(X)} = \left(\max \{ d_a(\underline{x}), d_a(\bar{x}) \} + \frac{\bar{x} - \underline{x} - |d_a(\underline{x}) - d_a(\bar{x})|}{2} \right)^\lambda.$$

The upper bound of the distance is exact, and computed as the maximal distance to the endpoints of the segment plus the maximal increase of the distance in the segment. This latter is the half of the width of the segment after the absolute difference $|d_a(\underline{x}) - d_a(\bar{x})|$ has been removed, see Figure 3.

When calculating interval arithmetic bounds for the profit function, we use form (2) of the market share. Having \hat{X}_o , a fixed interval (or a point), the upper bound of F_t calculated with interval arithmetic is

$$UB_{IA}(F_t(X_t, \hat{X}_o)) = UB_{IA}(cM_t(X_t, \hat{X}_o)) - LB_{IA}(G(X_t, q_t))$$

$$UB_{IA}(M_t(X_t, \hat{X}_o)) = \sum_{a \in V} \omega_a \left(\overline{\alpha_a} + (1 - \alpha_a) \frac{1}{1 + \gamma_a \underline{d_a^\lambda(X_t)}} \right),$$

inflection point, thus its DC decomposition is easy to obtain, see [2]. Namely $h_a(d) = h_a^+(d) - h_a^-(d)$, where

$$h_a^+(d) = \begin{cases} h_a(c_a) + h'_a(c_a)(d - c_a) & \text{if } d \leq c_a \\ h_a(d) & \text{if } d > c_a \end{cases},$$

$$h_a^-(d) = \begin{cases} h_a(c_a) + h'_a(c_a)(d - c_a) - h_a(d) & \text{if } d \leq c_a \\ 0 & \text{if } d > c_a \end{cases},$$

$$c_a = \left(\frac{\lambda - 1}{(1 + \lambda)\gamma_a} \right)^{\frac{1}{\lambda}}.$$

where c_a is the inflexion point of $h_a(d)$. Using the following proposition, we can get a DC decomposition for $m_a(x_t)$.

Assertion 3 (Blanquero et al. [2]) *Let $I \subset \mathbf{R}$ be an interval. Let $d : I \rightarrow \mathbf{R}$ be a concave function on I , and let $g : \mathbf{R} \rightarrow \mathbf{R}$ be DC, with a DC decomposition given by $g(x) = g^+(x) - g^-(x)$, with both g^+ and g^- non-increasing functions. Then, the function $f : I \rightarrow \mathbf{R}$ defined as $g(d(x))$ is DC on I and a DC decomposition is given by $f(x) = f^+(x) - f^-(x)$ where $f^+(x) = g^+(d(x))$ and $f^-(x) = g^-(d(x))$.*

Therefore, the DC decomposition of $m_a(x_t)$ is $m_a(x_t) = m_a^+(x_t) - m_a^-(x_t)$, where

$$m_a^+(x_t) = \alpha_a + (1 - \alpha_a)h_a^+(d_a(x_t)),$$

$$m_a^-(x_t) = (1 - \alpha_a)h_a^-(d_a(x_t)),$$

since $d_a(\cdot)$ is concave and $0 \leq \alpha < 1$. Finally, the DC decomposition of the market share is $M_t(x_t, \hat{x}_o) = M_t^+(x_t, \hat{x}_o) - M_t^-(x_t, \hat{x}_o)$, where

$$M_t^+(x_t, \hat{x}_o) = \sum_{a \in V} \omega_a m_a^+(x_t), \quad M_t^-(x_t, \hat{x}_o) = \sum_{a \in V} \omega_a m_a^-(x_t).$$

To construct an upper bound on M_t we need the upper bound of M_t^+ and the lower bound of M_t^- . The upper bound of M_t^+ is easily obtained. Let $x_t \in X_t = [\underline{x}_t, \bar{x}_t]$, then the upper bound is

$$UB_{DC}(M_t^+(X_t, \hat{x}_o)) = \max\{M_t^+(\underline{x}_t), M_t^+(\bar{x}_t)\},$$

since M_t^+ is a convex function. The lower bound of M_t^- can be computed from its linear underestimator, a tangent line at a point $\tilde{x}_t \in X_t$

$$LB_{DC}(M_t^-(X_t, \hat{x}_o)) = \min\{l(\underline{x}_t), l(\bar{x}_t)\},$$

where $l(x) = (M_t^-(\tilde{x}_t, \hat{x}_o))'(x - \tilde{x}_t) + M_t^-(\tilde{x}_t, \hat{x}_o)$. A DC upper bound of M_t utilizing the results obtained above is given by

$$UB_{DC}(M_t(X_t, \hat{x}_o)) = UB_{DC}(M_t^+(X_t, \hat{x}_o)) - LB_{DC}(M_t^-(X_t, \hat{x}_o)).$$

The DC decomposition of $M_t(\hat{x}_t, X_o)$, where we calculate the market share of a firm with its own location fixed, can be easily obtained from the zero-sum property, that is, $M_t(\hat{x}_t, X_o) = W - M_o(X_o, \hat{x}_t)$.

The lower bound of the market share $LB_{DC}(M_t)$ can be computed similarly. The DC decomposition of the cost function $G(x, q)$ can also be obtained from the decomposition of $h_a(d)$. Thus we have DC bounds for the profit as well.

4.3 DC bounds with interval arithmetic

When calculating the DC bounds the other firm's position can be given by an interval X_o . In this case the market share of this firm x_t is an interval valued function, see Figure 4. We can take the lower or upper bounding function of this interval valued function and calculate the DC bounds on these function.

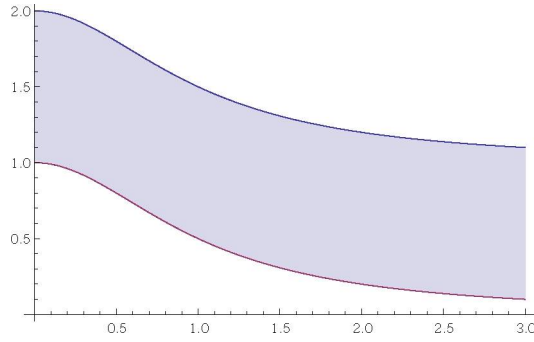


Fig. 4: The market share as an interval valued function for X_o .

Essentially, we have to calculate DC bounds for the lower bounding function

$$\sum_{a \in V} \omega_a \left(\frac{\alpha_a}{1 + \gamma_a d_a^\lambda(x_t)} + (1 - \alpha_a) \frac{1}{1 + \gamma_a d_a^\lambda(x_t)} \right),$$

where the DC decomposition and so the DC bounds of

$$\frac{1}{1 + \gamma_a d_a^\lambda(x_t)}.$$

can be calculated in the same way as we did in Subsection 4.2.

The following two subsections describe the inner and outer B&B methods designed to solve the leader's problem, using the bound calculations we have given above.

4.4 Inner Branch and Bound: refining the segments of the follower

We need to refine both the leader's and their corresponding follower's segments for the algorithm in order to achieve convergence. The inner B&B takes care of the refinement of the segments of the follower.

The stopping criterion of the inner B&B is to make the diameter of each segment of the follower at least as small as the corresponding segment of the leader. The output of the algorithm is the modified list of the segments of the follower. The selection rule chooses the widest segment, while the branching rule splits the given segment at its midpoint.

This method ensures for every segment of the leader, that its corresponding follower's segments will have the same size as the leader's segment. Each time a new leader segment is created, the inner B&B runs until its follower's segments are refined.

4.5 Outer Branch and Bound: solving the leader problem

The outer B&B refines the leader's segments and calls the inner B&B method for each new segment of the leader. Recall that a subproblem of the leader is an edge (or segment) with the corresponding set of segments for the follower. Thus, the initial subproblems are the edges as segments of the leader, and for each edge of the leader the set of edges as segments of the follower.

The output is the set of segments with objective value close to the global optimum enclosing the global optimizer(s). The selection rule selects the partial solution with the highest upper bound on the leader's profit, while the branching rule bisects the leader's segment at its midpoint and leaves the follower's segments unchanged but duplicated for the new leader's segments. The algorithm terminates when either the (relative or absolute) gap of the objective of all leader's segments get smaller than a prescribed ε or their length becomes smaller than a tolerance parameter δ . Formally, a leader's segment X_l gets on the solution list if

$$UB(F_l(X_l, \mathbb{X}_f)) - z^L < \varepsilon \text{ or } \frac{UB(F_l(X_l, \mathbb{X}_f)) - z^L}{UB(F_l(X_l, \mathbb{X}_f))} < \varepsilon \text{ or } width(X_l) < \delta$$

where \mathbb{X}_f is the set of the segments of the follower for X_l and z^L is the best objective value for the leader found so far by the algorithm.

4.6 Algorithm

Algorithm 1 initializes the outer B&B's partitions using the input network. The outer cycle defines the segments of the leader, while the inner cycle initializes the segments of the follower for every leader's segment. An upper bound z_e^U for edge $e \in E$ and a global lower bound z^L for the leader's problem are calculated. For each edge of the leader $e \in E$ and that of the follower $h \in E$

we compute lower and upper bounds v_{eh}^L and v_{eh}^U for the follower's profit. The global lower bound for the follower's problem is $v_e^L = \min v_{eh}^L$. The output is the set of initial subproblems A .

Algorithm 1: Initialization of the outer B&B

```

Input :  $N = \langle V, E \rangle$ 
1  $A := \{\}, z^L := 0$ 
2 foreach  $e \in E$  do
3    $X_e := [0, l_e], \mathbb{Y} = \{\}$ 
4    $v_e^L := 0$ 
5   foreach  $h \in E$  do
6      $Y_h := [0, l_h]$ 
7     Determine an upper bound  $v_{eh}^U$  of  $F_f$  over  $Y_h$ 
8     Compute lower bound  $v_{eh}^L$  of  $F_f$  at midpoint( $Y_h$ )
9     if  $v_{eh}^L > v_e^L$  then
10      |  $v_e^L := v_{eh}^L$ 
11    end
12    Associate  $Y_h$  as a follower segment with  $X_e, \mathbb{Y} = \mathbb{Y} \cup \{Y_h\}$ 
13  end
14  Determine an upper bound  $z_e^U$  of  $F_l$  over  $X_e$ 
15  Compute lower bound  $z_e^L$  of  $F_l$  at midpoint( $X_e$ )
16  if  $z_e^L > z^L$  then
17    |  $z^L := z_e^L$ 
18  end
19   $A := A \cup (X_e, \mathbb{Y})$ 
20 end
Output:  $A, z^L$ 

```

The pseudocodes of the inner and outer B&B algorithms are given in Algorithm 2. For the sake of simplicity let us denote the objective function by F_t (F_l for the outer and F_f for the inner B&B).

In line 1 we remove each segment from the partitions A known not to contain any global optimizer. The main cycle of the general Branch and Bound method is listed from line 2 to line 23. The main difference of the outer B&B from the inner B&B is the call of the inner method added in lines 14 to 16. In fact, the additional differences between the inner and outer procedure are hidden in the bound calculations, as well as in the selection and termination rules.

The output of Algorithm 2 is the set of segments which could not be eliminated and thus contain the global optimizer and the point at which the best lower bound was achieved.

5 Computational results

In this section we present the computational results of the algorithm described in Section 4. The algorithm was written in C++ using the PROFIL/BIAS

Algorithm 2: The inner and outer B&B method

```

Input :  $\Lambda, z^L$ 
1 Remove all  $X_i$  from  $\Lambda$  with  $z_i^U < z^L$ 
2 while  $\Lambda \neq \emptyset$  do
3   Select  $X$  from  $\Lambda$ 
4   Bisect  $X$  into  $X_1$  and  $X_2$ 
5   for  $i := 1$  to 2 do
6     Determine an upper bound  $z_i^U$  on  $X_i$ 
7     if not  $z_i^U < z^L$  then
8       Compute a lower bound  $z_i$  of  $F_t$  at  $\text{midpoint}(X_i)$ 
9       if  $z_i > z^L$  then
10         $z^L := z_i$ ,  $\text{BestPoint} := \text{midpoint}(X_i)$ 
11        Remove all  $X_i$  from  $\Lambda$  with  $z_i^U < z^L$ 
12      end
13      if not  $\text{TerminationCriterion}(X_i)$  then
14        if outer then
15          Call the inner B&B on the set of follower segments of  $X_i$ 
16        end
17         $\Lambda := \Lambda \cup \{X_i\}$ 
18      else
19         $\Gamma := \Gamma \cup \{X_i\}$ 
20      end
21    end
22  end
23 end
Output:  $\Gamma, \text{BestPoint}$ 

```

library [7], and executed on a cluster with 18 nodes. Each node has 16 cores (Intel Xeon E5 2650) and 64 GB of shared memory. Throughout the execution the memory requirement has never exceeded 4GB.

The relative and absolute accuracy used for the leader problem was $\varepsilon = 10^{-3}$, while the tolerance for the length of segments was $\delta = 10^{-6}$. The attraction was chosen according to the gravitational model, i.e. $\varphi(d) = d^2$, while the function for the operational costs was $\psi(d) = d^2 + b$ with $b = 0.5$. The income per unit of goods sold c , was chosen to be 1. The containers for the partial solutions were augmented red-black trees (a general balanced binary search tree) in all cases, sorted according to the selection rule.

The algorithm was tested on 8 networks from [2] where the number of nodes ranges from 150 to 298 and the number of edges varies from 296 to 597. For a given network several instances of the problem were generated using different numbers of existing facilities and different distributions for the two firms. The number of facilities tested were $m \in \{0, 8, 32\}$ each with distributions $k/m\% \in \{25\%, 50\%, 75\%\}$ denoting the percentage of facilities owned by the leader. Obviously for zero existing facilities the distribution was omitted. For each pair $(m, k/m\%)$ 5 problems were tested, with randomly generated demand at the vertices (demand points) using uniform distribution in the interval $[0, 10]$. The location of the facilities was generated by first choosing an edge e randomly, then placing the facility uniformly on the interval $[0, l_e]$.

The algorithm was first tested on two networks RAT195G and PR152G with many combinations of the different bounds for the inner and outer methods. In Table 1 the mean execution times in seconds are shown for the different combinations. In the first four lines the bounds used for a given column are shown, where DC stands for DC bounds, and IA stands for Interval Arithmetic bounds. We have highlighted the best result with a dark grey background, and the second best by light grey.

Table 1: Computational time in seconds for different combinations of upper and lower bounds for the problems of the leader and follower.

Leader	LB	DC	IA	DC	DC	IA	IA	DC	
	UB	DC	IA	IA	DC	IA	DC	DC	
Follower	LB	IA	DC	DC	DC	IA	IA	IA	
	UB	DC	IA	IA	DC	IA	IA	IA	
Network (nodes, edges)	m	$k/m\%$	Mean execution times in seconds						
PR152G (152, 296)	0	565	361	406	472	433	441	487	
	8	25	985	458	497	770	607	605	659
		50	749	349	380	546	466	465	508
		75	827	276	303	732	330	336	368
	32	25	899	365	401	727	467	460	505
		50	982	389	426	856	453	478	524
		75	819	263	290	736	299	310	342
	RAT195G (195, 336)	0	1902	1144	1214	1028	1870	1864	2001
8		25	2610	1374	1475	1376	2371	2298	2483
		50	1272	722	773	748	1180	1147	1232
		75	1259	713	760	771	1121	1106	1183
32		25	3710	1523	1626	2352	2438	2437	2645
		50	3404	1255	1359	2205	1998	1992	2171
		75	2506	959	1032	1542	1593	1589	1716

In all but one case the best computational time was reached by the combination of bounds, where the lower bound of the follower is calculated with DC decomposition, and all the other bounds with interval arithmetics. This may be explained by the higher computational requirement of the DC bounds. Surprisingly the differences between the configurations are rather small, and their ranking is not clear except for the winner.

The algorithm with the best combination of bounds was tested on the graphs with less than 300 nodes from [2]. The results are shown in Table 2, where the columns define the configuration of the existing facilities, and each row represents a network. For each network the number of nodes and edges are given by their names. Notice, that the average computational time for the different networks does not really depend on their size, although it is clear that the last graph UR532 is much harder to handle than the rest. For all problems the program could terminate in less than 4 hours except in a few cases, which are marked by * in the table since the average could not be computed. The last line shows the average values for the different graphs

Table 2: Average computational time in seconds for each settings using the best configuration of the bounds.

m $k/m\%$		0	8			32		
			25	50	75	25	50	75
Network	(Nodes, Edges)	Mean execution time in seconds						
KROA150G	(150, 297)	583	686	499	523	569	724	405
KROB150G	(150, 296)	427	586	521	436	861	735	524
PR152G	(152, 296)	361	458	349	276	365	389	263
RAT195G	(195, 336)	1144	1374	722	713	1523	1255	959
KROA200G	(200, 392)	755	735	712	469	987	943	1080
KROB200G	(200, 386)	592	1032	612	700	1670	1462	735
TS225G	(225, 306)	785	594	510	504	1295	930	582
UR532	(298, 597)	4212	3697	2852	*	3444	3236	*
Average*		664	781	561	517	1039	920	650

except the network UR532, where there were cases without results. It can be seen that the problems where the leader owns more existing facilities are generally easier, while more existing facilities overall make the problem more difficult to solve. In this sense, UR532 is an outlier, since it could not be solved when the leader owns a high portion of the existing facilities. Also, among the solved configurations the one with no existing facilities demanded the highest average computational time for the 5 generated problems.

6 Concluding remarks

In this paper a Huff-like Stackelberg problem on networks was studied which, unlike in other works in the literature, also considers operational costs for the new facilities. Having thus made the model more realistic it has become much harder to solve due to discontinuities that may emerge. The follower's problem may have several global optimizers for a given leader, hence selection among them is an issue to be taken care of. In a competitive setting the pessimistic choice is rather realistic, i.e. the follower minimizes the leader's profit. It is known that the pessimistic approach makes bilevel problems much harder [3].

We proposed a deterministic method for the solution of this Stackelberg problem. It is a Branch and Bound method designed to solve the leader's problem using an embedded B&B to solve the follower's problem as well. For bound calculations interval arithmetic and DC decomposition was used, as well as a mixed interval DC bound in some cases.

The computational results clearly show that using the DC bound pays off really well for computing the lower bound of the follower, while for the other bounds it is still competitive. The results show that the problem can be solved for medium sized networks in generally less than 20 minutes.

There is plenty of room for further research in this area. First of all, other bounding procedures may speed up the algorithm, as well as paralellization may bring the solution of larger-size problems within reach. An interesting

extension of the problem is to consider the quality of the facilities as decision variables. This calls for new bounding procedures.

References

1. R. Blanquero and E. Carrizosa. Continuous location problems and big triangle small triangle: constructing better bounds. *Journal of Global Optimization*, 45(3):389–402, 2009.
2. R. Blanquero, E. Carrizosa, A. Nogales-Gómez, and F. Plastria. Single-facility huff location problems on networks. *Annals of Operations Research*, pages 1–21, 2013.
3. S. Dempe. *Foundations of bilevel programming*. Springer, 2002.
4. T. Drezner and Z. Drezner. Facility location in anticipation of future competition. *Location Science*, 6(1–4):155–173, 1998.
5. E. Hansen and G.W. Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003.
6. R. Horst and N.V. Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.
7. O. Knüppel. PROFIL/BIAS - a fast interval library. *Computing*, 1(53):277–287, 1993.
8. H. Küçükdin, N. Aras, and I. Kuban Altinel. Competitive facility location problem with attractiveness adjustment of the follower: A bilevel programming model and its solution. *European Journal of Operational Research*, 208(3):206–220, 2011.
9. J.L. Redondo, A.G. Arrondo, J. Fernández, I. García, and P.M. Ortigosa. A two-level evolutionary algorithm for solving the facility location and design (1|1)-centroid problem on the plane with variable demand. *Journal of Global Optimization*, 56(3):983–1005, 2013.
10. J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. Solving the facility location and design (1|1)-centroid problem via parallel algorithms. *The Journal of Supercomputing*, 58(3):420–428, 2011.
11. J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. Heuristics for the facility location and design (1|1)-centroid problem on the plane. *Computational Optimization and Applications*, 45(1):111–141, 2010.
12. N. Saidani, F. Chu, and H. Chen. Competitive facility location and design with reactions of competitors already in the market. *European journal of operational research*, 219(1):9–17, 2012.
13. M.E. Sáiz, E.M.T. Hendrix, J. Fernández, and B. Pelegrín. On a branch-and-bound approach for a Huff-like Stackelberg location problem. *OR Spectrum*, 31:679–705, 2009.