The inverse 1-median problem on a cycle [*]

Rainer E. Burkard [†] Carmen Pleschiutschnig [‡] Jianzhong Zhang [§]

Dedicated to the memory of Professor G.B. Dantzig

Abstract. Let the graph $G = (V, E)$ be a cycle with $n + 1$ vertices, nonnegative vertex weights and positive edge lengths. The inverse 1-median problem on a cycle consists in changing the vertex weights at minimum cost such that a prespecified vertex becomes the 1-median. The cost is proportional to the increase or decrease of the corresponding weight. We show that this problem can be formulated as a linear program with bounded variables and a special structure of the constraint matrix: the columns of the linear program can be partitioned into two classes in which they are monotonically decreasing. This allows to solve the problem in $O(n^2)$-time.

Keywords: Location problem, 1-median, inverse optimization, special linear program

AMS-classification: 90B80, 90B85, 90C27, 90C05

1 Introduction and problem statement

Inverse optimization problems have recently found a considerable interest. In an *inverse optimization problem* an instance of an optimization problem and a special feasible solution are given. The task is to change the given parameters of the problem at minimum cost such that the given feasible solution becomes optimum. In 1992, Burton and Toint [4] introduced the inverse shortest path problem with an interesting application to geological sciences. Given a network, they change the edge lengths as little as possible such that a given path becomes the shortest path. Cai, Yang and Zhang [5] proved that the inverse center location problem is $\mathcal{NP}$-hard, though the underlying center location problem is

[*] The second author acknowledges financial support by the Spezialforschungsbereich F 003 "Optimierung und Kontrolle", Projektbereich Diskrete Optimierung. The first and third author acknowledge partial support of Hong Kong University Grant Council under the grant 9040883 (CITYU 103003).

[†] burkard@tugraz.at. Institute of Optimization and Discrete Mathematics, Graz University of Technology, Steyrergasse 30, A-8010 Graz, Austria.

[‡] pleschiutschnig@opt.math.tu-graz.ac.at. Institute of Optimization and Discrete Mathematics, Graz University of Technology, Steyrergasse 30, A-8010 Graz, Austria.

[§] mazhang@cityu.hk.edu. Department of Mathematics, City University of Hong Kong, Hong Kong.

polynomially solvable. Recently, Burkard, Pleschiutschnig and Zhang [3] showed that the inverse 1-median problem on a tree (and in the plane where distances are measured in the l_1-metric) can be solved by a greedy algorithm. For further results on inverse optimization including network and location models we refer the interested reader to the survey on inverse optimization compiled by Heuberger [6].

In contrast to inverse optimization, a *reverse optimization problem* tries to improve the objective function value of a given feasible solution by changing the parameters of the optimization problem at cost within a given budget. In 1992, Berman, Ingco and Odoni [1] published a paper on how to improve a transportation network by changing the length of the arcs and by introducing new arcs in order to improve the minisum objective value for a given facility. A similar question was treated by Zhang, Liu and Ma [9]. They present a strongly polynomial algorithm for shortening the lengths in a tree network within a given budget such that the longest distance from a given facility to all other nodes becomes minimum. On the other hand, the reverse 1-median problem on a cycle was solved by Burkard, Gassner and Hatzl [2]. In the latter paper the task is to use a budget for changing the length of some edges such that the overall sum of the weighted distances to a prespecified vertex becomes as small as possible.

In this paper we investigate the inverse 1-median problem on a cycle. The 1-median problem on a cycle can be stated in the following way. Let an (undirected) cycle graph $G = (V, E)$ with $n + 1$ clockwise numbered vertices be given, i.e., $|V| = |E| = n + 1$ and $V = \{v_0, v_1, \dots, v_n\}$ and $E = \{e_0, e_1, \dots, e_n\}$ where $e_j = [j, j + 1]$ for $j = 0, 1, \dots, n - 1$ and $e_n = [n, 0]$. All edges $e_j \in E$, $j = 0, 1, \dots, n$, have a positive length $l_j > 0$. Moreover, every vertex $v_j \in V$, $j = 0, 1, \dots, n$, has a nonnegative weight w_j. Let $d(i, j)$ denote the length of a shortest path between the two vertices v_i and v_j. The objective of the 1-median problem is to find a vertex $v_s \in V$ for which

$$\sum_{j=0}^{n} d(j, s) w_j$$

is minimum.

In the *inverse* 1*-median problem* on a cycle we want to change the vertex weights at minimum cost such that vertex v_0 becomes a 1-median of the given cycle. Every weight w_j can only be changed between a lower bound $\underline{w}_j \geq 0$ and an upper bound $\overline{w}_j$. We assume that the cost for changing each weight w_j by one unit is the same, say 1. Thus the total cost is measured by the function

$$\sum_{j=0}^{n} (p_j + q_j),$$

where p_j is the amount by which the weight w_j of vertex v_j is increased and q_j the amount by which w_j is decreased. We call a solution (p_j, q_j), $j = 0, 1, \dots, n$, feasible, if it guarantees that v_0 is a 1-median and all bounds for the weights are met.

Using this notation, the inverse 1-median problem on a cycle can be written as the linear program

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j=0}^{n} (p_j + q_j) \\
\text{s. t.} \quad & \sum_{j=0}^{n} (d(i,j) - d(0,j))(p_j - q_j) \geq \sum_{j=0}^{n} (d(0,j) - d(i,j)) w_j, \qquad i = 1, \dots, n, \quad (1) \\
& 0 \leq p_j \leq \overline{w}_j - w_j, \quad j = 0, 1, \dots, n, \\
& 0 \leq q_j \leq w_j - \underline{w}_j, \quad j = 0, 1, \dots, n.
\end{aligned}
$$

The constraints say that after a change of the weights the weighted sum of the distances from any vertex v_i to all other vertices is at least as large as the weighted sum of the distances from vertex v_0 to all other vertices, i.e., that vertex v_0 is a 1-median. Note that the coefficients of p_j and q_j differ only in their sign. It will turn out that the program (1) has a special structure which allows a fast solution of this problem.

In the next section, the linear program (1) will be analyzed. We will show that the columns of the constraint matrix can be partitioned into two classes such that in each class the entries of the matrix decrease in each row. This property implies that every optimal solution has a special form. Therefore, the problem can be written as (nonlinear) program in two variables as stated in Section 3. Based on this, we propose two different algorithms that solve the problem. The global solution method (Section 4) is a geometric approach that determines for each inequality in (1) the feasible set in $O(n)$ time. In this way, the problem can be written as linear program with two variables and $O(n^2)$ constraints. Since a linear program in two variables and n constraints can be solved in $O(n)$ time, a solution can be found in $O(n^2)$ time. Though this algorithm runs theoretically in $O(n^2)$ time, it is rather involved and therefore computationally impractical. For this reason, we state in Section 5 another simple solution method that is very fast in practice. The solution process of this iterative algorithm consists of two phases. First, either a feasible solution of the problem is found or it is shown that the problem is infeasible. In the second phase, an optimal solution is determined. In each iteration step of this algorithm, "local linear programs" with two variables and $O(n)$ constraints have to be solved. As such linear programs can easily be solved in linear time, this method quickly finds an optimal solution.

2 Properties of the Linear Program

We denote the length of the cycle by $L := \sum_{j=0}^{n} l_j$. Let $d^R(i,j)$ denote the distance between the vertices v_i and v_j when going clockwise from v_i to v_j and let $d^L(i,j)$ denote the counterclockwise distance between v_i and v_j. Further, M be the point on the cycle which is opposite to v_0, i.e., $d^R(v_0, M) = d^L(v_0, M) = L/2$. The point M may or may not coincide with a vertex. We define a new distance function $\hat{d}(j)$ for the vertices v_j,

$j = 0, 1, ..., n$, by

$$\hat{d}(j) := \begin{cases} 0 & \text{if } j = 0, \\ d(j, M) & \text{if } d(0, j) = d^R(0, j), \\ d(0, j) & \text{if } d(0, j) = d^L(0, j). \end{cases} \tag{2}$$

This means, $\hat{d}(j)$ is the clockwise distance between vertex v_j and the midpoint M of the cycle if v_j lies in the right half of the cycle, and $\hat{d}(j)$ is the clockwise distance between vertex v_j and vertex v_0 if v_j lies in the left half of the cycle.

Now we order the distances $\hat{d}(j)$ increasingly:

$$0 = \hat{d}(\pi(0)) \leq \hat{d}(\pi(1)) \leq \hat{d}(\pi(2)) \leq \cdots \leq \hat{d}(\pi(n)) \tag{3}$$

and we rewrite the linear program (1) as

$$\begin{aligned} \text{minimize} \quad & \sum_{j=0}^{n} x_j + \sum_{j=0}^{n} y_j \\ \text{s. t.} \quad & Ax + \bar{A}y \geq b, \\ & 0 \leq x \leq \overline{x}, \\ & 0 \leq y \leq \overline{y}, \end{aligned} \tag{4}$$

with $x_0 := p_0$ and

$$x_j := \begin{cases} q_{\pi(j)} & \text{if } v_{\pi(j)} \text{ is in the right halfcycle (incl. } M) \\ p_{\pi(j)} & \text{if } v_{\pi(j)} \text{ is in the left halfcycle} \end{cases} \qquad \text{for } 1 \leq j \leq n,$$

$$y_j := \begin{cases} p_{\pi(n-j)} & \text{if } x_{n-j} = q_{\pi(n-j)} \\ q_{\pi(n-j)} & \text{if } x_{n-j} = p_{\pi(n-j)} \end{cases} \qquad \text{for } 0 \leq j \leq n.$$

The coefficients of $A = (a_{ij})$ and $\bar{A} = (\bar{a}_{ij})$ are given by

$$a_{ij} := \begin{cases} d(i, 0) & \text{for } j = 0, \\ (-1)(d(i, \pi(j)) - d(0, \pi(j))) & \text{for } x_j = q_{\pi(j)}, \\ (d(i, \pi(j)) - d(0, \pi(j))) & \text{for } x_j = p_{\pi(j)} \end{cases}$$

and

$$\bar{a}_{ij} := -a_{i,n-j}.$$

The right hand side b has the coefficients

$$b_i := \sum_{j=0}^{n} (d(0, j) - d(i, j))w_j \quad \text{for } i = 1, \ldots, n.$$

The values $\overline{x}_j$ and $\overline{y}_j$ are the upper bounds of the original variables $p_{\pi(j)}$ and $q_{\pi(j)}$ which correspond to x_j and y_j.

Proposition 2.1 *The columns of matrix A ($\bar{A}$) are monotonically decreasing, i.e.,*

$$a_{ij} \geq a_{ik} \text{ for } 0 \leq j < k \leq n; \; i = 1, ..., n, \tag{5}$$

$$\bar{a}_{ij} \geq \bar{a}_{ik} \text{ for } 0 \leq j < k \leq n; \; i = 1, ..., n. \tag{6}$$

Proof. The result for matrix $\bar{A}$ is a direct consequence of $\bar{a}_{ij} = -a_{i,n-j}$ and (5). So, we only have to prove property (5).
We consider column 0 separately. If column j corresponds to a p-variable, then the triangle-inequality implies

$$a_{ij} = d(i, \pi(j)) - d(0, \pi(j)) \leq d(i, 0) + d(0, \pi(j)) - d(0, \pi(j)) = d(i, 0) = a_{i0}.$$

The case of column j corresponding to a q-variable can be shown in an analogous way. So, we have $a_{i0} \geq a_{ij}$ for all $i = 1, \ldots, n$ and for all $j = 1, \ldots, n$. Hence, it suffices to consider columns j and k with $1 \leq j < k \leq n$. We consider four cases.
Case 1: Both vertices $v_{\pi(j)}$ and $v_{\pi(k)}$ lie in the right halfcycle (including M).
In this case, both variables x_j and x_k correspond to q-variables. Since $\hat{d}(\pi(j)) \leq \hat{d}(\pi(k))$ we get

$$d(0, \pi(j)) = d(0, \pi(k)) + d(\pi(k), \pi(j)).$$

By means of the triangle-inequality we obtain for $i = 1, \ldots, n$

$$\begin{aligned} a_{ij} &= d(0, \pi(k)) + d(\pi(k), \pi(j)) - d(i, \pi(j)) \\ &\geq d(0, \pi(k)) + d(\pi(k), \pi(j)) - d(i, \pi(k)) - d(\pi(k), \pi(j)) = a_{ik}. \end{aligned}$$

Case 2: Both vertices $v_{\pi(j)}$ and $v_{\pi(k)}$ lie in the left halfcycle.
In an analogous way as in the first case it can be shown that also in this case $a_{ij} \geq a_{ik}$ holds.
Case 3: Vertex $v_{\pi(j)}$ lies in the left halfcycle, vertex $v_{\pi(k)}$ lies in the right halfcycle.
In this case, x_j corresponds to a p-variable and x_k to a q-variable. Using $\hat{d}(\pi(j)) \leq \hat{d}(\pi(k))$, we see that the shorter way between $v_{\pi(j)}$ and $v_{\pi(k)}$ on the cycle leads via the vertex v_0. Thus, by means of the triangle-inequality, we get

$$d(0, \pi(k)) + d(0, \pi(j)) = d(\pi(j), \pi(k)) \leq d(i, \pi(j)) + d(i, \pi(k)),$$

which shows that $a_{ij} \geq a_{ik}$ holds for all $i = 1, \ldots, n$.
Case 4: Vertex $v_{\pi(j)}$ lies in the right halfcycle, vertex $v_{\pi(k)}$ lies in the left halfcycle.
In this case, x_j corresponds to a q-variable and x_k to a p-variable. We distinguish three cases.
If $\pi(k) \leq i \leq n$, then $d(0, \pi(k)) = d(0, i) + d(i, \pi(k))$ holds. By means of the triangle inequality we obtain

$$\begin{aligned} d(i, \pi(j)) + d(i, \pi(k)) &= d(i, \pi(j)) + d(0, \pi(k)) - d(0, i) \\ &\leq d(0, \pi(j)) + d(0, \pi(k)), \end{aligned}$$

which means that $a_{ij} \geq a_{ik}$.
If $\pi(j) \leq i \leq \pi(k)$, then $d(i,\pi(j)) + d(i,\pi(k)) = d(\pi(i),\pi(k))$. Using the triangle-inequality we obtain

$$d(i,\pi(j)) + d(i,\pi(k)) = d(\pi(j),\pi(k)) \leq d(0,\pi(k)) + d(0,\pi(j)),$$

and consequently $a_{ij} \geq a_{ik}$.
If, however, $1 \leq i < \pi(j)$ holds, then we have $d(0,\pi(j)) = d(0,i) + d(i,\pi(j))$. By means of the triangle-inequality we get

$$\begin{aligned} d(0,\pi(j)) + d(0,\pi(k)) &= d(0,i) + d(i,\pi(j)) + d(0,\pi(k)) \\ &\geq d(i,\pi(k)) + d(i,\pi(j)), \end{aligned}$$

which shows that $a_{ij} \geq a_{ik}$.
This concludes the proof. ■

The optimal solutions of the feasible linear program (4) have a special structure as stated in the subsequent proposition.

Proposition 2.2 *If the linear program* (4) *is feasible, there exist indices r and s such that*

$$\begin{aligned} x_j^* &= \overline{x}_j && \text{for all } j = 0,1,\ldots,r-1, \\ x_r^* &\geq 0 && \\ x_j^* &= 0 && \text{for all } j = r+1,\ldots,n, \end{aligned} \tag{7}$$

and

$$\begin{aligned} y_j^* &= \overline{y}_j && \text{for all } j = 0,1,\ldots,s-1, \\ y_s^* &\geq 0 && \\ y_j^* &= 0 && \text{for all } j = s+1,\ldots,n, \end{aligned} \tag{8}$$

is an optimal solution of problem (4)*. Moreover, we can always assume that either x_r^* or y_s^* is strictly positive.*

Proof. Let $(\hat{x}_j, \hat{y}_j)$, $j = 0,1,\ldots,n$, be an optimal solution of the linear program (4). We assume that $\hat{x}_r$ is the first component smaller than its upper bound and that there is an index l, $r+1 \leq l \leq n$, with $\hat{x}_l > 0$. According to Proposition 2.1 we have $a_{ir} \geq a_{il}$ for all $i = 1,\ldots,n$. So, increasing $\hat{x}_r$ to $\hat{x}_r + \epsilon$ while decreasing $\hat{x}_l$ to $\hat{x}_l - \epsilon$ maintains the feasibility and does not change the objective function value. We can choose ϵ as large as possible until either $\hat{x}_r + \epsilon = \bar{x}_r$ or $\hat{x}_l - \epsilon = 0$ is met first. Proceeding iteratively like this we obtain an optimal solution which fulfills (7). Starting from this solution, we can apply the same arguments to $\hat{y}$. ■

If a vertex weight is increased, it does not make sense to decrease it at the same time, since this will only enlarge the costs. So every optimal solution (x^*, y^*) of (4) fulfills an orthogonality relation:

Proposition 2.3 *For every optimal solution* (x_j^*, y_j^*), $j = 0, 1, \ldots, n$, *of the linear program* (4),

$$x_j^* \, y_{n-j}^* = 0, \quad j = 0, 1, \ldots, n, \tag{9}$$

must hold.

In combination with Proposition 2.2 and the fact that either x_r^* or y_s^* is strictly positive, we immediately get

Corollary 2.4 *An optimal solution of the form stated in Proposition 2.2 always fulfills*

$$r + s \le n.$$

3 Reformulation as Linear Program in Two Variables

For $\xi \ge 0$, $\eta \ge 0$ we define piecewise linear functions $g_i(\xi)$ and $\bar{g}_i(\eta)$, $i = 1, \ldots, n$, as follows.

$$\begin{aligned} g_i(\xi) &:= \sum_{j=0}^{r-1} a_{ij}\bar{x}_j + a_{ir}(\xi - \sum_{j=0}^{r-1} \bar{x}_j) \quad \text{for} \quad \sum_{j=0}^{r-1} \bar{x}_j \le \xi \le \sum_{j=0}^{r} \bar{x}_j, \\ \bar{g}_i(\eta) &:= \sum_{k=0}^{s-1} \bar{a}_{ik}\bar{y}_k + \bar{a}_{is}(\eta - \sum_{k=0}^{s-1} \bar{y}_k) \quad \text{for} \quad \sum_{k=0}^{s-1} \bar{y}_k \le \eta \le \sum_{k=0}^{s} \bar{y}_k. \end{aligned}$$

Proposition 2.2 implies that by setting

$$\xi := \sum_{j=0}^{n} x_j \quad \text{and} \quad \eta := \sum_{k=0}^{n} y_k$$

we can write the linear program (4) as the following problem in two variables which has the same set of optimal solutions:

$$\begin{aligned} \text{minimize} \quad & \xi + \eta \\ \text{s. t.} \quad & g_i(\xi) + \bar{g}_i(\eta) \ge b_i, \quad i = 1, \ldots, n, \\ & 0 \le \xi \le \sum_{j=0}^{n} \bar{x}_j, \\ & 0 \le \eta \le \sum_{k=0}^{n} \bar{y}_k. \end{aligned} \tag{10}$$

Due to Proposition 2.1 the functions $g_i(\xi)$ and $\bar{g}_i(\eta)$ are concave for each $i = 1, 2, ..., n$ as their slopes are monotonically decreasing. As the sum of concave functions is again concave and the set $\{x : g(x) \ge b\}$ is convex for any concave function g we get

Lemma 3.1 *The set* B *defined by*

$$B := \big\{(\xi, \eta) \mid g_i(\xi) + \bar{g}_i(\eta) \ge b_i, \ 1 \le i \le n\big\}$$

is convex.

The next lemma is essential for estimating of the running time.

Lemma 3.2 *The boundary of B intersects at most $4(n+1)$ boxes*

$$B(r,s) := \left\{ (\xi,\eta) : \sum_{j=0}^{r-1} \bar{x}_j \le \xi \le \sum_{j=0}^{r} \bar{x}_j,\ \sum_{k=0}^{s-1} \bar{y}_k \le \eta \le \sum_{k=0}^{s} \bar{y}_k \right\}. \tag{11}$$

Proof. For fixed η, we have $n+1$ boxes in ξ-direction; for fixed ξ we have $n+1$ boxes in η-direction. The convexity of the set B immediately implies that when going in one direction around the boundary one changes the box in ξ-direction at most $2(n+1)$ and in η-direction at most $2(n+1)$ times. That is, the boundary intersects at most $4(n+1)$ boxes. ■

Every point (x_r, y_s) in $B(r,s)$ corresponds in a unique way to a solution of the form described in Proposition 2.2, namely by setting $x_j := \overline{x}_j$ for all $0 \le j \le r-1$ and $x_j := 0$ for $j \ge r+1$. Similarly, $y_j = \overline{y}_j$ for all $0 \le j \le s-1$ and $y_j := 0$ for $j \ge s+1$. Vice versa, for every point of the form described in Proposition 2.2 there are indices r and s such that this point uniquely corresponds to a point in $B(r,s)$.

Since $g_i(\xi)$ and $\bar{g}_i(\eta)$ are concave functions, $g_i(\xi)+\bar{g}_i(\eta) \ge b_i$ can for every $i = 1,\dots,n$, be replaced by the set of constraints

$$\sum_{j=0}^{r-1} a_{ij}\overline{x}_j + a_{ir}(\xi - \sum_{j=0}^{r-1} \overline{x}_j) + \sum_{k=0}^{s-1} \bar{a}_{ik}\overline{y}_k + \bar{a}_{is}(\eta - \sum_{k=0}^{s-1} \overline{y}_k) \ge b_i,$$
$$r = 0,1,\dots,n, \quad s = 0,1,\dots,n,$$

that is obtained by combining

$$l_{ir}(\xi) := \sum_{j=0}^{r-1} a_{ij}\overline{x}_j + a_{ir}(\xi - \sum_{j=0}^{r-1} \overline{x}_j)$$

for each $r = 0,1,\dots,n$ with

$$\bar{l}_{is}(\eta) := \sum_{k=0}^{s-1} \bar{a}_{ik}\overline{y}_k + \bar{a}_{is}(\eta - \sum_{k=0}^{s-1} \overline{y}_k)$$

for each $s = 0,1,\dots,m$.

This yields a linear program (LP1) in two variables with $O(n^3)$ constraints, as $m \le n$.

4 Global Solution Method

Since there are only two variables in the linear program (LP1), it can be solved in $O(n^3)$ time with Megiddo's algorithm (see Megiddo [7]). We call this a *global solution approach*, since all constraints defining the set B of feasible solutions are considered at the same time. This global approach can be refined: instead of considering all n^3 constraints, we can improve on the complexity if we find for every $i = 1, 2, ..., n$ those linear functions $l_{ir}(\xi) + \bar{l}_{is}(\eta)$ which define the boundary of

$$B_i := \big\{(\xi, \eta) \mid g_i(\xi) + \bar{g}_i(\eta) \geq b_i\big\}.$$

Lemma 3.2 applied to B_i instead of B combined with the fact that the boundary of B_i in a box $B(r, s)$ is given by at most one inequality $l_{ir}(\xi) + \bar{l}_{is}(\eta) \geq b_i$ shows that B_i can be written as intersection of at most $O(n)$ halfplanes:

$$B_i := \big\{(\xi, \eta) \mid l_{ir}(\xi) + \bar{l}_{is}(\eta) \geq b_i, (r, s) \in \mathcal{C}_i\big\},$$

where

$$\mathcal{C}_i := \big\{(r, s) \mid l_{ir}(\xi) + \bar{l}_{is}(\eta) = b_i\big\},$$

As the feasible set B is the intersection of the sets B_i, B can be described by the $O(n^2)$ inequalities of

$$\mathcal{C} := \bigcup_{i=1,\ldots,n} \mathcal{C}_i.$$

So, (LP1) can be written as a linear program with two variables and $O(n^2)$ constraints which can be solved in time $O(n^2)$.

Variables with an upper bound 0 can be fixed beforehand and are deleted during the following solution process. This means, we can assume in the following that every box $B(r, s)$ is non degenerated. After fixing some variables we are left with, say, n free variables x_j and m free variables y_k, $m \leq n$.

The determination of the sets $\mathcal{C}_i$ is a task of elementary geometry and can be performed in $O(n)$ steps. First, the points with the largest and the smallest η-value, respectively, that lie on the boundary are determined. Since $g_i(\xi)$ and $\bar{g}_i(\eta)$ are both concave, these points can be found by calculating the value $\hat{\xi}$ for which the function $g_i(\xi)$ is maximized and the corresponding η-values. Since the functions are piecewise linear, $\hat{\xi}$ can be found in linear time. Starting from these points, we compute the intersection points of the boundary with the lines $\xi = \xi_r$ and $\eta = \eta_s$, where

$$\xi_r := \sum_{j=0}^{r} \overline{x}_j, \quad r = -1, 0, 1, \ldots, n$$

and

$$\eta_s := \sum_{k=0}^{s} \overline{y}_k, \quad s = -1, 0, 1, \ldots, m,$$

by going from $\hat{\xi}$ right and then left until no intersection point, just one intersection point or infinitely many intersection points are found. We will use the fact that every box $B(r, s)$ contains at most one linear part of the boundary. Finally, we determine the halfplanes defining the feasible set by connecting all intersection points. For a detailed description see Pleschiutschnig [8].

The calculation of all intersection points to the right and to the left side of $\hat{\xi}$ needs $O(n)$ time, since the boundary of B_i is convex and meets according to Lemma 3.2 at most $O(n)$ boxes. Thus the set of constraints $\mathcal{C}_i$ can be found in $O(n)$ time.

As there are n inequalities to be considered, the overall time required for calculating all sets $\mathcal{C}_i$, $i = 1, \ldots, n$, is $O(n^2)$. Thus the linear program (LP1) has two variables and $O(n^2)$ constraints. Therefore it can be solved in $O(n^2)$ time.

Altogether, we get

Theorem 4.1 *The inverse 1-median problem on cycles with nonnegative weights, positive edge-lengths and uniform cost can be solved in $O(n^2)$ time.*

5 Iterative Solution Method

Whereas the two main components of the global solution method, the determination of the set $\mathcal{C}$ and Megiddo's method, lead to a rather involved algorithm, there exists a very simple, iterative, method for solving LP's of the form (4). Computational tests strongly suggest (see Table 1) that the worst case complexity of the iterative method is also $O(n^2)$ time. We were not able to show this, as we cannot bound the number of local problems until a feasible solution is found. But $O(n^3)$ time is certainly an upper bound for the iterative method.

A straightforward idea of finding an optimal solution of the (nonlinear) program (10) is to solve the *local* LPs

$$\begin{aligned} \text{minimize} \quad & x_r + y_s \\ \text{s. t.} \quad & a^r x_r + \bar{a}^s y_s \geq b - \sum_{j=0}^{r-1} \overline{x}_j a^j - \sum_{k=0}^{s-1} \overline{y}_k \bar{a}^k, \\ & 0 \leq x_r \leq \overline{x}_r, \\ & 0 \leq y_s \leq \overline{y}_s \end{aligned}$$

in each box $B(r, s)$, $r = 0, 1, \ldots, n$, $s = 0, 1, \ldots, m$ and to compare the objective function values of feasible solutions. Since in each one of the $O(n^2)$ boxes the optimal solution of a linear program with two variables and $O(n)$ constraints has to be determined, this method runs in $O(n^3)$ time.

We can speed up this basic method in the following way. First, we determine a feasible solution or prove that the problem is infeasible. In the second phase we start from a feasible point and compute an optimal solution. Thus the solution process consists of two phases.

Let us first assume that a feasible solution of (4) is already known. For finding an optimal solution, we start from a feasible solution with objective function value z. Let us assume that this feasible solution corresponds – in the way described above – to a point

(x_r, y_s) in $B(r,s)$, where we assume that $|x_r| + |y_s| > 0$. We solve the local linear program (LP(r,s))

$$\begin{aligned} \text{minimize} \quad & x_r + y_s \\ \text{s. t.} \quad & a^r x_r + \bar{a}^s y_s \geq b - \sum_{j=0}^{r-1} \overline{x}_j a^j - \sum_{k=0}^{s-1} \overline{y}_k \bar{a}^k, \\ & 0 \leq x_r \leq \overline{x}_r, \\ & 0 \leq y_s \leq \overline{y}_s. \end{aligned} \tag{LP(r,s)}$$

Due to the convexity of B we immediately get:

Lemma 5.1 *If the optimal solution x_r^*, y_s^* of (LP(r,s)) fulfills one of the following three conditions*

$$0 < x_r^* < \overline{x}_r \quad \textit{and} \quad 0 < y_s^* < \overline{y}_s, \tag{12}$$

$$(r = 0 \quad \textit{and} \quad x_0^* = 0) \quad \textit{or} \quad (r = n \quad \textit{and} \quad x_n^* = \overline{x}_n), \tag{13}$$

or

$$(s = 0 \quad \textit{and} \quad y_0^* = 0) \quad \textit{or} \quad (s = m \quad \textit{and} \quad y_m^* = \overline{y}_m), \tag{14}$$

then the corresponding solution of (4) *is optimal.*

In the case that the optimal solution of (LP(r,s)) does not fulfill one of the conditions (12)-(14), we have to solve one, two or three additional local LPs in order to meet a decision. We distinguish eight cases:

- The optimum is attained in the lower left corner of the box $B(r,s)$. In this case we solve the linear programs in the boxes $B(r-1,s)$, $B(r-1,s-1)$ and $B(r,s-1)$, provided the corresponding boxes are non empty.

- The optimum is attained in an inner point of the left side of box $B(r,s)$ and $r > 0$. Then we have to check whether there is not a better solution in the box $B(r-1,s)$.

- The optimum is attained in an inner point of the right side of box $B(r,s)$ with $r < n$. Then we have to check whether there is not a better solution in the box $B(r+1,s)$.

- The optimum is attained in an inner point of the upper side of box $B(r,s)$ and $s < m$. Then we have to check whether there is not a better solution in the box $B(r,s+1)$.

- The optimum is attained in an inner point of the lower side of box $B(r,s)$ and $s > 0$. Then we have to check whether there is not a better solution in the box $B(r,s-1)$.

- The optimum is attained in the upper left corner of the box $B(r,s)$ with $r > 0, s < m$. Then due to the convexity of set B we have to check whether there is not a better solution in the box $B(r-1,s+1)$.

- The optimum is attained in the lower right corner of the box $B(r, s)$ with $r > 0, s > 0$. Then due to the convexity of set B we have to check whether there is not a better solution in the box $B(r + 1, s - 1)$.
- The optimum is attained in the upper right corner of the box $B(r, s)$ with $r < n, s < m$. In this case we have to check two new boxes, namely the box $B(r, s + 1)$ and the box $B(r + 1, s)$.

Let z_1, z_2 and z_3 be the objective function values of the solutions of (4) which correspond to the optimal solutions of the currently first, second or third local LP. (If there is only one new local LP, we set $z_2 := z_3 := \infty$, and analogously, in the case of only two local LPs, $z_3 := \infty$). If

$$\min(z_1, z_2, z_3) = z,$$

then the solution corresponding to z is optimal. If $\min(z_1, z_2, z_3) < z$ and the minimum is attained by two values, say z_1 and z_2, then the solution with objective function value z_1 is optimal. Otherwise we replace the previous feasible point by the solution with minimum objective function value z_1, z_2 or z_3 and proceed from this point. First, we check whether Lemma 5.1 can be applied. Otherwise, we again solve new local LPs according the list above.

Lemma 5.2 *Starting from a feasible solution, an optimal solution is found in $O(n^2)$ time.*

Proof. Every local linear program is an LP with two variables and n constraints and can, therefore, be solved in $O(n)$ time. Whenever the solution of a local LP is attained in the lower left corner and $r > 0, s > 0$, we decrease r and s. Thus, this situation occurs only $O(n)$ times. On the other hand, due to Lemma 3.2 the boundary of B intersects at most $O(n)$ boxes. This means that once a boundary point is found, will find the optimal solution by solving at most $O(n)$ local LPs. ∎

Now, we address the question how a first feasible point can be found, provided that the problem is feasible. We proceed in a similar way as above and solve, starting from $r := s := 0$ a series of local LPs with different objective functions. If $x_o = 0,\ y_0 = 0$ is infeasible, the right-hand side vector b has at least one positive component, say, b_i. In this case, we choose the i-th row of the coefficient matrix as new objective function and we solve the local LP

$$\begin{array}{rl} \text{maximize} & a_{i0}x_0 + \bar{a}_{i0}y_0 \\ \text{s. t.} & a^0 x_0 + \bar{a}^0 y_0 \geq \bar{b}, \\ & 0 \leq x_0 \leq \overline{x}_0, \\ & 0 \leq y_0 \leq \overline{y}_0, \end{array} \tag{15}$$

where $\bar{b}$ is defined by

$$\bar{b}_i := \begin{cases} 0 & \text{if } b_i \geq 0, \\ b_i & \text{if } b_i < 0. \end{cases}$$

This definition ensures that all already feasible inequalities stay feasible. Let (x_0^*, y_0^*) be an optimal solution of this local LP. We replace the right hand side b by $\hat{b} := b - x_0^* a^0 - y_0^* \bar{a}_0$. If $\hat{b} \leq 0$, the current point is feasible for the LP (4) and we continue with Phase 2. Otherwise there are two possibilities. If the current point fulfills the i-th constraint we continue from this point with a new objective function (which stems from a component $\hat{b}_i > 0$). Otherwise we keep the old objective and solve a new local LP starting from the current point.

Lemma 5.3 *If the inverse 1-median problem on a cycle is feasible, then the algorithm stated above determines a feasible point.*

Proof. By definition of the objective function of the local linear programs, the search direction coincides with the gradient of the actually considered inequality. The constraints of the local linear programs guarantee that we stay within the feasible set of the inequalities which are already met and that the right-hand side of not yet fulfilled inequalities does not increase. Thus, if there are inequalities which are not met but there is no search direction such that the problem is improved, i.e., if all current local LPs have a non-positive objective function value or there are no more local LPs according to the eight cases stated above that can be considered, then there is no feasible solution. ■

The computational behaviour of the algorithm leads to the conjecture that in fact only $O(n)$ local linear programs have to be considered until a feasible solution is found. We run this method on more than 15000 randomly generated test instances with 10 to 500 vertices, see Table 1. For each instance the weights, upper and lower bounds of the weights and the edge lengths were randomly chosen inbetween 0 and *maxval*. For every problem set, *num* test instances have been solved. Table 1 shows in column $avLPfeas$ the average number of local linear programs solved until a feasible solution is found or the infeasibility of the problem is proven. Column $avLPopt$ shows the average number of local linear programs solved until, starting from a feasible solution, an optimal solution is found. The maximum number of linear programs solved until a feasible solution is determined or the infeasibility of the problem is seen is shown in column $maxLPfeas$. The maximum number of local linear programs solved until, starting from a feasible solution, an optimal solution is found is shown in column $maxLPopt$.

This table shows that for finding a feasible solution never more than $O(n)$ local linear programs have been solved. On average, this number is even very small. This is a strong hint that the iterative method runs in $O(n^2)$ time. Moreover, these results show that in practice the inverse 1-median problem on a cycle can be solved in a very fast and simple manner by the iterative approach.

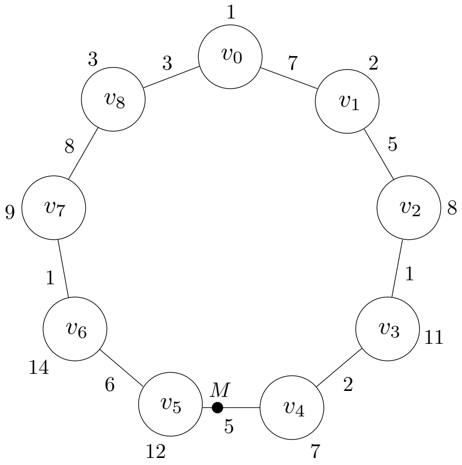
6 An Example

The following example shall illustrate the reformulations of the inverse 1-median problem on a cycle and illustrate the iterative method. We consider a cycle with the 9 vertices

$n+1$	$maxval$	num	$avLPfeas$	$avLPopt$	$maxLPfeas$	$maxLPopt$
10	10	2000	4.7	1.6	26	6
10	100	2000	4.9	1.5	31	5
20	20	2000	8.2	1.9	50	8
20	400	2000	8.0	1.9	50	7
50	50	2000	14.5	2.3	80	11
50	2500	2000	14.6	2.3	78	10
100	100	2000	21.8	2.7	104	12
100	10000	2000	22.9	2.7	105	12
200	200	500	26.6	2.7	123	14
500	500	200	51.2	3.9	198	20

Table 1: Number of local linear programs solved by the iterative method

$v_0, v_1, \ldots, v_8$, where v_0 should become the 1-median. We assume that the cycle has the following edge lengths l_j, vertex weights w_j as well as lower and upper bounds for the weights:



j	w_j	l_j	$\underline{w}_j$	$\overline{w}_j$
0	1	7	0	10
1	2	5	0	5
2	8	1	0	10
3	11	2	7	15
4	7	5	7	10
5	12	6	7	15
6	14	1	7	15
7	9	8	4	10
8	3	3	0	10

Figure 1: Cycle with 9 vertices

By ordering $\hat{d}$ increasingly, we get

$$x_0 := p_0,\ x_1 := p_8,\ x_2 := q_4,\ x_3 := q_3,\ x_4 := q_2,\\ x_5 := p_7,\ x_6 := p_6,\ x_7 := q_1,\ x_8 := p_5$$

and

$$y_0 := q_5,\ y_1 := p_1,\ y_2 := q_6,\ y_3 := q_7,\ y_4 := p_2,\\ y_5 := p_3,\ y_6 := p_4,\ y_7 := q_8,\ y_8 := q_0.$$

Thus, we obtain the subsequent coefficient matrix A of the linear program (4)

$$A = \begin{pmatrix} 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & -5 \\ 12 & 12 & 12 & 12 & 12 & 4 & 2 & 2 & -10 \\ 13 & 13 & 13 & 13 & 11 & 3 & 1 & 1 & -11 \\ 15 & 15 & 15 & 11 & 9 & 1 & -1 & -1 & -13 \\ 18 & 12 & 10 & 6 & 4 & -4 & -6 & -6 & -18 \\ 12 & 6 & 4 & 0 & -2 & -10 & -12 & -12 & -12 \\ 11 & 5 & 3 & -1 & -3 & -11 & -11 & -11 & -11 \\ 3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \end{pmatrix}$$

The set of feasible solutions is shown in Figure 2.

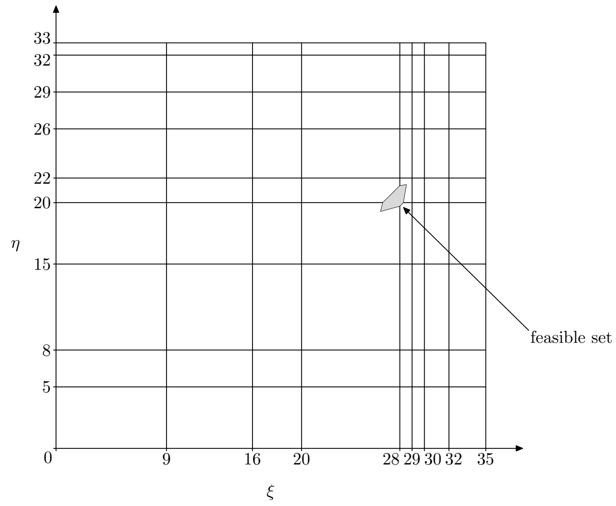


Figure 2: Feasible set of the example

Let us now apply the iterative method to solve this instance. First, we determine a feasible solution. At the beginning, $b = (67, 324, 363, 397, 438, 360, 323, 27)$. The component b_1 is positive, so we try to find a point which is feasible for the first inequality. We solve the local linear program with the corresponding objective function in the box $B(0, 0)$ and obtain as optimal solution $x_0^* = 9$, $y_0^* = 5$.

After that, $b = (-21, 166, 191, 197, 186, 192, 169, -15)$. We keep the first inequality feasible and try to find a point which is also feasible for the second inequality. The corresponding local LP in this box $B(0, 0)$ yields the point $(9, 5)$. So we turn to the boxes $B(0, 1)$, $B(1, 1)$ and $B(1, 0)$. The maximal objective value is attained in the box $B(1, 1)$ with $x_1^* = 7$, $y_1^* = 2$ and yields $b = (-56, 76, 90, 78, 78, 114, 102, 0)$. Thus we have still to consider the second constraint. As the optimum was attained in an inner point of the right side of box $B(1, 1)$ and the box $B(2, 1)$ has been deleted since $\overline{x}_2 = 0$, we check the box $B(3, 1)$ next. The optimal solution is $x_3^* = 1$, $y_1^* = 3$. We get $b = (-56, 76, 90, 78, 78, 114, 102, 0)$ which shows that the second inequality still is not met in the corresponding point. Since the optimum was attained in an inner point of the upper side of box $B(3,1)$, we consider the box $B(3,2)$ next and obtain as optimal solution $x_3^* = 4$, $y_2^* = 3$ with $b = (-56, 46, 54, 42, 42, 78, 72, 0)$. The last optimum was attained in an inner point of the right side of box $B(3,2)$, so we consider the box $B(4,2)$ next and obtain as locally optimal solution $x_4^* = 4$, $y_2^* = 7$ with $b = (-56, 6, 14, 2, 2, 38, 40, 0)$. As the optimal solution was attained in an inner point of the upper side of box $B(4,2)$, we check box $B(4,3)$ next and obtain $x_4^* = 8$,$y_3^* = 4$ with $b = (-56, -26, -18, -30, -30, 6, 8, 0)$. Now the first five inequalities are met. Thus the sixth inequality defines the new objective function. We consider the box $B(4,3)$ once again and obtain as optimal solution $x_4^* = 6.6364$, $y_3^* = 5$ with $b = (-39.4545, -5.6364, 0, -16.7273, -28.5455, -6.7273,\ -7.0909, -7.0909)$. Thus, a feasible solution has been found.

We start now the second phase and determine an optimal solution of the original given problem. For this, we resolve a local linear program in the actual box $B(4,3)$ and obtain $x_4^* = 6.4464$, $y_3^* = 4.3036$. Since the locally optimal solution is obtained in an inner point of the actual box, an optimal solution with objective function value 45.75 has been found.

7 Conclusions

We have shown that the inverse 1-median problem on a cycle can be transformed to a linear program of the form

$$\begin{aligned} \text{minimize} \quad & \sum_{j=0}^{n} x_j + \sum_{j=0}^{n} y_j \\ \text{s. t.} \quad & Ax + \bar{A}y \geq b, \\ & 0 \leq x \leq \overline{x}, \\ & 0 \leq y \leq \overline{y}. \end{aligned} \tag{16}$$

where the columns of the matrices A and $\bar{A}$ are both monotonically decreasing. As the results of Section 3 apply to any linear program of the form stated above, such LPs can be transformed to a linear program with two variables and thus be solved in a fast manner. We describe two solution methods, a global approach for which a low time complexity is established, and a computational fast iterative approach for which we were not able to prove the same time complexity. Computational tests, however, strongly support that the iterative method has the same worst case behaviour as the global method. To show this is an interesting task to be addressed in the future.

References

[1] Berman O., D.I. Ingco and A.R. Odoni, Improving the location of minisum facilities through network modification. *Annals of Operations Research* **40**, 1992, 1-16.

[2] Burkard, R.E., E. Gassner and J. Hatzl, A linear time algorithm for the reverse 1-median problem on a cycle. To appear in *Networks* 2006.

[3] Burkard, R.E., C. Pleschiutschnig and J.Z. Zhang, Inverse p-median problems. *J. Discrete Optimization* **1**, 2004, 23–39.

[4] Burton, D. and Ph.L. Toint, On an instance of the inverse shortest path problem. *Mathematical Programming* **53**, 1992, 45-61.

[5] Cai, M.C., X.G. Yang and J.Z. Zhang, The complexity analysis of the inverse center location problem. *Journal of Global Optimization* **15**, 1999, 213-218.

[6] Heuberger, C., Inverse optimization: a survey on problems, methods, and results. *J. Combinatorial Optimization*, **8** 2004, 329–361.

[7] Megiddo, N., Linear-time Algorithms for Linear Programming in $\mathbb{R}^3$ and Related Problems, *SIAM J. Comput.* **12** 1983, 759–776.

[8] Pleschiutschnig, C., Inverse median problems. PhD thesis, Institute of Mathematics B, Graz University of Technology, Graz (Austria), 2005.

[9] Zhang, J., Z. Liu and Z. Ma, Some reverse location problems. *Europ. J. of Oper. Research* **124**, 2000, 77–88.