Comput. Optimiz. and Appl. **21**(2002), 213–229

THE NEWTON BRACKETING METHOD FOR CONVEX MINIMIZATION

YURI LEVIN AND ADI BEN-ISRAEL

Abstract. An iterative method for the minimization of convex functions $f : \mathbb{R}^n \to \mathbb{R}$, called a Newton Bracketing (NB) method, is presented. The NB method proceeds by using Newton iterations to improve upper and lower bounds on the minimum value. The NB method is valid for $n = 1$, and in some cases for $n > 1$ (sufficient conditions given here). The NB method is applied to large scale Fermat–Weber location problems.

1. Introduction

We introduce an iterative method, called the *Newton Bracketing* (*NB* for short) method, for the minimization of convex functions $f : \mathbb{R}^n \to \mathbb{R}$ with attained infimum $f_{\min}$. The Newton Bracketing approach is based on the following geometric facts:

- a Newton iteration uses the tangent T^k of f at the current iterate $\mathbf{x}^k$,
- the graph of a convex function lies above its tangents.

The method works by improving bounds on the minimum $f_{\min}$, rather than approximating a solution $\mathbf{x}^*$ satisfying the optimality condition $\mathbf{0} \in \partial f(\mathbf{x}^*)$.

An iteration begins with an interval $[L^k, U^k]$, called a *bracket*, containing the minimum value $f_{\min}$,

$$L^k \le f_{\min} \le U^k \ . \tag{1}$$

The upper bound is $U^k := f(\mathbf{x}^k)$ where $\mathbf{x}^k$ is the current iterate, an initial lower bound L^0 is assumed known. If the bracket is sufficiently small, say

$$U^k - L^k < \epsilon \tag{2}$$

then the current $\mathbf{x}^k$ is declared optimal, and computations stop. Else, select a value M^k inside the bracket, i.e.

$$M^k := \alpha\, U^k \ + \ (1-\alpha)\, L^k \ , \ \text{for some } 0 < \alpha < 1, \tag{3}$$

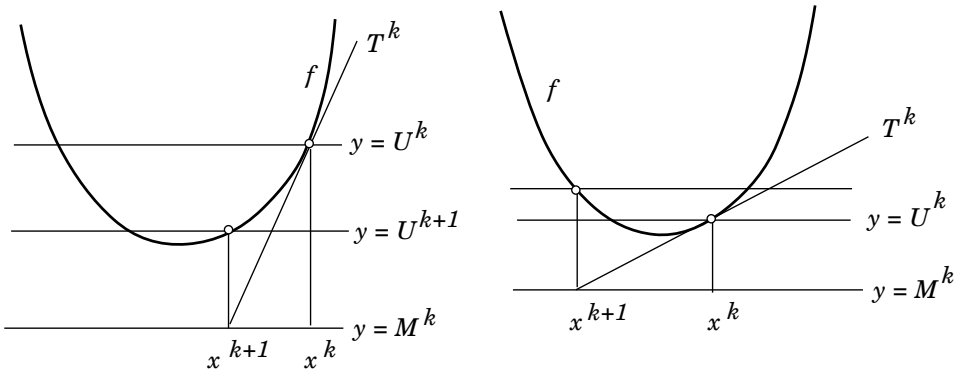
and do one Newton iteration for solving

$$f(\mathbf{x}) = M^k \ , \tag{4}$$

Date: September 14, 1999.

1991 *Mathematics Subject Classification.* Primary 65K05, 90C25; Secondary 90C06, 49M15, 90B85.

Key words and phrases. Directional Newton method, convex functions, unconstrained minimization, Fermat–Weber location problem.



(a) Case 1: $f(x^{k+1}) < f(x^k)$ (b) Case 2: $f(x^{k+1}) \geq f(x^k)$

FIGURE 1. An illustration of the Newton Bracketing Method

to get $\mathbf{x}^{k+1}$. The new value $f(\mathbf{x}^{k+1})$ then allows narrowing the bracket $[L^k, U^k]$, either by lowering U^k or by raising L^k, as follows

$$\text{Case 1: if } f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k) \text{ then } U^{k+1} := f(\mathbf{x}^{k+1}) \,, \tag{5a}$$

$$\text{Case 2: if } f(\mathbf{x}^{k+1}) \geq f(\mathbf{x}^k) \text{ then } L^{k+1} := M^k \,,\ \mathbf{x}^{k+1} := \mathbf{x}^k \,, \tag{5b}$$

see Figure 1. An iteration is said to be of *Type 1* or *Type 2*, corresponding to Case 1 or Case 2, respectively. In either case the bracket is reduced, the *reduction ratio* is

$$\frac{U^{k+1} - L^{k+1}}{U^k - L^k} = \begin{cases} \dfrac{f(x^{k+1}) - L^k}{f(x^k) - L^k} & \text{in Case 1 ,} \\ 1 - \alpha & \text{in Case 2 ,} \end{cases} \tag{6}$$

where M^k is selected by (3).

The NB method is *valid* for minimizing f if every iteration produces a bracket, i.e. if (1) holds for all k. To prove validity it suffices to show that the lower bound L^{k+1} in (5b) is correct (the update in (5a) is clearly valid).

Differentiability of f is not an issue, in as much as a tangent T^k (i.e. a supporting hyperplane of the epigraph of f at $(\mathbf{x}^k, f(\mathbf{x}^k))$) can be constructed using any subgradient of f at $\mathbf{x}^k$. For simplicity, we assume throughout that the function f is differentiable.

The NB method for minimizing convex functions of one variable is studied in § 3, its validity is straightforward, see Theorem 1.

Convex functions $f : \mathbb{R}^n \to \mathbb{R}$, $n > 1$, are studied in § 5. In this case we must first define a *Newton step* for solving (4): we select a step of the gradient directional Newton method [21],

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \frac{f(\mathbf{x}^k) - M^k}{\|\nabla f(\mathbf{x})\|^2} \nabla f(\mathbf{x}) \,, \tag{7}$$

and the algorithm is valid provided the level sets of f are not too narrow, see precise statements in Theorems 2–5.

In § 6 we apply the Algorithm to the Fermat–Weber location problem. Numerical experiments with random data show that the brackets $[L, U]$ are on the average halved at each iteration, so that after 10 iterations the bracket has been reduced to about 0.001 of its initial size. These performance figures are independent of the problem size $N \times n$ where N is the number of points in $\mathbb{R}^n$. In § 7 we report numerical experiments comparing our method with the Weiszfeld Method, the standard method for solving the location problem.

2. The Newton Bracketing (NB) method

Consider the problem

$$\inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{8}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, differentiable and bounded below, and the infimum is attained. If the value of (8) is known, say $f_{\min}$, then a minimizing $\mathbf{x}$ can be found by solving the equation

$$f(\mathbf{x}) = f_{\min} \ , \tag{9}$$

using the Newton method. In general $f_{\min}$ is unknown, however a lower bound $L^0 < f_{\min}$ may be known. The kth iteration begins with:

a *current iterate* $\mathbf{x}^k$, and

a *bracket* $[L^k, U^k]$, i.e. an interval containing the value of (8). At the initial iteration the upper bound is taken as

$$U^0 := f(\mathbf{x}^0)$$

where $\mathbf{x}^0$ is the initial itereate. A description of the general iteration for the case $n = 1$ follows.

Algorithm 1 (The NB method for $n = 1$)**.**

1	**if**	$U^k - L^k < \epsilon$	**then solution** $:= x^k$, **stop**
2	**endif**	**select**	$M^k := \alpha\, U^k + (1-\alpha)\, L^k$
3		**do**	$x^{k+1} := x^k - \dfrac{f(x^k) - M^k}{f'(x^k)}$
4	**if**	$f(x^{k+1}) < f(x^k)$	**set** $U^{k+1} := f(x^{k+1})$, $L^{k+1} := L^k$
5		**else**	**set** $L^{k+1} := M^k$, $U^{k+1} := U^k$, $x^{k+1} := x^k$
	endif	$k := k + 1$	**return**

Notes.

• Line 3 is a Newton iteration for the equation (4) where the value M^k is selected in line 2 for some $0 < \alpha < 1$. Typical choices of α are:

$$\alpha \ = \ 0.5 \quad \text{(bisection) , and} \tag{10a}$$

$$\alpha \ = \ \frac{\sqrt{5}-1}{2} \approx 0.61803 \quad \text{(golden section) .} \tag{10b}$$

• Lines 4 and 5 represent the two possible cases (5a) and (5b) for the value of $f(x^{k+1})$. Case 2 occurs if the Newton iteration gives a point x^{k+1} on the "other side" of the minimum, and sufficiently far to get $f(x^{k+1}) \geq f(x^k)$, see e.g. Figure 1(b).
• In iterations of Type 2 the point x^k is unchanged, and therefore there is no need to recompute $f(x)$ and $f'(x)$. This is a considerable saving, since iterations of Type 2 occur often (in our experience, about 75% of the time).
• Line 1 is a stopping rule: If (2) holds for a given tolerance ϵ, the current iterate x^k is declared optimal. This can be used in conjunction with other stopping rules, such as $|f'(x^k)| < \epsilon$, $|x^{k+1} - x^k| < \epsilon$ or an upper bound on the number of iterations.
• Algorithm 1 generates iterates with nonincreasing values of f,

$$f(x^0) \geq \cdots \geq f(x^k) \geq f(x^{k+1}) \geq \cdots \tag{11}$$

with equality $f(x^k) = f(x^{k+1})$ iff iteration k is of type 2. □

For functions $f : \mathbb{R}^n \to \mathbb{R}$, it is not a priori clear which Newton method to use in Line 3. A directional Newton method can be described as

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \frac{f(\mathbf{x}^k)}{\nabla f(\mathbf{x}^k) \cdot \mathbf{d}^k}\, \mathbf{d}^k \ , \quad k = 0, 1, \ldots \tag{12}$$

where $\mathbf{d}^k$ is the direction at the k th iteration. This method, studied in [21], converges at a quadratic rate to a solution of $f(\mathbf{x}) = 0$, under standard assumptions on f, $\mathbf{x}^0$, and the directions $\{\mathbf{d}^k\}$. In particular, for $\mathbf{d}^k$ along the gradient $\nabla f(\mathbf{x}^k)$, (12) becomes

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \frac{f(\mathbf{x}^k)}{\| \nabla f(\mathbf{x}^k) \|^2}\, \nabla f(\mathbf{x}^k) \ , \ k = 0, 1, \ldots \tag{13}$$

The general iteration of the NB method is then described as follows:

Algorithm 2 (The NB method using a gradient directional Newton step)**.**

1	**if**	$U^k - L^k < \epsilon$	**then solution** $:= \mathbf{x}^k$, **stop**
2	**endif**	**select**	$M^k := \alpha\, U^k + (1 - \alpha)\, L^k$
3		**do**	$\mathbf{x}^{k+1} := \mathbf{x}^k - \dfrac{f(\mathbf{x}^k) - M^k}{\|\nabla f(\mathbf{x}^k)\|^2}\, \nabla f(\mathbf{x}^k)$
4	**if**	$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$	**set** $U^{k+1} := f(\mathbf{x}^{k+1})$, $L^{k+1} := L^k$
5		**else**	**set** $L^{k+1} := M^k$, $U^{k+1} := U^k$, $\mathbf{x}^{k+1} := \mathbf{x}^k$
	endif	$k := k + 1$	**return**

Notes (in addition to the notes after Algorithm 1).
• Algorithm 2 is not valid in general for convex functions $f : \mathbb{R}^n \to \mathbb{R}$, even if a directional Newton method should work for such f. The problem is in Case 2 (Line 5 in Algorithm 2), where $M^k > f_{\min}$ is possible, so M^k is not a valid lower bound on $f_{\min}$. This is apparent for functions f with long and narrow level sets.
• It is possible to avoid Case 2, by raising M^k and repeating until Case 1

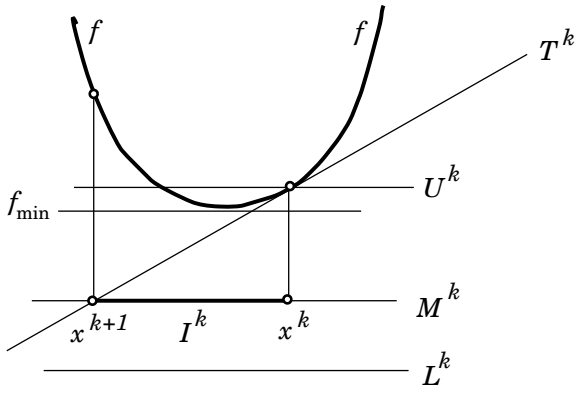


FIGURE 2. Illustration of case 2

happens, however convergence is slow.

• Algorithm 2 can still be used for line searches, i.e. minimizing f along a fixed direction $\mathbf{d}$, see [2]. The only change is in Line 3, where the directional Newton iteration (12) with $\mathbf{d}^k = \mathbf{d}$ should be used to solve (4).

• Denote the *level sets* of f by $\mathrm{lev}(f|\,\beta) := \{\mathbf{x} : f(\mathbf{x}) \le \beta\}\ ,\ \beta \in \mathbb{R}$. The monotonicity property (11) here becomes

$$\mathbf{x}^k \in \mathrm{lev}(f|\,f(\mathbf{x}^j))\ ,\ \text{for all } k > j = 0, 1, \cdots \tag{14}$$

i.e. iterates stay in previous level sets.

3. Convex functions of one variable

Theorem 1. If $f : \mathbb{R} \to \mathbb{R}$ is a convex function with attained infimum, then Algorithm 1 is valid for its minimization.

Proof. Only the updates in Case 2 need proof. Let T^k denote the line through the points $(x^k, f(x^k))$ and (x^{k+1}, M^k), and let I^k denote the interval between x^k and x^{k+1}, see Figure 2. By definition, T^k is tangent to the graph of f at $(x^k, f(x^k))$. Since f is convex, its graph lies above T^k. Also, in I^k, the horizontal line $y = M^k$ lies below T^k (the two lines cross at (x^{k+1}, M^k)), and therefore $M^k \le f(x)$ for all $x \in I^k$. The inequality $f(x^{k+1}) \ge f(x^k)$ implies that the minimum is in I^k, and therefore

$$f_{\min} \ge M^k\ . \tag{15}$$

The update $L^{k+1} := M^k$ is therefore correct. □

The following example shows that it is possible, for f with unattained infimum, to have no iterations of Type 2.

Example 1. $f(x) = e^{-x}$ has the unattained infimum $\lim_{x\to\infty} f(x) = 0$. For any x^0 and lower bound $L^0 \le 0$, all iterations are of Type 1. Since the lower bound is changed only in iterations of Type 2, it is never changed

here. For L^0 negative, all brackets $[L^k, U^k]$ are therefore of size $\geq |L^0|$ and the algorithm never stops if $\epsilon < |L^0|$. $\diamondsuit$

The following example shows that for quadratic f, Type 2 iterations must occur near a minimum, and so must iterations of Type 1.

Example 2. Let $f(x) = x^2$, with $f_{\min} = 0$. We assume that the current point x is positive, and a negative lower bound L. The upper bound U is the current value of $f(x)$, $U := x^2$. The value M is given by (3) as

$$M := \alpha U + (1-\alpha) L = \alpha x^2 + (1-\alpha) L .$$

A Newton iteration gives the next point

$$\begin{aligned} x_+ &:= x - \frac{x^2 - M}{2x} = x - \frac{x^2 - \alpha x^2 - (1-\alpha) L}{2x} \\ &= \frac{(1+\alpha)}{2} x + \frac{(1-\alpha)}{2} \frac{L}{x} \end{aligned} \tag{16}$$

Case 2 occurs when $x_+ \leq -x$. Using (16) we get an inequality for α,

$$\alpha \leq \frac{|L| - 3x^2}{|L| + x^2} \tag{17}$$

If RHS(17) is negative, i.e. if

$$\sqrt{\frac{|L|}{3}} < x$$

then Case 2 is impossible since α is, by definition, nonnegative. If RHS(17) is positive then Case 2 occurs for α and x satisfying

$$0 \leq \alpha \leq \frac{|L| - 3x^2}{|L| + x^2} .$$

If α is fixed throughout, then case 2 occurs for x and L satisfying

$$x^2 \leq \frac{1-\alpha}{3+\alpha} |L| . \tag{18}$$

Since in Case 2 the lower bound is raised ($|L|$ is decreased) but x is unchanged, inequality (18) becomes less likely if Case 2 is repeated. $\diamondsuit$

4. Numerical results

Assume that M^k is selected by (3). In Case 2 we have a guaranteed reduction of $1 - \alpha$. Clearly, Case 2 is more likely for lower values of M^k, i.e. for smaller α. This is illustrated in Example 2. However, the reduction ratio $1 - \alpha$ is bigger (i.e. worse) for smaller α. This tradeoff suggests that there is an optimal α. Unlike the situation in unimodal search, [19], there is no reason why the optimal α should correspond to a Fibonnaci search, or asymptotically to the golden section search.

To illustrate how the performance of Algorithm 1 depends on α, the algorithm was tested on 200 random quadratic convex scalar functions $f(x) := x^2 + 2bx + c$, with random b, c from the uniform distribution on $[-2, 2]$. The

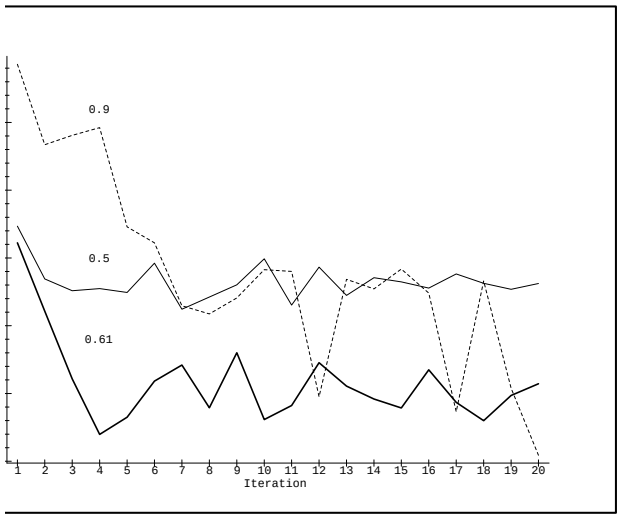


FIGURE 3. Average reduction ratio for each iteration for $\alpha = 0.5$, $\frac{\sqrt{5}-1}{2}$ (thick line) and 0.9 (dashed line).

initial solution x_0 was chosen randomly in $[-2, 2]$. Each problem was solved by Algorithm 1 with three typical values of the parameter α, $\alpha = 0.5$, $\frac{\sqrt{5}-1}{2}$ and 0.9. The stopping rule in all problems was identical, using (2) with a tolerance of $\epsilon = 10^{-12}$ and at most 20 iterations. For each iteration $k = 1, ..., 20$ we computed a reduction ratio (6), then the average reduction for all problems. Figure 3 shows the average reduction ratio for each iteration for the three values of α. The average reduction for $\alpha = \frac{\sqrt{5}-1}{2}$ is about 0.5 for all iterations (the thick line in Figure 3). The choices $\alpha = 0.5$ and $\alpha = 0.9$ resulted in bigger ratios, i.e. these choices are worse.

5. Convex functions of n variables

Sufficient conditions for the validity of Algorithm 2 are given in Theorems 2–5. First we introduce.

Definition 1. Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, $\mathbf{x}$ a point where f is differentiable, and $\nabla f(\mathbf{x}) \neq \mathbf{0}$. The *maximal nonincrease gradient step at* $\mathbf{x}$ is

$$\theta(\mathbf{x}) := \max \left\{ t : f\left(\mathbf{x} - t\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) \leq f(\mathbf{x}) \right\} . \tag{19}$$

Geometrically, $\theta(\mathbf{x})$ is the maximal step in the direction $-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$ that stays in $\mathrm{lev}(f|\, f(\mathbf{x}))$.

Example 3. The following results are easily computed.
(a) $f(x) = \|\mathbf{x}\|$, $\theta(\mathbf{x}) = 2\,\|\mathbf{x}\|$ for $\mathbf{x} \neq \mathbf{0}$.
(b) More generally, if $f(\mathbf{x})$ is a convex function with concentric circular level sets, centered at $\mathbf{x} = 0$, then $\theta(\mathbf{x}) = 2\,\|\mathbf{x}\|$ for $\mathbf{x} \neq \mathbf{0}$.
(c) $f(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x}$, Q positive definite, $\theta(\mathbf{x}) = 2\,\dfrac{\mathbf{x}^T Q^2 \mathbf{x}}{\mathbf{x}^T Q^3 \mathbf{x}}\,\|Q\mathbf{x}\|$ for $\mathbf{x} \neq \mathbf{0}$.
In all parts, $\theta(\mathbf{0})$ can be defined as $\mathbf{0}$. ♢

Theorem 2. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function, with attained infimum $f_{\min}$. If for all $\mathbf{x}$

$$\theta(\mathbf{x})\,\|\,\nabla f(\mathbf{x})\| \;\geq\; f(\mathbf{x}) - f_{\min}\,, \tag{20}$$

then Algorithm 2 is valid for minimizing f.

Proof. At a point $\mathbf{x}^k$, Case 2 occurs if the Newton step is greater than $\theta(\mathbf{x}^k)$, by definition of θ. In such a case the decrease along the Newton step ($(U^k - M^k)$ in Figure 2) is greater than $\theta(\mathbf{x}^k)\,\|\nabla f(\mathbf{x}^k)\|$, the LHS of (20) for $\mathbf{x} = \mathbf{x}^k$, showing that M^k is a correct lower bound for f. Therefore, if (20) holds for all $\mathbf{x}$, Algorithm 2 is valid. □

Example 4. For $f(\mathbf{x}) = \|\mathbf{x}\|$, $\mathbf{x} \neq \mathbf{0}$,

$$\theta(\mathbf{x})\,\|\nabla f(\mathbf{x})\| = 2\,\|\mathbf{x}\|$$

and condition (20) holds at all $\mathbf{x} \neq \mathbf{0}$. ♢

Theorem 3. For any differentiable convex function with concentric circular level sets, Condition (20) holds at all non–optimal $\mathbf{x}$.

Proof. Without loss of generality assume that the infimum is attained at $\mathbf{x} = \mathbf{0}$. By the Mean Value Theorem,

$$f(\mathbf{x}) - f(\mathbf{0}) = \|\mathbf{x}\|\,\|\,\nabla f(\boldsymbol{\xi})\|\,,\ \text{for some } \boldsymbol{\xi} \text{ between } \mathbf{0} \text{ and } \mathbf{x}\,.$$

The convexity of f implies that $\|\nabla f(\boldsymbol{\xi})\| \leq \|\nabla f(\mathbf{x})\|$. Therefore:

$$f(\mathbf{x}) - f(\mathbf{0}) \;\leq\; \|\mathbf{x}\|\,\|\nabla f(\mathbf{x})\| \;<2\|\mathbf{x}\|\,\|\nabla f(\mathbf{x})\| = \theta(\mathbf{x})\,\|\nabla f(\mathbf{x})\|\,,$$

by Example 3(b), showing that Condition (20) holds for all $\mathbf{x} \neq \mathbf{0}$. □

Theorem 4. Let

$$f(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} - c^T\mathbf{x} + \gamma \tag{21}$$

where the matrix Q is positive definite with eigenvalues

$$0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n\,.$$

Then Algorithm 2 is valid for minimizing f if

$$\frac{\lambda_1\lambda_n}{(\lambda_1 + \lambda_n)^2} \;\geq\; \frac{1}{16}\,. \tag{22}$$

Proof. The quadratic function (21) can be written in the form

$$f(\mathbf{x}) = \frac{1}{2}\,(\mathbf{x} - Q^{-1}c)^T Q(\mathbf{x} - Q^{-1}c) + \bar{\gamma}\ ,$$

where $\bar{\gamma}$ is a constant. Therefore, without loss of generality we can prove the theorem for

$$f(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x}\ .$$

Using Example 3(c), Condition (20) becomes

$$2\,\frac{\mathbf{x}^T Q^2\mathbf{x}}{\mathbf{x}^T Q^3\mathbf{x}}\,\|Q\mathbf{x}\|^2 \;\geq\; \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} \tag{23}$$

Substituting $\mathbf{x} = Q^{-1}\mathbf{y}$ we write (23) as

$$\|\mathbf{y}\|^4 \geq \frac{1}{4}\,(\mathbf{y}^T Q\mathbf{y})\,(\mathbf{y}^T Q^{-1}\mathbf{y})\ . \tag{24}$$

Recall the *Kantorovich Inequality*

$$\|\mathbf{y}\|^4 \geq \frac{4\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2}\,(\mathbf{y}^T Q\mathbf{y})\,(\mathbf{y}^T Q^{-1}\mathbf{y})\ . \tag{25}$$

Comparing (24) and (25) we conclude that (23) is guaranteed by (22). □

Remark 1. Theorem 4 says that the ratio λ_1/λ_n cannot be too small. From (22) we calculate

$$\frac{\lambda_1}{\lambda_n} \;\geq\; 7 - \sqrt{48} \approx .071796768 \tag{26}$$

Theorem 5. Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable convex function, $\mathbf{x}^*$ a minimizer of f. If there is a neighborhood N of $\mathbf{x}^*$ where for all $\mathbf{x} \neq \mathbf{x}^* \in N$ the Hessian matrix $H(\mathbf{x})$ is positive definite with eigenvalues

$$0 < \lambda_1(\mathbf{x}) \leq \lambda_2(\mathbf{x}) \leq \cdots \leq \lambda_n(\mathbf{x})\ ,$$

satisfying

$$\frac{\lambda_1(\mathbf{x})\lambda_n(\mathbf{x})}{(\lambda_1(\mathbf{x})+\lambda_n(\mathbf{x}))^2} \geq \frac{1}{16}\ , \tag{27}$$

then Algorithm 2, with initial point $\mathbf{x}^0$, is valid if the level set $\mathrm{lev}(f|\,f(\mathbf{x}^0))$ is contained in N.

Proof. Follows from Theorem 4 and the Taylor expansion of f. □

6. The Fermat–Weber location problem

The sufficient conditions (20), (22) or (27) are not meant to be checked before using Algorithm 2, any more than say the convergence conditions of Newton's method are checked before using it. These sufficient conditions merely indicate the types and characteristics of problems where the NB method is a reasonable choice. One such problem is discussed below.

Given a set of points $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\} \subset \mathbb{R}^d$ and positive weights $\{w_1, w_2, \ldots, w_N\}$, the *Fermat–Weber location problem* is to find a point $\mathbf{x}$ in $\mathbb{R}^d$ that minimizes

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \, \|\mathbf{x} - \mathbf{a}_i\| \; , \tag{28}$$

the sum of the weighted Euclidean distances. The gradient of f

$$\nabla f(\mathbf{x}) = \sum_{i=1}^{N} w_i \, \frac{\mathbf{x} - \mathbf{a}_i}{\|\mathbf{x} - \mathbf{a}_i\|} \tag{29}$$

exists for all $\mathbf{x} \notin \mathcal{A}$. A point $\mathbf{x}^*$ is optimal iff $\mathbf{0} \in \partial f(\mathbf{x}^*)$, which reduces to $\nabla f(\mathbf{x}^*) = \mathbf{0}$ if f is differentiable at $\mathbf{x}^*$. It follows then from (29) that $\mathbf{x}^*$ is a convex combination of the points of $\mathcal{A}$,

$$\mathbf{x}^* = \sum_{i=1}^{N} \lambda_i(\mathbf{x}^*) \, \mathbf{a}_i \; , \tag{30}$$

with weights

$$\lambda_i(\mathbf{x}) = \frac{w_i \, \|\mathbf{x} - \mathbf{a}_i\|^{-1}}{\sum\limits_{j=1}^{N} w_j \, \|\mathbf{x} - \mathbf{a}_j\|^{-1}} \; . \tag{31}$$

The *Weiszfeld Method* [33] for solving this problem is an iterative method with updates

$$\mathbf{x}_+ := \sum_{i=1}^{N} \lambda_i(\mathbf{x}) \, \mathbf{a}_i \; , \tag{32}$$

giving the *next iterate* $\mathbf{x}_+$ as a convex combination, with weights $\lambda_i(\mathbf{x})$ computed by (31) for the *current iterate* $\mathbf{x}$. Note that $\lambda_i(\mathbf{x})$ is undefined if $\mathbf{x} = \mathbf{a}_i$.

If the Weiszfeld iterates converge to a point $\mathbf{x}^*$, then $\mathbf{x}^*$ is optimal by (30).

The Weiszfeld method is the best–known method for solving the Fermat–Weber location problem, see the history in [24, § 1.3]. Below are selected references to the vast literature on this subject:

- The Fermat–Weber problem: [1], [3], [5], [6], [7], [8], [9], [10], [11], [12], [13], [18], [24].
- The Weiszfeld method and generalizations: [4], [6], [14], [17], [20], [22], [24], [23], [27], [28], [29], [30], [32].
- Other methods: [26], [31], [34].

We propose here an alternative method for the approximate solution of the Fermat–Weber problem (28): Algorithm 2 with f and ∇f given by (28) and (29) respectively. An initial bracket $[L^0, U^0]$ is given by:

$$\begin{aligned} U^0 &= f(\mathbf{x}^0) \text{ where } \mathbf{x}^0 \text{ is the initial iterate, and} \\ L^0 &= \|\mathbf{a}_i - \mathbf{a}_j\| \, \min\{w_i, w_j\} \text{ for any two points in } \mathcal{A} \, . \end{aligned}$$

Notes.
• The gradient (29) exists at all points $\mathbf{x} \notin \mathcal{A}$. If the point $\mathbf{x}$ coincides[1] with one of the N points $\mathbf{a}^i \in \mathcal{A}$, then the index i can be dropped from the summation in RHS(29), which means replacing the non-existing gradient $\nabla \left(\| \mathbf{x} - \mathbf{a}^i \| \right)$ by the vector $\mathbf{0}$ in the subgradient $\partial \ \| \mathbf{x} - \mathbf{a}^i \|$.
• There is no need to recompute $f(\mathbf{x})$ and $\nabla f(\mathbf{x})$ in iterations of Type 2, i.e. where only the lower bound L^k is updated, see Line 5 in Algorithm 2. This presents considerable economy since, in our experience, iterations of Type 2 occur about 75% of the time.
• The brackets $[L^k, U^k]$ are on the average halved at each iteration, so that after 10 iterations the bracket has been reduced to about 0.001 of its initial size.
• For large N and random data, the level sets of (28) are approximately concentric and circular, see e.g. Figure 4(b). The validity of Algorithm 2 is then guaranteed by Theorem 3.
• Algorithm 2 is easily applied to non–Euclidean norms. In particular, the Fermat–Weber problem with an ℓ_1–norm in $\mathbb{R}^n$ separates to n one–dimensional problems (see, e.g. [24, § 2.2]), each solvable by Algorithm 1 which is always valid by Theorem 1.

Example 5. The worst case for the Algorithm 2 is $N = 2$, i.e. two points with equal weights, in which case any point on the segment joining the points is optimal.

Let the point set $\mathcal{A}$ consist of two points, $(-c, 0)$ and $(c, 0)$, where c is positive, and let $w_1 = w_2 = 1$. The level sets of the function (28) are then a family of ellipses

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \ ,$$

parametrized by $\kappa \in [1, \infty)$,

$$a = \kappa \, c \ , \ b = \sqrt{a^2 - c^2} = \sqrt{\kappa^2 - 1} \ c \ .$$

The value $\kappa = 1$ gives a degenerate ellipse, the segment $[-c, c]$ on the x–axis, that is the optimal set. Condition (20) holds if

$$\begin{aligned} \frac{b^2}{a^2} \ = \ \frac{\kappa^2 - 1}{\kappa^2} \ &\geq \ 7 - \sqrt{48} \ . \\ \therefore \ \kappa \ &\geq \ \frac{1}{\sqrt{\sqrt{48} - 6}} \approx 1.037954849 \ . \end{aligned}$$

The critical value $\kappa = 1.037954849$ gives a narrow ellipse, containing and closely approximating the optimal set. ♢

[1]An event of zero probability if the algorithm is used with random data $\mathcal{A}$ and random initial iterate $\mathbf{x}^0$.

7. Numerical experiments

In the numerical experiments reported here the weights w_i in (28) were all taken equal to 1.

Algorithm 2 was tested on random problems of size $N \times n$, where N is the number of points, $N = 1000, 2000, \ldots, 10000$, and n is the dimension of the space, $n = 2, 3, \ldots, 10$.

For each of the 90 combinations of N and n, we solved 20 problems, a total of 1800 problems. Each problem (of size $N \times n$) used a random point set $\mathcal{A}$ of N points from the uniform distribution in the box $[-10, 10]^n$. The mass center $\boldsymbol{\mu}$ and the standard deviation σ of the set $\mathcal{A}$ were computed, and the initial point $\mathbf{x}^0$ was taken as $\mathbf{x}^0 := \boldsymbol{\mu} - \sigma\mathbf{e}$ where $\mathbf{e}$ is the vector of ones. The stopping rule in all problems was identical, using tolerance $\epsilon = 10^{-6}$ on the ratio $(U^k - L^k)/(U^0 - L^0)$. The average numbers of iterations, for 20 random problems of size $N \times n$ with $N = 1000, 2000, ..., 10000$ points of dimension $n = 2, 3, ..., 10$ are tabulated in Table 1. In these problems, the average percentage of iterations of Type 2 was 75%.

Table 1 shows that the average performance of the algorithms (average number of iterations needed for a bracket reduction of 10^{-6}) does not depend on the problem size $N \times n$, indeed the average number of iterations is about 20 for all 90 combinations of $N = 1000, 2000, \ldots, 10,000$ and $n = 2, 3, \ldots, 10$. The fact that about 20 iterations are needed to reduce the initial bracket $[L^0, U^0]$ to a millionth of its size shows that the average reduction ratio per iteration is ≈ 0.5 .

Figures 4 illustrate one random problem of size $N \times n$ with $N = 1000$, $n = 10$ from Table 1. Because the dimension is 10, we show the first two coordinates (x_1, x_2). The point set $\{\mathbf{a}_i : i = 1, \ldots, 1000\}$ is shown in Figure 4(a). Figure 4(b) shows the level set of the distance function $\sum_{i=1}^{N} \| \mathbf{x} - \mathbf{a}^i \|$ and the trajectory of the iterates. Recall that the iterate $\mathbf{x}^k$ does not change in an iteration of Type 2: this explains why the trajectory in Figure 4(b) shows only about 5 iterates (the last ones are too close to separate).

In another test, hundreds of problems with randomly generated N–point sets $(10 \le N \le 100)$ in $\mathbb{R}^n$ $(2 \le n \le 10)$ were solved by Algorithm 2 and by the Weiszfeld Method (32), using a cold start (i.e. a random initial point $\mathbf{x}^0$) or a warm start (e.g. $\mathbf{x}^0$ the mean of the points in $\mathcal{A}$).

The results from 1800 randomly generated location problems with N points in $\mathbb{R}^n$, $N = 10, 20, ..., 100$ and $n = 2, 3, ..., 10$, with cold starts, are shown in Table 2. For each such N and n, 20 random problems were generated and solved by Algorithm 2 and by the Weiszfeld Method (32). Table 2 shows the percentage of problems for which Algorithm 2 gives a lower value than the Weiszfeld Method after 10 iterations. The advantage of using Algorithm 2 increases with the number of iterations: for 20 iterations the corresponding table has 100% for all N and n, i.e. after 20 iterations Algorithm 2 is always better than the Weiszfeld Method. On the other hand,

$n =$ dimension	$N=$ number of points									
	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
2	19.6	20.1	20.4	20.6	20.1	20.1	20.5	20.2	20.5	20.4
3	20.2	19.8	19.5	19.9	19.8	20.0	19.9	19.9	20.2	19.6
4	19.6	19.4	19.6	19.8	19.6	19.6	19.7	19.7	19.3	19.5
5	19.3	19.3	19.5	19.4	19.7	19.2	19.7	19.3	19.2	19.7
6	19.3	19.4	19.4	19.4	19.7	19.3	19.4	19.4	19.6	19.6
7	19.7	19.3	19.8	19.3	19.5	19.7	19.7	19.6	19.7	19.6
8	19.4	19.0	19.5	19.9	19.8	19.7	20.0	19.9	19.9	19.9
9	19.3	19.2	20.2	19.4	19.6	19.9	19.8	20.5	19.9	19.9
10	19.7	19.5	19.6	19.8	19.9	19.8	19.9	19.6	19.7	20.1

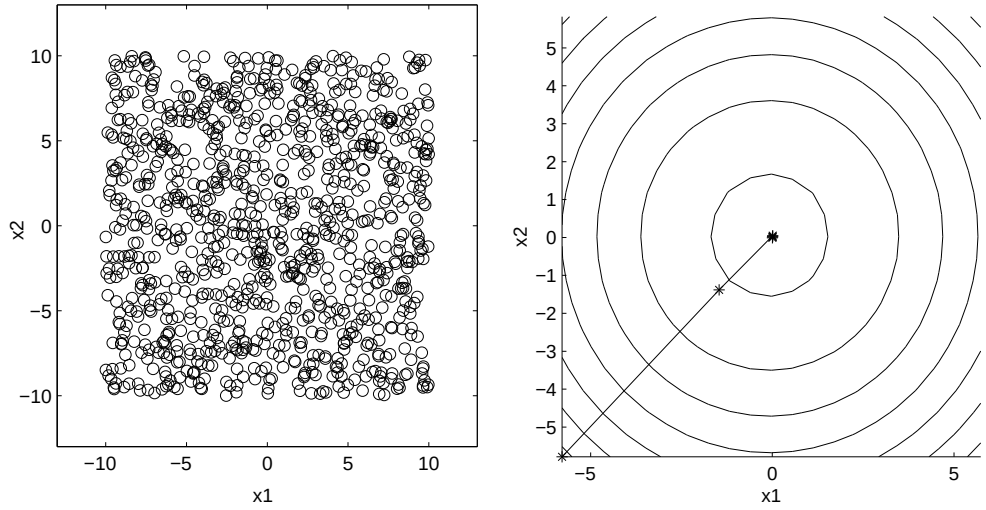
TABLE 1. The average numbers of iterations, for 20 location problems with N points in $\mathbb{R}^n$, $N = 1000, 2000, ..., 10000$ and $n = 2, 3, ..., 10$

$n =$ dimension	$N=$ number of points									
	10	20	30	40	50	60	70	80	90	100
2	95	100	100	100	100	100	100	100	100	100
3	90	100	95	100	95	95	100	95	100	100
4	90	80	95	80	85	95	100	80	95	100
5	85	85	75	80	70	90	85	90	85	85
6	80	50	75	75	60	80	60	75	60	70
7	55	40	70	60	60	50	45	35	70	50
8	75	45	45	55	40	40	55	50	50	40
9	50	30	30	35	55	25	35	35	45	35
10	50	45	30	40	25	30	40	25	40	55

TABLE 2. The percentages of problems for which Algorithm 2 gives better results than Weiszfeld Method after 10 iterations, for 20 location problems with N points in $\mathbb{R}^n$, $N = 10, 20, ..., 100$ and $n = 2, 3, ..., 10$

the Weiszfeld Method performed better when the experiment was repeated with an upper bound of 5 iterations.

The comparison between Algorithm 2 and Weiszfeld Method was essentially the same when warm starts were used.



(a) Point set (b) Level sets and trajectory of the solution

FIGURE 4. A problem with 1000 random points in $[-10, 10]^{10} \in \mathbb{R}^{10}$. Pictures show first two coordinates (x_1, x_2).

Example 6. Consider the Fermat–Weber problem with five points $\mathbf{a}_i$ and corresponding weights w_i as follows:

$$\begin{aligned}
\mathbf{a}_1 &= (-1, -1) \ , w_1 = 1 \\
\mathbf{a}_2 &= (-1, 1) \quad , w_2 = 1 \\
\mathbf{a}_3 &= (1, -1) \quad , w_3 = 1 \\
\mathbf{a}_4 &= (1, 1) \qquad , w_4 = 1 \\
\mathbf{a}_5 &= (100, 0) \quad , w_5 = 4
\end{aligned}$$

This simple example, based on [11, p. 277], is difficult for the Weiszfeld method. The first twenty iterations given in Table 3, illusstrate very slow convergence of Weiszfeld's method, which requires millions of iterations for the initial solution $x_0 = (80, 0)$ (and for other x_0 on the x–axis). In contrast, the NB algorithm converges reasonably fast, as does Drezner's accelerated Weiszfeld method, [11]. ♢

8. Conclusions and suggestions for further research

(a) A convex minimization problem with constraints can be formulated as a sequence of unconstrained problems, using suitable penalty functions or Lagrange multipliers. Therefore, the NB method can be used for solving constrained convex minimization problems. Further research is needed to determine if the NB method is practical for such problems.
(b) The average performance of Algorithms 1–2 depends on the choice of the parameter α in (3). Determining the optimal choice of α in Algorithms 1–2

	Weiszfeld Method	The NB Method
0	[80.0, 0]	[80.0, 0]
...	...	...
11	[80.0037, 0]	[80.0, 0]
12	[80.00387991, 0]	[95.84, 0]
13	[80.00397613, 0]	[95.84, 0]
14	[80.00406547, 0]	[95.84, 0]
15	[80.00414883, 0]	[98.17, 0]
16	[80.00422694, 0]	[99.69, 0]
17	[80.00430047, 0]	[99.69, 0]
18	[80.00436990, 0]	[99.69, 0]
19	[80.00443568, 0]	[99.84, 0]
20	[80.00449821, 0]	[99.93, 0]

TABLE 3. 20 iterations for Example 6

requires further analysis and/or experiments.
(c) Our numerical experiments suggest the following comparison between Algorithm 2 and the Weiszfeld Method:

- Both methods require, on the average, about the same work per iteration (recall that in Algorithm 2, $f(\mathbf{x})$ and $\nabla f(\mathbf{x})$ need to be calculated only in iterations of Type 1).
- Algorithm 2 works better at the final iterations, i.e. near the optimal solution.
- The Weiszfeld Method works better at the starting iterations.
- Algorithm 2 has a natural stopping rule (2). In contrast, a stopping rule for the Weiszfeld method requires computing a lower bound, see [9], [15], [16], [23], [25] and [24, pp. 16–17, 88–99].

These observations suggest that a hybrid method, starting with Weiszfeld and ending with Algorithm 2 iterations, may perform better than either method.

Finally, accelerated versions of Weiszfeld's method were developed by several authors, see, e.g., [4], [11], and [22]. Algorithm 2 can be accelerated in an analogous manner, a topic for future research.

ACKNOWLEDGMENT

We thank Gabriela Alexe, Mikhail Nediak, Paulo Silva, Yehuda Vardi, Cun-Hui Zhang and the referees for constructive suggestions.

REFERENCES

[1] C. Antoni and A. Dalena, The Fermat–Weber problem and the Lagrangian duality theory, pp. 5–12 in *New Trends in Mathematical Programming*, Kluwer Acad. Publ., Boston, 1998
[2] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995

[3] J. Brimberg, The Fermat–Weber location problem revisited, *Math. Prog.* **71** (1995), no. 1 (Ser. A), 71–76
[4] J. Brimberg, R. Chen and D. Chen, Accelerating convergence in the Fermat–Weber location problem, *Oper. Res. Lett.* **22**(1998), no. 4-5, 151–157
[5] J. Brimberg and R. F. Love, Local convexity results in a generalized Fermat–Weber problem, *Comput. Math. Appl.* **37** (1999), 87–97
[6] R. Chandrasekaran and A. Tamir, Open questions concerning Weiszfeld's algorithm for the Fermat–Weber location problem, *Math. Prog.* **44**(1989), no. 3 (ser. A), 293–295
[7] R. Chandrasekaran and A. Tamir, Algebraic Optimization: the Fermat-Weber location problem, *Math. Prog.* **46**(1990), no. 2 (ser. A), 219–224
[8] F. Cordellier and J. Ch. Fiorot, On the Fermat–Weber problem with convex cost functions, *Math. Prog.* **14** (1978), 295–311
[9] Z. Drezner, The planar two–center and two–median problems, *Transportation Sci.* **18**(1984), 351–361
[10] Z. Drezner, A note on the Weber location problem, *Ann. Oper. Res.* **40** (1992), 153–161
[11] Z. Drezner, A note on accelerating the Weiszfeld procedure, *Location Science* **3** (1995), no. 4, 275–279
[12] Z. Drezner and A. J. Goldman, On the set of optimal points to the Weber problem, *Transportation Sci.* **25** (1991), 3–8
[13] Z. Drezner and D. Simchi-Levi, Asymptotic behavior of the Weber location problem on the plane, *Ann. Oper. Res.* **40** (1992), 163–172
[14] U. Eckhardt, Weber's problem and Weiszfeld's algorithm in general spaces, *Math. Prog.* **18**(1980), no. 2, 186–196
[15] J. Elzinga and D. W. Hearn, On stopping rules for facilities location algorithms, *IIE Transactions* **15**(1983), 81–83
[16] H. Juel, On a rational stopping rule for facilities location algorithms, *Naval Res. Logist. Quart.* **31**(1984), 9–11
[17] I. N. Katz, Local convergence in Fermat's problem, *Math. Prog.* **6**(1974), 89–104
[18] W. Kaplan and Wei H. Yang, Duality theorem for a generalized Fermat-Weber problem, *Math. Prog.* **76** (1997), no. 2 (Ser A) 285–297
[19] J. Kiefer, Sequential minimax search for a maximum, *Proc. Amer. Math. Soc.* **4**(1953), 502–506
[20] H. Kuhn, A note on fermat's problem, *Math. Prog.* **4**(1973), 98–107
[21] Y. Levin and A. Ben-Israel, Directional Newton methods in n variables, *Math. of Computation* (to appear)
[22] Y. Li, A Newton acceleration of the Weiszfeld algorithm for minimizing the sum of Euclidean distances, *Comput. Optim. Appl.* **10**(1998), no. 3, 219–242
[23] R. Love and P. D. Dowling, A generalized bounding method for multifacility location models, *Oper. Res.* **37**(1989), 89–104
[24] R. Love, J. Morris and G. Wesolowsky, Facilities Location: Models and Methods, *North-Holland*, 1988
[25] R. Love and W. Yeong, A stopping rule for facilities location algorithms, *AIIE Transactions* **13**(1981), 357–362
[26] C. Michelot and O. Lefebvre, A primal-dual algorithm for the Fermat–Weber problem involving mixed gauges, *Math. Prog.* **39** (1987), 319–335
[27] J. G. Morris, Convergence of the Weiszfeld algorithm for Weber problems using a generalized "distance" function, *Oper. Res.* **29** (1981), 37–48
[28] J. G. Morris and W. A. Verdini, Minisum l_p distance location problems solved via a perturbed problem and Weiszfeld's algorithm, *Oper. Res.* **27** (1979), 1180–1188
[29] L. M. Ostresh, Jr., Convergence and descent in the Fermat location problem, *Transportation Sci.* **12** (1978), 153–164

[30] L. M. Ostresh, Jr., On the convergence of a class of iterative methods for solving the Weber location problem, *Oper. Res.* **26** (1978), 597–609
[31] M. L. Overton, A quadratically convergent method of minimizing a sum of Euclidean norms, *Math. Prog.* **27** (1983), 34–63
[32] H. Voß and U. Eckhardt, Linear convergence of generalized Weiszfeld's method, *Computing* **25** (1980), 243–251
[33] E. Weiszfeld, Sur le point par lequel la somme des distances de n points donnés est minimum, *Tohoku Math. J.* **43** (1937), 355–386
[34] S. Wu, A polynomial time algorithm for solving the Fermat-Weber location problem with mixed norms, *Optimization* **30** (1994), no. 3, 227–234

Yuri Levin, RUTCOR–Rutgers Center for Operations Research, Rutgers University, 640 Bartholomew Rd, Piscataway, NJ 08854-8003, USA
E-mail address: ylevin@rutcor.rutgers.edu

Adi Ben-Israel, RUTCOR–Rutgers Center for Operations Research, Rutgers University, 640 Bartholomew Rd, Piscataway, NJ 08854-8003, USA
E-mail address: bisrael@rutcor.rutgers.edu