ON NEWTON'S METHOD FOR THE FERMAT-WEBER LOCATION PROBLEM

Simone Görner and Christian Kanzow

University of Würzburg
Institute of Mathematics
Emil-Fischer-Str. 30
97074 Würzburg
Germany
SimoneGoerner@gmx.de
kanzow@mathematik.uni-wuerzburg.de

December 8, 2015

Abstract

This paper considers the Fermat-Weber location problem. It is shown that, after a suitable initialization, the standard Newton method can be applied to the Fermat-Weber problem and is globally and locally quadratically convergent. A numerical comparison with the popular Weiszfeld algorithm shows that Newton's method is significantly more efficient than the Weiszfeld scheme.

Key Words: Fermat-Weber location problem, Weiszfeld method, Newton method, global convergence, local quadratic convergence

1 Introduction

Consider the nonsmooth optimization problem

$$\min_x \; f(x) := \sum_{i=1}^{m} \omega_i \|x - a_i\| \tag{1}$$

that we call the *Fermat-Weber problem*, though it comes with many different names in the literature, see the discussion in [5]. Historically, the Fermat-Weber problem goes back to the French mathematician Pierre de Fermat for the special case $n = 2, m = 3, \omega_i = 1$ $(i = 1, 2, 3)$, whereas the more general formulation (1) is named after the German economist Alfred Weber. The vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ denote pairwise disjoint points, sometimes called anchor points, the scalars $\omega_i > 0$ are some positive weights, and $\|\cdot\|$ stands for the Euclidean vector norm. Throughout this note, we assume without loss of generality that $m \geq 3$.

The Fermat-Weber problem is one of the most famous problems in location theory [4, 11, 14], and it has been the subject of intense research. Here we are interested in the numerical solution of this problem. A recent survey on some solution methods can be found in [2], and the interested reader is referred to the extensive list of references in that paper for more details.

The most prominent method for the solution of (1) is Weiszfeld's algorithm. This is a fixed-point iteration based on a suitable fixed-point reformulation of the optimality condition $\nabla f(x) = 0$. Formally, this gradient does not exist everywhere, and this is indeed the reason why Weiszfeld's algorithm may not converge. But these problems can be fixed in different ways, so one gets convergent variants of Weiszfeld's algorithm, see, again, the survey paper [2]. Nevertheless, the pure Weiszfeld method still seems to be one of the most popular methods for the solution of the Fermat-Weber problem.

A disadvantage of Weiszfeld's method and its convergent modifications is the poor rate of convergence. This slow convergence has to be expected, since this method may be interpreted as a gradient method, combined with a special choice of a stepsize [2]. Though f is nonsmooth, this observation allows to use ideas from accelerated gradient-type methods, originally proposed by Nesterov [13], in order to improve the speed of convergence. Nevertheless, the corresponding methods are still gradient-type methods with a (sub-) linear rate of convergence.

To overcome this deficiency, some authors suggest to incorporate Newton-type steps or to apply modified Newton iterations. In particular, Katz [9] investigates the local acceleration of Weiszfeld's method by Newton steps. Jiang et al. [8] switch between Weiszfeld and Newton iterations based on a certain criterion. Levin and Ben-Israel [10] use the Newton bracketing method for solving the Fermat-Weber problem which is based on computing suitable lower and upper bounds; the idea mainly works for one-dimensional problems, whereas its generalization for higher dimensions is essentially a gradient-type method. The two related papers [16] by Overton and [3] by Calamain and Conn present methods with a quadratic rate of convergence by using an active set strategy and corresponding projected Newton steps in order to circumvent problems arising from the nondifferentiability of the objective function.

All these methods can use an arbitrary starting point. Here we propose a simple method which first constructs a suitable starting point in an initialization phase and then applies the standard Newton method to the solution of (1). It turns out that the method converges globally and locally quadratically. Many of the corresponding ingredients that are used in our approach are known from the literature, but it seems that the overall method has not yet been applied to the Fermat-Weber problem.

In principle, Newton's method is more expensive than gradient-type methods like the Weiszfeld algorithm. On the other hand, typical applications of the Fermat-Weber problem use $n = 2$ or $n = 3$, hence the solutions of the Newton equations at each step of our Newton iteration are very cheap. Indeed, a numerical comparison of our method with the Weiszfeld algorithm shows that our method is also significantly faster in terms of CPU-times than Weiszfeld's algorithm.

The organization is as follows: We first recall some known facts in Section 2. The algorithm together with a global and local convergence theory are presented in Section 3. Some numerical results are given in Section 4. We conclude with some final remarks in Section 5.

Notation: The symbol $\|x\|$ always denotes the Euclidean norm of a vector x. Since the mapping f from (1) is convex, it is directionally differentiable everywhere, and we denote its (one-sided) directional derivative at a point $x \in \mathbb{R}^n$ in the direction $d \in \mathbb{R}^n$ by $f'(x; d)$, i.e., $f'(x; d) := \lim_{t \to 0^+} \frac{f(x+td)-f(x)}{t}$.

2 Preliminaries

We first recall some standard properties of the Fermat-Weber problem.

Proposition 2.1. *The following statements hold:*

(a) The function f from (1) *is convex (though not differentiable everywhere).*

(b) The Fermat-Weber problem (1) *always has a solution.*

(c) The solution of (1) *is unique if the points $a_1, \ldots, a_m$ are not collinear (i.e., if not all of them are on the same line).*

Statement (a) is clear, statement (b) uses the continuity of f together with the obvious fact that the level sets

$$\mathcal{L}(x^0) := \{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\}$$

are nonempty and compact for any given $x^0 \in \mathbb{R}^n$. The last statement (c) holds since one can verify that f is strictly convex under the given assumption.

There is also a simple and constructive criterion for one of the anchor points a_i to be a minimum. The corresponding result is summarized in the following lemma which will play an important role also for the initialization phase of our Newton method, see, e.g., [2].

Proposition 2.2. *Consider the Fermat-Weber problem* (1) *and determine an index*

$$p \in \{1, \ldots, m\} \quad \textit{such that} \quad f(a_p) = \min_{i=1,\ldots,m} f(a_i).$$

Then a_p is a solution of (1) if and only if

$$\left\| \sum_{i=1, i\neq p}^{m} \omega_i \frac{a_p - a_i}{\|a_p - a_i\|} \right\| \leq \omega_p$$

holds.

The previous result follows from the fact that a solution of a (possibly nondifferentiable) convex function can be characterized by zero being an element of the corresponding subdifferential.

Suppose now that a_p is not a minimum of f. Then there exists a descent direction of f at a_p. One possible descent direction can be computed by solving the optimization problem

$$\min_d \; f'(a_p; d) \quad \text{s.t.} \quad \|d\| = 1. \tag{2}$$

Its solution is known analytically and given in the following result, cf. [2].

Proposition 2.3. *The vector*

$$d_p := -R_p/\|R_p\| \quad \textit{with} \quad R_p := \sum_{i=1, i\neq p}^{m} \omega_i \frac{a_p - a_i}{\|a_p - a_i\|}$$

is a solution of (2).

Since a_p is not a minimum of f, it holds that $f'(a_p; d_p) < 0$. Hence, from the definition of the (one-sided) directional derivative, it follows that $f(a_p + t d_p) < f(a_p)$ holds for all $t > 0$ sufficiently small. A suitable stepsize satisfying this condition can be computed by simple backtracking. On the other hand, there also exist explicit formulas for appropriate stepsizes. For example, we have $f(a_p + t_p d_p) < f(a_p)$ for the particular stepsize

$$t_p := \big(\|R_p\| - \omega_p\big)/L(a_p) \quad \text{with} \quad L(a_p) := \sum_{i=1, i\neq p}^{m} \frac{\omega_i}{\|a_p - a_i\|}, \tag{3}$$

cf. [2, 18] for more details.

3 Algorithm and Convergence

Based on the results recalled in the previous section, we follow a very simple strategy in order to solve the Fermat-Weber problem: We first determine an anchor point a_p where the minimum $\min\big\{f(a_1), \ldots, f(a_m)\big\}$ is attained. Note that the index p is not necessarily unique. We then check whether a_p is already a minimum of f by using the constructive criterion from Proposition 2.2. If this criterion is satisfied, we are done. Otherwise we compute a descent direction d_p based on Proposition 2.3 and a suitable step size $t_p > 0$ (like the one from (3)) such that $f(a_p + t_p d_p) < f(a_p)$ holds. Then we begin Newton's method using the starting point $x^0 := a_p + t_p d_p$.

To this end, note that f is twice continuously differentiable on an open set containing $\mathcal{L}(x^0)$. Hence, this allows us to apply Newton's method to the mapping f with the above starting point x^0. The method is globalized using an Armijo line search, in particular, we therefore have the descent property $f(x^{k+1}) \leq f(x^k)$ so that all iterates x^k belong to $\mathcal{L}(x^0)$.

Algorithm 3.1. *(Newton-type Method for Fermat-Weber Problem)*

(S.0) Determine $p \in \{1,\dots,m\}$ such that $f(a_p) = \min\{f(a_1),\dots,f(a_m)\}$. If a_p satisfies the condition from Proposition 2.2 for a minimum: STOP. Otherwise compute d_p and t_p from Proposition 2.3 and (3), respectively, set $x^0 := a_p + t_p d_p, k := 0$, and choose parameters $\varepsilon \geq 0, \rho \in (0,1), \sigma \in (0,1/2)$.

(S.1) If $\|\nabla f(x^k)\| \leq \varepsilon$: STOP.

(S.2) Compute the Newton direction d^k by solving $\nabla^2 f(x^k)d = -\nabla f(x^k)$.

(S.3) Compute a stepsize $t_k := \max\{\rho^\ell \mid \ell = 0,1,2,\dots\}$ such that

$$f(x^k + t_k d^k) \leq f(x^k) + \sigma t_k \nabla f(x^k)^T d^k.$$

(S.4) Set $x^{k+1} := x^k + t_k d^k, k \leftarrow k+1$, and go to (S.1).

Before we start our convergence analysis, let us recall that f is known to be strictly convex on the level set $\mathcal{L}(x^0)$, where x^0 denotes the starting point from Algorithm 3.1. But this alone does not guarantee global convergence of Newton's method, since a couterexample from [7] shows that Newton's method may not converge for strictly convex functions even in combination with the stronger Wolfe line search rule.

On the other hand, Newton's method combined with the Wolfe line search rule is known to be globally convergent if the search directions satisfy an angle condition, see, e.g. [15, Section 3.2] or [17, Theorem 3.2.5]. In fact, it is not the Wolfe line search rule that makes Newton's method globally convergent, but the fact that one uses an efficient line search in the sense defined in [19]. In general, the Armijo rule used in our Algorithm 3.1 is not efficient, but much simpler to implement than the Wolfe conditions.

Having these observations in mind, we start the convergence analysis by proving that the Hessians of f are positive definite on the level set $\mathcal{L}(x^0)$. Note that this is a stronger property than the strict convexity of f on this set. In particular, it guarantees that steps (S.2) and (S.3) in Algorithm 3.1 are well-defined.

Proposition 3.2. *For any $x \in \mathcal{L}(x^0)$, the Hessian $\nabla^2 f(x)$ is positive definite.*

Proof. First recall that f is twice continuously differentiable on an open set containing the level set $\mathcal{L}(x^0)$. Since f is convex, this implies that the Hessian matrices $\nabla^2 f(x)$ are positive semidefinite for all $x \in \mathcal{L}(x^0)$. In order to show positive definiteness, take an arbitrary $x \in \mathcal{L}(x^0)$ and note that the pure existence of this vector automatically implies that the points $a_1,\dots,a_m$ are not collinear. Furthermore, by an elementary calculation, we get the representation

$$\nabla^2 f(x) = \sum_{i=1}^{m} \frac{\omega_i}{\|x-a_i\|^3}\big(\|x-a_i\|^2 I - (x-a_i)(x-a_i)^T\big).$$

Using the Cauchy-Schwartz inequality, this implies

$$\begin{aligned} d^T\nabla^2 f(x)d &= \sum_{i=1}^m \frac{\omega_i}{\|x-a_i\|^3}\big(\|x-a_i\|^2\|d\|^2 - (d^T(x-a_i))^2\big) \\ &\geq \sum_{i=1}^m \frac{\omega_i}{\|x-a_i\|^3}\big(\|x-a_i\|^2\|d\|^2 - \|d\|^2\|x-a_i\|^2\big) \\ &= 0 \end{aligned}$$

for all $d \neq 0$, and equality can hold only if all vectors d and $x - a_i$ $(i = 1, \ldots, m)$ are linearly dependent. But this would imply that the vectors $a_1, \ldots, a_m$ are collinear, and this contradiction completes the proof. □

As noted in [12], the globalized Newton method based on the Wolfe conditions may still not converge to a stationary point even if the Hessian matrices $\nabla^2 f(x^k)$ are positive definite at all iterates x^k. In our setting, however, the situation is different: Since the level set $\mathcal{L}(x^0)$ is compact and the minimal and maximal eigenvalues of the matrices $\nabla^2 f(x)$ depend continuously on $x \in \mathcal{L}(x^0)$, we immediately obtain the following consequence of Proposition 3.2 which essentially states that f is a uniformly convex function on the compact level set $\mathcal{L}(x^0)$.

Corollary 3.3. *There exist constants $\beta \geq \alpha > 0$ such that*

$$\alpha\|d\|^2 \leq d^T\nabla^2 f(x^k)d \leq \beta\|d\|^2$$

for all $d \in \mathbb{R}^n$ and all $x \in \mathcal{L}(x^0)$.

Note that Algorithm 3.1 uses an Armijo-line search in order to globalize the local Newton method. In general, the Armijo-rule is not an efficient line search in the sense of [19]. This is in contrast to many other, but more complicated line search rules. Our next results states, however, that in our particular context, the Armijo rule is an efficient line search (without giving an explicit definition of an efficient line search here).

Lemma 3.4. *There exists a constant $\theta > 0$ such that*

$$f(x^k + t_k d^k) \leq f(x^k) - \theta\left(\frac{\nabla f(x^k)^T d^k}{\|d^k\|^2}\right)^2 \quad \forall k \in \mathbb{N}.$$

Proof. Recall that $d^k = -\nabla^2 f(x^k)^{-1}\nabla f(x^k)$, hence $\nabla^2 f(x^k)d^k = -\nabla f(x^k)$ holds for all $k \in \mathbb{N}$. Using Corollary 3.3, this implies

$$-\frac{\nabla f(x^k)^T d^k}{\|d^k\|^2} = \frac{(d^k)^T\nabla^2 f(x^k)d^k}{\|d^k\|^2} \leq \beta \quad \forall k \in \mathbb{N}. \tag{4}$$

Now, for the moment, consider a fixed iteration index $k \in \mathbb{N}$. Let $\varphi_k(t) := f(x^k + td^k)$. Since f is continuous and bounded from below, there exists a smallest $\hat{t}_k > 0$ such that

$$\varphi_k'(\hat{t}_k) = \sigma\varphi_k'(0).$$

Since ∇f is Lipschitz-continuous on $\mathcal{L}(x^0)$ and $x^k + td^k \in \mathcal{L}(x^0)$ for all $0 < t \leq \hat{t}_k$, there exists a constant $L > 0$ satisfying

$$\sigma\varphi_k'(0) = \varphi_k'(\hat{t}_k) = \varphi_k'(0) + \big(\varphi_k'(\hat{t}_k) - \varphi_k'(0)\big) \leq \varphi_k'(0) + \hat{t}_k L\|d^k\|^2.$$

This implies

$$\hat{t}_k \geq -\frac{(1-\sigma)\varphi_k'(0)}{L\|d^k\|^2} \tag{5}$$

(recall that $\varphi_k'(0) = \nabla f(x^k)^T d^k < 0$). We now distinguish two cases.

Case 1: $t_k = 1$ in the Armijo rule. Then (4) implies

$$t_k = 1 \geq -\frac{1}{\beta}\frac{\nabla f(x^k)^T d^k}{\|d^k\|^2}$$

and, therefore,

$$f(x^k + t_k d^k) = \varphi_k(t_k) \leq \varphi_k(0) + \sigma t_k \varphi_k'(0) \leq f(x^k) - \frac{\sigma}{\beta}\left(\frac{\nabla f(x^k)^T d^k}{\|d^k\|^2}\right)^2.$$

Note that the constant σ/β is independent of k.

Case 2: $t_k < 1$ in the Armijo rule. Then t_k/ρ violates the Armijo condition. On the other hand, $\hat{t}_k$ and all stepsizes $0 < t \leq \hat{t}_k$ satisfy the Armijo condition, hence it follows that

$$\frac{1}{\rho}t_k > \hat{t}_k.$$

Using (5), this implies

$$\begin{aligned} f(x^k + t_k d^k) &= \varphi_k(t_k) \\ &\leq \varphi_k(0) + \sigma t_k \varphi_k'(0) \\ &\leq \varphi_k(0) + \sigma\rho\hat{t}_k\varphi_k'(0) \\ &\leq f(x^k) - \frac{\sigma\rho(1-\sigma)}{L}\left(\frac{\nabla f(x^k)^T d^k}{\|d^k\|}\right)^2, \end{aligned}$$

where also the constant $\sigma\rho(1-\sigma)/L$ is independent of the fixed k.

Therefore, for all $k \in \mathbb{N}$, Cases 1 and 2 together imply that the statement holds with $\theta := \min\big\{\sigma/\beta, \sigma\rho(1-\sigma)/L\big\}$. □

Recall that, in contrast to the Wolfe conditions, Armijo's line search rule is usually not an efficient line search in the sense of [19], at least not under the assumptions that guarantee the efficiency of the Wolfe conditions. However, our analysis shows that the Armijo rule is efficient for strongly convex functions on bounded level sets.

Putting the previous results together, we finally obtain the following global and local convergence result where, implicitly, we assume that Algorithm 3.1 does not terminate after finitely many iterations in step (S.1).

Theorem 3.5. *If Algorithm 3.1 does not terminate in step (S.0), then the sequence* $\{x^k\}$ *generated by this method converges to the unique solution of the Fermat-Weber problem. Furthermore, the local rate of convergence is quadratic.*

Proof. First note that the Hessian $\nabla^2 f$ is locally Lipschitz on the level set $\mathcal{L}(x^0)$. Furthermore, Corollary 3.3 implies

$$-\frac{\nabla f(x^k)^T d^k}{\|\nabla f(x^k)\| \, \|d^k\|} \geq \frac{(d^k)^T \nabla^2 f(x^k) d^k}{\|d^k\|^2 \|\nabla^2 f(x^k)\|} \geq \frac{\alpha}{\beta} > 0$$

for all $k \in \mathbb{N}$, where we assumed that $\|\nabla^2 f(x^k)\| \leq \beta$ holds. Hence the search directions d^k satisfy an angle condition, while the stepsizes t_k are efficient in view of Lemma 3.4. Together, using some standard results from [15], for example, this yields that the entire sequence $\{x^k\}$ converges to the unique minimum x^* of f. Since the Hessian of f is positive definite at x^* by Proposition 3.2, it also follows from the twice continuous differentiability of f on the level set $\mathcal{L}(x^0)$ that $\big(f(x^k + d^k) - f(x^k)\big)/(\nabla f(x^k)^T d^k \to 1/2$. Since $\sigma \in (0, 1/2)$, this implies that eventually the full stepsize is accepted. Hence our method inherits the local convergence properties of Newton's method and is therefore locally quadratically convergent. □

4 Numerical Results

Here we present some numerical results comparing the Weiszfeld algorithm with Algorithm 3.1. The Weiszfeld iteration uses the formula

$$x^{k+1} := T(x^k), \quad \text{where} \quad T(x) := \frac{1}{\sum_{i=1}^m \frac{\omega_i}{\|x - a_i\|}} \sum_{i=1}^m \frac{\omega_i a_i}{\|x - a_i\|},$$

and can alternatively be rewritten as a steepest descent method

$$x^{k+1} := x^k - t_k \nabla f(x^k) \quad \text{with stepsize} \quad t_k := \frac{1}{\sum_{i=1}^m \frac{\omega_i}{\|x^k - a_i\|}}.$$

In order to avoid problems with nondifferentiable points, we use the same initialization for Weiszfeld's method as in step (S.0) of Algorithm 3.1. Recall that this version of Weiszfeld's method is well-defined and globally convergent to a solution, cf. [1, 2].

We compare this method with Algorithm 3.1 using the parameters $\beta = 0.5$ and $\rho = 10^{-4}$. Since both methods calculate (explicitly or implicitly) the gradient of f, we use the same termination criterion $\|\nabla f(x^k)\| \leq \varepsilon$ for all examples. While Newton's method usually has no problems in achieving high accuracy, the Weiszfeld iteration is not locally fast convergent, and we therefore use the moderate value $\varepsilon = 10^{-5}$ to stop the iterations.

The actual implementation of Algorithm 3.1 replaces the monotone Armijo rule from step (S.3) by a nonmonotone version first suggested in [6]. This nonmonotone Armijo rule does not affect the local rate of convergence. Moreover, since our level sets $\mathcal{L}(x^0)$ are compact, it follows from the global convergence theory in [6] that Theorem 3.5 remains true also for the nonmonotone version of Algorithm 3.1.

In principle, the numerical behaviour of both Weiszfeld's method for the Fermat-Weber problem and Newton's method for (strongly convex) optimization problems are well-investigated. We therefore concentrate on a comparison of these two methods with a particular emphasis on the corresponding times that are required by both

methods. To this end, we use different values of n and m, and for each pair (n, m), we generate 100 test problems using MATLAB's `rand` command. The entries of all anchor points a_i were taken from the interval $[0, 100]$, and the same is true for all weigths ω_i.

All test runs were done on a Lenovo ThinkPad with an Intel Core i7 processor. In Table 1, we summarize the numerical results by presenting, for each pair (n, m), the following information: total elapsed times required by Weiszfeld's method and Algorithm 3.1, respectively, based on MATLAB's `tic` and `toc` commands, and the average number of iterations needed by these two methods. Note that all elapsed times are computed without the initialization phase from step (S.0), since this initialization is the same for both methods, so we only present the times for those parts of the two methods where they are actually different.

In our computations, it happened only a very few times (for small values of n and m) that an anchor point was already a solution. Furthermore, Newton's method solved all test problems, whereas Weiszfeld's method had one failure for $(n, m) = (3, 2)$ since the iteration limit $k_{\max} = 1000$ was reached. Of course, the number of iterations required by Newton's method is significantly less than the corresponding number in Weiszfeld's algorithm. Nevertheless, it is interesting to note that, in almost all examples, Algorithm 3.1 only needs two or three iterations. Furthermore, the nonmonotone Armijo rule allows to take full stepsizes $t_k = 1$ in essentially all test problems. This means that the nonmonotonicity also helps to save some time because no additional function evaluations are required in the inner loop to find a suitable stepsize.

The main observation is certainly the fact that Algorithm 3.1 is between three and eleven times faster than Weiszfeld's method. More precisely, in the most important case $n = 2$, our method is around ten times faster. The improvement decreases when n gets larger since the solution of the corresponding linear systems of equations is getting more time-consuming, but even for $n = 10$, the Newton-type method is about four times faster than Weiszfeld's method. Furthermore, for fixed n and increasing m, the efficiency of Newton's method in relation to the Weiszfeld fixed point iteration even increases. This has certainly to do with the way we compute our starting point for both methods in the initialization phase from step (S.0) of Algorithm 3.1: For larger m, it becomes more likely that we start both methods closer to a solution, which is obviously helpful for Weiszfeld's method, but especially for Newton's method since then we start closer to the local area where this method is quadratically convergent.

5 Final Remarks

This paper considers the Fermat-Weber location problem and shows that the standard Newton method can be used efficiently to solve this problem. Since this single-facility Fermat-Weber problem is also the basis of, for example, suitable methods for the solution of multi-facility location problems, it is an interesting question how this method can be adapted within a framework for solving multi-facility location problems. Moreover, the current method was designed to solve the classical Fermat-Weber problem based on the Euclidean norm, but the ideas might also apply to the

n		m = 10	m = 100	m = 10^3	m = 10^4
2	total time W.	0.0092	0.0420	0.3606	4.0287
	total time N.	0.0011	0.0044	0.0381	0.3630
	av. iter. W.	44.87	31.82	27.75	28.44
	av. iter. N.	2.96	3.02	2.99	2.68
3	total time W.	0.0085	0.0256	0.2319	2.2478
	total time N.	0.0012	0.0043	0.0336	0.2412
	av. iter. W.	40.33	18.97	17.93	18.92
	av. iter. N.	3.12	2.99	2.65	2.07
4	total time W.	0.0051	0.0258	0.1967	1.8662
	total time N.	0.0011	0.0057	0.0319	0.2478
	av. iter. W.	23.76	14.98	14.46	15.33
	av. iter. N.	3.03	2.96	2.38	2.02
5	total time W.	0.0054	0.0189	0.1711	1.6921
	total time N.	0.0012	0.0045	0.0309	0.2579
	av. iter. W.	23.68	12.96	12.71	13.38
	av. iter. N.	3.05	2.88	2.25	2.01
6	total time W.	0.0047	0.0167	0.1564	1.6972
	total time N.	0.0012	0.0042	0.0316	0.2927
	av. iter. W.	19.26	11.77	11.26	12.00
	av. iter. N.	2.98	2.75	2.22	2.03
7	total time W.	0.0039	0.0178	0.1448	1.4191
	total time N.	0.0011	0.0046	0.0297	0.2656
	av. iter. W.	18.33	10.88	10.45	11.01
	av. iter. N.	3.01	2.52	2.07	2.01
8	total time W.	0.0042	0.0149	0.1366	1.3525
	total time N.	0.0012	0.0039	0.0289	0.2614
	av. iter. W.	17.50	10.21	9.92	10.70
	av. iter. N.	2.99	2.36	2.04	2.01
9	total time W.	0.0035	0.0144	0.1289	1.2539
	total time N.	0.0013	0.0037	0.0292	0.2594
	av. iter. W.	15.85	9.50	9.17	10.00
	av. iter. N.	2.98	2.20	2.02	2.00
10	total time W.	0.0035	0.0133	0.1280	1.2507
	total time N.	0.0012	0.0036	0.0299	0.2674
	av. iter. W.	15.73	9.16	8.99	9.67
	av. iter. N.	2.96	2.21	2.00	2.00

Table 1: Numerical results for Weiszfeld's method and the Newton-type method from Algorithm 3.1. Abbreviations: W. = Weiszfeld, N. = Newton, av. iter. = average number of iterations

case of more general ℓ_p-norms.

References

[1] A. Beck: *Introduction to Nonlinear Optimization. Theory, Algorithms, and Applications with MATLAB.* MOS-SIAM Series on Optimization, Philadelphia, PA, 2014.

[2] A. Beck and S. Sabach: *Weiszfeld's method: Old and new results.* Journal of Optimization Theory and Applications 164, 2015, pp. 1–40.

[3] P.H. Calamai and A.R. Conn: *A projected Newton method for ℓ_p norm location problems.* Mathematical Programming 38, 1987, pp. 75–109.

[4] Z. Dreszner (ed.): *Facility Location. A Survey of Applications and Methods.* Springer, 1995.

[5] Z. Drezner, K. Klamroth, A. Schöbel, and G.O. Wesolowsky: *The Weber problem.* In Z. Dreszner and H.W. Hamacher (eds.): *Facility Location. Applications and Theory.* Springer, 2002, pp. 1–36.

[6] L. Grippo, F. Lampariello, and S. Lucidi: *A nonmonotone line search technique for Newton's method.* SIAM Journal on Numerical Analysis 23, 1986, pp. 707–716.

[7] F. Jarre and Ph.L. Toint: *Simple examples for the failure of Newton's method with line search for strictly convex minimization.* Mathematical Programming, to appear.

[8] J.-l. Jiang, K. Cheng, C.-C. Wang, and L.-p. Wang: *Accelerating the convergence in the single-source and multi-source Weber problems.* Applied Mathematics and Computation 218, 2012, pp. 6814–6824.

[9] I.N. Katz: *Local convergence in Fermat's problem.* Mathematical Programming 6, 1974, pp. 89–104.

[10] Y. Levin and A. Ben-Israel: *The Newton bracketing method for convex minimization.* Computational Optimization and Applications 21, 2002, pp. 213–229.

[11] R.F. Love, J.G. Morris and G.O. Wesolowsky: *Facilities Location. Models & Methods.* Elsevier Science Publishing Co., 1988.

[12] W.F. Mascarenhas: *Newton's iterates can converge to non-stationary points.* Mathematical Programming 112, 2008, pp. 327–334.

[13] Yu. Nesterov: *A method for solving a convex programming problem with rate of convergence $O(1/k^2)$.* Soviet Mathematics Doklady 269, 1983, pp. 543–547 (in Russian).

[14] S. Nickel and J. Puerto: *Location Theory: A Unified Approach.* Springer, Berlin, Heidelberg, 2005.

[15] J. Nocedal and S.J. Wright: *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering, Springer, New York, second edition 2006.

[16] M.L. Overton: *A quadratically convergent method for minimizing a sum of Euclidean norms.* Mathematical Programming 27, 1983, pp. 34–63.

[17] W. Sun and Y.-X. Yuan: *Optimization Theory and Methods. Nonlinear Programming.* Springer Optimization and its Applications, Springer, 2006.

[18] Y. Vardi and C.H. Zhang: *A modified Weiszfeld algorithm for the Fermat-Weber location problem.* Mathematical Programming 90, 2001, pp. 559–566.

[19] W. Warth and J. Werner: *Effiziente Schrittweitenfunktionen bei unrestringierten Optimierungsaufgaben.* Computing 19, 1977, pp. 59–72.