Newton’s Method for Multiobjective Optimization

Mauricio Graña Drummond,∗ Jörg Fliege † Rolando García Otero ‡ and B. F. Svaiter §

July 22, 2007

Abstract

We propose Newton’s Method for unconstrained multiobjective optimization (multicriteria optimization, vector optimization). The method does not scalarize the original vector optimization problem, and neither ordering information nor weighting factors for the different objective functions are assumed to be known. The objective functions are assumed to be twice continuously differentiable. Under these conditions, it is shown that the method converges to a point satisfying certain first-order necessary conditions for Pareto optimality. The speed of convergence is at least superlinear and quadratic in case the second derivatives are Lipshitz continuous. In the single objective case, we retrieve the classical Newton method.

Keywords: Multicriteria optimization, multi-objective programming, vector optimization, Pareto points, Newton’s method.

∗bolsigeno@gmail.com, Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brasil
†fliegej@bham.ac.uk, School of Mathematics, University of Southampton, Southampton SO17 1BJ, U.K.
‡rgarciga@impa.br, Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brasil
§benar@impa.br, Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brasil
1 Introduction

In multicriteria optimization, several objective functions have to be minimized simultaneously. Usually, no single point will minimize all given objective functions at once, and so the concept of optimality has to be replaced by the concept of Pareto-optimality or efficiency. A point is called Pareto-optimal or efficient, if there does not exist a different point with the same or smaller objective function values, such that there is a decrease in at least one objective function value. Applications for this type of problem can be found in engineering [15, 10, 28] (especially truss optimization [9], antenna design [27], space exploration [37]), statistics [5], management science [16, 2, 22, 33, 38, 1], environmental analysis [30, 17, 18], cancer treatment planning [25], etc.

One of the main solution strategy for multicriteria optimization problems is the scalarization approach, first introduced by Geoffrion [21]. Here, one or several parameterized single-objective (i.e. classical) optimization problems are solved, resulting in a corresponding number of Pareto-optimal points. The disadvantage of this approach is that the choice of the parameters is not known in advance, leaving the modeler and the decision-maker with the burden of choosing them. Only recently, adaptive scalarization techniques, where scalarization parameters are chosen automatically during the course of the algorithm such that a certain quality of approximation is maintained have been proposed [7, 23, 14, 19]. Still other techniques, working only in the bicriteria case, can be viewed as choosing a fixed grid in a particular parameter space [8, 9].

Multicriteria optimization algorithms that do not scalarize haven recently been developed (see, e.g., [4, 3] for an overview on the subject). Some of these techniques are extensions of scalar optimization algorithms (notably the steepest descent algorithm [20, 35, 12, 13] with at most linear convergence), while others borrow heavily from ideas developed in heuristic optimization [29, 32]. For the latter, no convergence proofs are known, and empirical results show that convergence generally is, as in the scalar case, quite slow [39]. Other parameter-free multicriteria optimization techniques use an ordering of the different criteria, i.e. an ordering of importance of the components of the objective function vector. In this case, the ordering has to be prespecified. Moreover, the corresponding optimization process is usually augmented by an interactive procedure [31], adding an additional burden to the task of the decision maker.
In this paper, we propose a parameter-free optimization method for computing a point satisfying first-order necessary conditions for multicriteria optimization. Neither ordering information nor weighting factors for the different objective functions is assumed to be known. The speed of convergence is at least superlinear for twice continuously differentiable functions and quadratic in case the second derivatives are Lipshitz continuous. In this, Newton’s method for scalar case is exactly mimicked.

The rest of this paper is as follows. Section 2 establishes the problem considered and the necessary notation. Section 3 introduces a first-order optimality condition for multiobjective optimization and derives a direction search program based on this. Such program is equivalent to an optimization problem with linear objective and convex quadratic constraints. Section 4 establishes the algorithm under consideration, while Section 5 contains the convergence results: superlinear convergence is discussed in Subsection 5.2, and quadratic convergence is discussed in Subsection 5.3. Numerical results shown in Section 6.2 show the applicability of our method.

2 Notation

Denote by $\mathbb{R}$ the set of real numbers, by $\mathbb{R}_+$ the set of nonnegative real numbers, and by $\mathbb{R}_+^+$ the set of strictly positive real numbers. Assume that $U \subset \mathbb{R}^n$ is an open set. The problem is to find a Pareto optimum of

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^m,$$

i. e. a point $z \in \mathbb{R}^n$ such that there does not exist a point $y \in \mathbb{R}^n$ with $F(y) \leq F(z)$ and $F(y) \neq F(z)$. Here, the inequality sign $\leq$ between vectors is to be understood in a componentwise sense. Likewise, in what follows, a strict inequality $F(y) < F(z)$ is to be understood componentwise, too. A point $z \in \mathbb{R}^n$ is called locally Pareto optimal, if there is a neighborhood $V \subseteq U$ of $z$ such that there does not exist a point $y \in V$ with $F(y) \leq F(z)$ and $F(y) \neq F(z)$. Note that each local Pareto optimal point is globally Pareto optimal as soon as all functions $F_i \ (i = 1, \ldots, m)$ are convex.

Throughout the paper, we will generally assume that $F \in C^2(U, \mathbb{R}^m)$ and that for all $x \in U$ we have

$$\nabla^2 F_j(x) > 0, \quad j = 1, \cdots, m.$$
Denote by $DF(x) \in \mathbb{R}^{m \times n}$ the Jacobian of the function $F$ at $x \in U$ and by $\nabla F_j(x)$ resp. $\nabla^2 F_j(x)$ the gradient resp. the Hessian of the function $F_j$ ($j = 1, \ldots, m$) at $x \in U$.

In what follows, the Euclidean norm in $\mathbb{R}^n$ will be denoted by $\| \cdot \|$, and we will use the same notation on all induced norms on corresponding matrix spaces. Let $B[x, r]$ denote the ball of radius $r$ with center $x \in \mathbb{R}^n$, and $I \in \mathbb{R}^{n \times n}$ the unit matrix. For an arbitrary matrix $M \in \mathbb{R}^{m \times n}$, we denote the image space of $M$ by $R(M) \subset \mathbb{R}^m$.

\section{The Newton Direction}

We start by characterizing Pareto-optimality or, more precisely, what we mean by a critical point $x \in U$.

**Theorem 1.** We have the following:

1. For general, nonconvex $F \in C^1(U, \mathbb{R}^m)$ we have the following. Let $x \in U$ be Pareto-optimal. Then,

   $$R(DF(x)) \cap (-\mathbb{R}^m_{++}) = \emptyset. \tag{1}$$

2. Let $F \in C^1(U, \mathbb{R}^m)$ be convex and $x \in U$. If (1) holds, then $x$ is Pareto-optimal.

**Proof.** Trivial. \hfill \Box

In general, we call a point $x \in U$ critical if at this point the equation (1) holds. This notion of criticality has already been used in [20] to define a parameter-free steepest descent algorithm for multiobjective optimization.

We proceed by defining what we mean by the Newton direction of the multi-objective problem under consideration. As in the classical one-criteria case, we characterize the Newton direction a solution to a suitably defined direction search program. Moreover, the direction obtained by solving this program at a critical point will be the zero direction $0 \in \mathbb{R}^n$.

For $x \in U$, let $\theta(x)$ and $s(x)$ be, respectively, the optimal value and the solution of the optimization problem

$$\begin{align*}
\begin{cases}
\min_{j=1, \ldots, m} & \max \nabla F_j(x)^T s + \frac{1}{2}s^T \nabla^2 F_j(x) s \\
\text{s.t.} & s \in \mathbb{R}^n.
\end{cases}
\end{align*} \tag{2}$$
Hence,
\[ \theta(x) = \inf_s \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s, \tag{3} \]

and
\[ s(x) = \arg \min_s \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s. \tag{4} \]

**Lemma 2.** We have the following:

1. The mappings \( x \mapsto s(x) \), and \( x \mapsto \theta(x) \) are continuous in \( U \).
2. For any \( x \in U \) we have \( \theta(x) \leq 0 \).
3. The conditions
   
   (a) \( R(DF) \cap (-\mathbb{R}^m_{++}) = \emptyset \),
   
   (b) \( \theta(x) = 0 \),
   
   (c) \( s(x) = 0 \)

   are equivalent.

**Proof.** Evident. \( \Box \)

**Lemma 3.** Let \( V \subset U \) be convex. Suppose that for any \( x, y \in V \) we have that if \( \|y - x\| < \delta \) holds, then
\[ \|\nabla^2 F_j(y) - \nabla^2 F_j(x)\| < \varepsilon \quad (j = 1, \ldots, m) \]
follows.

Under this assumption, for any \( x, y \in V \) such that \( \|y - x\| < \delta \) we have that
\[ \|\nabla F_j(y) - [\nabla F_j(x) + \nabla^2 F(x)(y - x)]\| < \varepsilon \|y - x\|, \]

and
\[ F_j(y) - \left[ F_j(x) + \nabla F_j(x)^T(y - x) + \frac{1}{2}(y - x)^T \nabla^2 F_j(x)(y - x) \right] < \frac{\varepsilon}{2} \|y - x\|^2 \]
holds for \( j = 1, \ldots, m \).
Corollary 4. Let there be given $x \in U$ and $0 < \sigma < 1$. If $\theta(x) < 0$, there exists $\bar{t} \in \mathbb{R}$ such that $0 < \bar{t} \leq 1$ and

$$x + ts(x) \in U, \quad F_j(x + ts(x)) \leq F_j(x) + \sigma t \theta(x), \quad \forall t \in [0, \bar{t})$$

holds for $j = 1, \cdots, m$.

Elementary duality theory applied to problem (2) shows that

$$\theta(x) = \sup_{\lambda \geq 0, \sum \lambda_j = 1} \inf_s \sum_{j=1}^m \lambda_j \left( \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s \right). \quad (5)$$

Therefore, there exists $\lambda(x) \in \mathbb{R}^m$ such that

$$\lambda(x) \geq 0, \quad \sum_{j=1}^m \lambda_j(x) = 1,$$

$$\theta(x) = \inf_s \sum_{j=1}^m \lambda_j(x) \left( \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s \right). \quad (6)$$

$$s(x) = \arg \min_s \sum_{j=1}^m \lambda_j(x) \left( \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s \right).$$

and

$$s(x) = - \left[ \sum_{j=1}^m \lambda_j(x) \nabla^2 F_j(x) \right]^{-1} \sum_{j=1}^m \lambda_j(x) \nabla F_j(x) \quad (7)$$

and we see that the Newton direction defined in this paper is a Newton direction for a standard scalar optimization problem, induced by weighting the given objective functions by nonnegative Lagrange multipliers. As a consequence, the standard weighting factors [21], well known in multiobjective programming, do show up in our approach, albeit implicitly. In particular, it is not necessary to fix such weights in advance, every point $x \in U$ defines such weights by way of the Lagrange multipliers in the corresponding direction search program.

4 The Algorithm

The algorithm for finding a Pareto point is now as follows.

1. (Initialization) Choose $x_0 \in U$, $0 < \sigma < 1$. Define $J = \{1/2^n | n = 0, 1, \cdots \} \cup \{0\}$ and set $k := 0$. 

6
2. **(Main loop)** For \( k = 0, 1, \ldots \)

(a) Solve the direction search program (2) to obtain \( s(x_k) \) and \( \theta(x_k) \).

(b) If \( \theta(x_k) = 0 \), then stop. Else, proceed to the line search, step 2. (c).

(c) **(Line Search)** Choose \( t_k \) as the largest \( t \in J \) such that

\[
x_k + ts(x_k) \in U, \quad F_j(x_k + ts(x_k)) \leq F_j(x_k) + \sigma t \theta(x_k), \quad j = 1, \ldots, m.
\]

(d) **(Update)** Define

\[
x_{k+1} = x_k + t_k s(x_k)
\]

and set \( k := k + 1 \). Go to step 2. (a).

**Theorem 5.** *The algorithm is well defined.*

*Proof.* This follows immediately with Corollary 4. \( \square \)

## 5 Convergence Results

The main results of this paper are presented in this section. First, some preliminary results are discussed in Subsection 5.1. Then, superlinear convergence is established in Subsection 5.2, before quadratic convergence is proven in Subsection 5.3.

### 5.1 Preliminary Results

The following auxiliary results provide estimates on the length of the Newton direction \( s(x) \) and on the optimal value of the direction search program, \( \theta(x) \).

**Lemma 6.** Take \( x \in U, \) \( 0 < a \leq b \). If

\[
aI \leq \nabla^2 F_j(x) \leq bI, \quad j = 1, \ldots, m,
\]

then

\[
\frac{a}{2} \| s(x) \|^2 \leq |\theta(x)| \leq \frac{b}{2} \| s(x) \|^2.
\]
Proof. Define
\[
H := \sum_{j=1}^{m} \lambda_j(x) \nabla^2 F_j(x), \quad v := \sum_{j=1}^{m} \lambda_j(x) \nabla F_j(x).
\]

Using (6) we have
\[
aI \leq H \leq bI, \quad s(x) = -H^{-1}v,
\]
and
\[
\theta(x) = v^T s(x) + \frac{1}{2} s(x)^T H s(x)
= (-Hs(x))^T s(x) + \frac{1}{2} s(x)^T H s(x)
= -\frac{1}{2} s(x)^T H s(x)
\]
Combining the two equations above, the conclusion follows. \qed

Lemma 7. Take \( x \in U, 0 < a \). If
\[
aI \leq \nabla^2 F_j(x), \quad j = 1, \ldots, m
\]
then
\[
|\theta(x)| \leq \frac{1}{2a} \left\| \sum_{j=1}^{m} \lambda_j \nabla F_j(x) \right\|^2
\]
for all \( \lambda_j \geq 0 \) (\( j = 1, \ldots, m \)).

Proof. Let \( \lambda_j \geq 0 \) (\( j = 1, \ldots, m \)) with \( \sum_{j=1}^{m} \lambda = 1 \) be given. Define \( v := \sum_{j=1}^{m} \lambda_j \nabla F_j(x) \). Then, by (5),
\[
0 \geq \theta(x) \geq \inf_{s} \sum_{j=1}^{m} \lambda_j(x) \left( \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s \right)
= \inf_{s} \left( v^T s + \sum_{j=1}^{m} \lambda_j \frac{1}{2} s^T \nabla^2 F_j(x) s \right)
\geq \inf_{s} \left( v^T s + \frac{1}{2} a \| s \|^2 \right) = -\frac{\| v \|^2}{2a}.
\]
\qed

8
5.2 Superlinear Convergence

**Theorem 8.** Denote by \((x_k)_k\) the sequence generated by the algorithm above. Suppose that \(V \subset U\), \(0 < \sigma < 1\), \(a, r, \delta, \varepsilon > 0\) and

(a) \(aI \leq \nabla^2 F_j(x)\) for all \(x \in V\), \(j = 1, \ldots, m\),
(b) \(\|\nabla^2 F_j(x) - \nabla^2 F(y)\| \leq \varepsilon\) for all \(x, y, \in V\) with \(\|x - y\| < \delta\),
(c) \(\varepsilon / a \leq (1 - \sigma)\),
(d) \(B[x_0, r] \subset V\),
(e) \(\|s(x_0)\| \leq \min \{\delta, r (1 - \varepsilon / a)\}\).

Then, we have the following for all \(k\):

1. \(\|x_k - x_0\| \leq \|s(x_0)\| \frac{1 - (\varepsilon / a)^k}{1 - \varepsilon / a}\)
2. \(\|s(x_k)\| \leq \|s(x_0)\| (\varepsilon / a)^k\)
3. \(t_k = 1\)
4. \(\|s(x_{k+1})\| \leq \|s(x_k)\| (\varepsilon / a)\).

Moreover, the sequence \((x_k)_k\) converges to some \(\bar{x} \in \mathbb{R}\) with

\[
\|\bar{x} - x_0\| \leq \frac{\|s(x_0)\|}{(1 - \varepsilon / a)} \leq r
\]

and \(R(DF(\bar{x})) \cap (-\mathbb{R}^m_{++}) = \emptyset\). The speed of convergence of \((x_k)_k\) is super-linear.

**Proof.** First we show that if items 1 and 2 hold for some \(k\), then items 3 and 4 also hold. Assume that item 1 and item 2 holds for some \(k\). Using this, we have

\[
\|(x_k + s(x_k)) - x_0\| \leq \|s_0\| \frac{1 - (\varepsilon / a)^{k+1}}{1 - \varepsilon / a}
\]

Hence,

\[x_k, x_k + s(x_k) \in B[x_0, r], \quad \|(x_k + s(x_k)) - x_k\| \leq \delta.\]
Using now Lemma 3 we conclude that, for \( j = 1, \cdots, m \),
\[
F_j(x_k + s(x_k)) \leq F_j(x_k) + \nabla F_j(x_k)^T s(x_k) + \frac{1}{2} s(x_k)^T \nabla F_j(x_k) s(x_k) + \frac{\varepsilon}{2} \| s(x_k) \|^2
\]
\[
\leq F_j(x_k) + \theta(x_k) + \frac{\varepsilon}{2} \| s(x_k) \|^2
\]
\[
= F_j(x_k) + \sigma \theta(x_k) + (1 - \sigma) \theta(x_k) + \frac{\varepsilon}{2} \| s(x_k) \|^2
\]

Since \( \theta(x_k) \leq 0 \), using Lemma 6 we obtain
\[
(1 - \sigma) \theta(x_k) + \frac{\varepsilon}{2} \| s(x_k) \|^2 \leq (\varepsilon - a(1 - \sigma)) \frac{\| s(x_k) \|^2}{2} \leq 0.
\]

Combining the above equations we conclude that item 3 holds:
\[ t_k = 1. \]

Therefore,
\[ x_{k+1} = x_k + s(x_k), \quad x_k, x_{k+1} \in B[x_0, r], \quad \| x_{k+1} - x_k \| \leq \delta. \]

Now define
\[
\bar{v}_{k+1} = \sum_{j=1}^m \lambda_j(x_k) \nabla F_j(x_{k+1}). \tag{9}
\]

Using Lemma 7, we have
\[
|\theta(x_{k+1})| \leq \frac{1}{2a} \| \bar{v}_{k+1} \|^2. \tag{10}
\]

To estimate \( \| \bar{v}_{k+1} \| \), define for \( x \in U \)
\[
G_k(x) := \sum_{j=1}^m \lambda_j(x_k) F_j(x), \tag{11}
\]

where \( \lambda_j(x_k) \) are the Lagrangian multipliers from (6) for \( x = x_k \). Observe that
\[ \bar{v}_{k+1} = G_k(x_{k+1}) = G_k(x_k + s(x_k)). \]
Direct calculation yields
\[ \nabla G_k(x) = \sum_{j=1}^{m} \lambda_j(x_k) \nabla F_j(x), \quad \nabla^2 G_k(x) = \sum_{j=1}^{m} \lambda_j(x_k) \nabla^2 F_j(x), \]
and
\[ s(x_k) = - \left[ \nabla^2 G_k(x_k) \right]^{-1} \nabla G_k(x_k), \quad (12) \]
cmp. (7). Note that
\[ \| \nabla^2 G_k(y) - \nabla^2 G_k(x) \| \leq \varepsilon, \quad \forall x, y \in V, \| y - x \| \leq \delta. \]
Therefore the estimate
\[ \| \nabla G_k(x_k + s(x_k)) - \left[ \nabla G_k(x_k) + \nabla^2 G_k(x_k) s(x_k) \right] \| \leq \varepsilon \| s(x_k) \|. \quad (13) \]
follows, and with (12) we arrive at
\[ \| \tilde{v}_{k+1} \| = \| \nabla G_k(x_{k+1}) \| \leq \varepsilon \| s(x_k) \|, \quad (14) \]
The combination of (14) with (10) leads us to
\[ | \theta(x_{k+1}) | \leq \frac{(\varepsilon \| s(x_k) \|)^2}{2a}, \]
and with an application of Lemma 6 we arrive at
\[ \frac{a}{2} \| s(x_{k+1}) \|^2 \leq \frac{(\varepsilon \| s(x_k) \|)^2}{2a}. \]
Therefore,
\[ \| s(x_{k+1}) \| \leq \| s(x_k) \| \frac{\varepsilon}{a}, \]
and item 4 also holds.
Now we will prove by induction that items 1 and 2 hold for all \( k \). For \( k = 0 \) they hold trivially. If items 1 and 2 hold for some \( k \) then items 3 and 4 also hold for such \( k \) and this implies that items 1 and 2 also hold for \( k + 1 \).
As items 1 and 2 hold for all \( k \), items 3 and 4 also hold for all \( k \).
Convergence of the sequence \( x_k \) to some \( \bar{x} \) and inequality (8) now follows trivially from item 2 and item 1. Since \( \| s(x_k) \| \) converges to 0, \( \theta(x_k) \) converges to 0 and \( \theta(\bar{x}) = 0 \). Hence \( R(D\bar{F}(\bar{x})) \cap (-\tilde{\mathbb{R}}^m_{++}) = \emptyset \).
To prove superlinear convergence, define
\[ r_k = \|s_0\| (\varepsilon/a)^k, \quad \delta_k = \|s_0\| (\varepsilon/a)^k, \quad k = 0, 1, \ldots \]

Using items 1 and 2 we conclude that
\[ B[x_k, r_k] \subset B[x_0, r]. \]

Take any \( \tau > 0 \) and define
\[ \hat{\varepsilon} = \min \left\{ a \frac{\tau}{1 + 2\tau}, \varepsilon \right\}. \]

For \( k \) large enough
\[ \|\nabla^2 F_j(x) - \nabla^2 F_j(y)\| \leq \hat{\varepsilon} \quad \forall x, y \in B[x_k, r_k], \|x - y\| \leq \delta_k. \]

Hence, assumptions (a)–(e) are satisfied for \( \hat{\varepsilon} \) and
\[ \hat{x}_0 = x_k, \quad \hat{r} = r_k, \quad \hat{\delta} = \delta_k \]

and \( B[\hat{x}_0, \hat{r}] \subset V \). Therefore, using item 4 we have
\[ \|s(x_{k+j})\| \leq \|s(x_k)\| (\hat{\varepsilon}/a)^j, \quad j = 1, 2, \ldots \]

which leads to
\[ \|\bar{x} - x_k\| \leq \|s(x_k)\| \frac{1}{1 - \hat{\varepsilon}/a} \]
as well as
\[ \|\bar{x} - x_{k+1}\| \leq \|s(x_{k+1})\| \frac{1}{1 - \hat{\varepsilon}/a} \leq \|s(x_k)\| \frac{\hat{\varepsilon}/a}{1 - \hat{\varepsilon}/a}. \]

Using the above inequalities and the triangle inequality get
\[ \|\bar{x} - x_k\| \geq \|x_{k+1} - x_k\| - \|\bar{x} - x_{k+1}\| \]
\[ \geq \|s(x_k)\| - \|s(x_k)\| \frac{\hat{\varepsilon}/a}{1 - \hat{\varepsilon}/a} \]
\[ = \|s(x_k)\| \frac{1 - 2\hat{\varepsilon}/a}{1 - \hat{\varepsilon}/a}. \]
Using the fact that $\hat{\varepsilon}/a < 1/2$ and the two inequalities above we obtain
\[
\|\bar{x} - x_{k+1}\| \leq \|\bar{x} - x_k\| \frac{\hat{\varepsilon}/a}{1 - 2\hat{\varepsilon}/a}
\]
which, combined with the definition of $\hat{\varepsilon}$ yields
\[
\|\bar{x} - x_{k+1}\| \leq \tau \|\bar{x} - x_k\|
\]

\[\square\]

**Corollary 9.** If $R(DF(x)) \cap (-R^m_+) = \emptyset$, then there exists an $r > 0$ such that, for all $x_0 \in B[x, r] \subset U$, the algorithm generates a sequence which converges superlinearly to some $\bar{x}$ with $\theta(\bar{x}) = 0$.

**Corollary 10.** If $x_0 \in U$ is in a compact level set of $F$, then the algorithm generates from $x_0$ a sequence which converges superlinearly to some $\bar{x}$ with $\theta(\bar{x}) = 0$.

### 5.3 Quadratic Convergence

**Theorem 11.** Suppose that, in addition to all assumptions of Theorem 8, we have

(f) $\nabla^2 F_j$ Lipshitz continuous on $V$ with Lipshitz constant $L$ ($j = 1, \ldots, m$).

Then there exists a $k_0$ such that $\tau_k := (L/a)\|s(x_k)\|^2 < 1/2$ for all $k \geq k_0$, and we have
\[
\|\bar{x} - x_{k+1}\| \leq \frac{L}{a} \frac{(1 - \tau_k)}{(1 - 2\tau_k)^2} \|\bar{x} - x_k\|^2
\]

**Proof.** Due to item 4 of Theorem 8, it is clear that $(L/a)\|s(x_k)\|^2 < 1/2$ for $k$ large enough. We follow the proof of Theorem 8 by defining vectors $\tilde{v}_{k+1}$ as in (9) and functions $G_k$ as in (11). If all $\nabla^2 F_j$ are Lipshitz continuous, $\nabla^2 G_k$ is also Lipshitz continuous. Therefore, we can refine (13) to
\[
\|\tilde{v}_{k+1}\| \leq L\|s(x_k)\|^2,
\]
and (10) leads to
\[
|\theta(x_{k+1})| \leq \frac{1}{2a} L^2 \|s(x_k)\|^4.
\]
Therefore,
\[
(a/2)\|s(x_{k+1})\|^2 \leq \frac{1}{2a} L^2 \|s(x_k)\|^4.
\]
\[
\|s(x_{k+1})\| \leq \frac{L}{a} \|s(x_k)\|^2.
\]

Let \( k \geq k_0 \). Then,
\[
\|\bar{x} - x_k\| \leq \sum_{j=k+1}^{\infty} \|s(x_j)\| \leq (L/a)\|s(x_k)\|^2 \sum_{j=0}^{\infty} \tau^j_k = (L/a)\|s(x_k)\|^2 \frac{1}{1 - \tau_k}.
\]

Hence
\[
\|\bar{x} - x_k\| \geq \|x_{k+1} - x_k\| - \|\bar{x} - x_{k+1}\|
\geq \|s(x_k)\| - \|s(x_k)\|^2 \frac{L/a}{1 - \tau_k}
= \|s(x_k)\| \frac{1 - 2\tau_k}{1 - \tau_k}
\]
and
\[
\|\bar{x} - x_{k+1}\| \leq (L/a) \left[ \|\bar{x} - x_k\| \frac{1 - \tau_k}{1 - 2\tau_k} \right]^2 \frac{1}{1 - \tau_k} = \|\bar{x} - x_k\|^2 \frac{(L/a)(1 - \tau_k)}{(1 - 2\tau_k)^2}.
\]

\[\square\]

**Corollary 12.** If all \( \nabla^2 F_j \) are Lipshitz continuous and if \( R(DF(x)) \cap (-R^m_{+}) = \emptyset \), then there exists an \( r > 0 \) such that, for all \( x_0 \in B[x, r] \subset U \), the algorithm generates a sequence which converges quadratically to some \( \bar{x} \) with \( \theta(\bar{x}) = 0 \).

**Corollary 13.** If \( x_0 \in U \) is in a compact level set of \( F \) on which all all \( \nabla^2 F_j \) are Lipshitz continuous, then the algorithm generates from \( x_0 \) a sequence which converges quadratically to some \( \bar{x} \) with \( \theta(\bar{x}) = 0 \).

### 6 Numerical Results

#### 6.1 Implementation Issues

Of course, the assumption \( \nabla^2 F(x) > 0 \) is quite restrictive. Once we have superlinear or quadratic convergence of the (almost) “pure” version of Newton’s Method, is natural to deal with regularized versions. As in classical
optimization, the idea is to use a quadratic regularization to enforce (local) convexity. The regularized Newton step is the solution of

\[
\begin{aligned}
\min \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s + \frac{\mu}{2} \|s\|^2 \\
\text{s.t.} \quad s \in \mathbb{R}^n.
\end{aligned}
\tag{15}
\]

where \(\mu > 0\) is large enough to enforce convexity of the above problem. In this setting, Hence,

\[
\theta_{\mu}(x) = \inf_s \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s + \frac{\mu}{2} \|s\|^2,
\tag{16}
\]

and

\[
s_{\mu}(x) = \arg\min_s \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T \nabla^2 F_j(x) s + \frac{\mu}{2} \|s\|^2.
\tag{17}
\]

etc.

6.2 Test Results

A MATLAB prototype implementation of the method described in the last sections was tested on various problems from the literature. Since many of the test problems considered are of nonconvex nature, the direction search program was augmented by an additional constraint of the form \(\|d\|_{\text{inf}} \leq 2\). For all test cases, the value \(\delta = 5 \times 10^{-5}\) for the stopping criterion and \(\sigma = 0.1\) for the line search was used, and all cases consider bicriteria problems, i.e. \(m = 2\) holds. The implementation was tested on various problems from the literature, whose main characteristics are outlined in Table 1. While all problems are \(C^\infty\) on their domain, note that only the problems JOS1 and ZDT1 are convex, so we are mainly concerned with the amount of work (iterations, function evaluations) it takes in finding local Pareto points by using our local search method. All problems where solved 200 times using starting points from a uniform random distribution between a lower bound \(U \in \mathbb{R}^n\) and an upper bound \(L \in \mathbb{R}^n\). Average number of iterations and average number of function evaluations are reported in Table 2.

We proceed by discussing our findings some further by taking a closer look at some of the test problems and the corresponding results. Problem No. 1 is a simple convex quadratic test function used by various authors to benchmark algorithms. As it can be seen, this problem poses no
<table>
<thead>
<tr>
<th>Name</th>
<th>$n$</th>
<th>$U^T$</th>
<th>$L^T$</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOS1</td>
<td>*100</td>
<td>$[-2,\ldots,2]$</td>
<td>$[2,\ldots,2]$</td>
<td>[26, Ex. 1]</td>
</tr>
<tr>
<td>Hil</td>
<td>2</td>
<td>$[0,\ldots,0]$</td>
<td>$[5,\ldots,5]$</td>
<td>[24]</td>
</tr>
<tr>
<td>ZDT1</td>
<td>*50</td>
<td>$[1,\ldots,1]/100$</td>
<td>$[1,\ldots,1]$</td>
<td>[39, Ex. 1]</td>
</tr>
<tr>
<td>ZDT2</td>
<td>*50</td>
<td>$[0,\ldots,0]$</td>
<td>$[1,\ldots,1]$</td>
<td>[39, Ex. 2]</td>
</tr>
<tr>
<td>ZDT3</td>
<td>*50</td>
<td>$[1,\ldots,1]/100$</td>
<td>$[1,\ldots,1]$</td>
<td>[39, Ex. 3]</td>
</tr>
<tr>
<td>PNR</td>
<td>2</td>
<td>$[-2,\ldots,2]$</td>
<td>$[2,\ldots,2]$</td>
<td>[34]</td>
</tr>
<tr>
<td>JOS4</td>
<td>*50</td>
<td>$[1,\ldots,1]/100$</td>
<td>$[1,\ldots,1]$</td>
<td>[26, Ex. 4]</td>
</tr>
<tr>
<td>SD</td>
<td>4</td>
<td>$[1,\sqrt{2},\sqrt{2},1]$</td>
<td>$[3,\ldots,3]$</td>
<td>[36]</td>
</tr>
<tr>
<td>Deb</td>
<td>2</td>
<td>$[0.1,1]$</td>
<td>$[1,1]$</td>
<td>[11]</td>
</tr>
<tr>
<td>LTDZ</td>
<td>3</td>
<td>$[0,0,0]$</td>
<td>$[1,1,1]$</td>
<td>[29]</td>
</tr>
</tbody>
</table>

Figure 1: Main data for the problem instances considered. A '*' marker indicates that the parameter can be chosen by the user.

<table>
<thead>
<tr>
<th>prob.</th>
<th>iter</th>
<th>feval</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOS1</td>
<td>4.88</td>
<td>7.38</td>
</tr>
<tr>
<td>Hil</td>
<td>2.63</td>
<td>7.55</td>
</tr>
<tr>
<td>ZDT1</td>
<td>4.01</td>
<td>4.91</td>
</tr>
<tr>
<td>ZDT2</td>
<td>4.45</td>
<td>14.60</td>
</tr>
<tr>
<td>ZDT3</td>
<td>3.12</td>
<td>5.64</td>
</tr>
<tr>
<td>PNR</td>
<td>3.19</td>
<td>4.13</td>
</tr>
<tr>
<td>JOS4</td>
<td>6.19</td>
<td>6.44</td>
</tr>
<tr>
<td>SD</td>
<td>2.25</td>
<td>12.29</td>
</tr>
<tr>
<td>Deb</td>
<td>3.66</td>
<td>14.47</td>
</tr>
<tr>
<td>LTDZ</td>
<td>2.44</td>
<td>12.61</td>
</tr>
</tbody>
</table>

Figure 2: Average number of iterations (iter) and average number of function evaluations (feval) for the problems from Table 1. Note that the number of gradient and Hessian evaluations is equal to the number of function evaluations.
challenges to the method proposed here: on average, less than 5 iterations
where necessary to find a Pareto point, and the number of function eval-
uations shows that the simple Armijo-like step size rule employed here is
successful either immediately or after one steps/ Some further insights can
also be obtained here. Consider Problem No. 1 in the dimensions $n = 10$
and $n = 50$ with 100 randomly chosen starting points as above. Figure 3
shows the value space ($\mathbb{R}^n = \mathbb{R}^2$) of the problem and in there the computed
values of the efficient points found as well as the trajectories starting at the
corresponding starting points.
Efficient points where generated with an average of 2.73 iteration steps and
4.43 function evaluations for $n = 10$, as well as 3.41 iteration steps and
3.61 function evaluations for $n = 50$, showing that the method is fairly
robust. However, Figure 3 clearly reveals that while 100 random starting
points are enough to produce a reasonable cover of the set of efficient points
in the $n = 10$ case, the same number of starting points fails to produce a
reasonable approximation in the $n = 50$ case. Similar behaviour has to be
expected for other problems.
Problem No. 2 is a nonconvex, albeit smooth multicriteria problem. As it
can be seen from Figure 4, our method is able to generate a rough outline of
the set of efficient points (cmp. [24]), but it also gets stuck in local efficient
points.
Likewise, Figure 5 shows clearly that, given a reasonable number of starting
Figure 4: Value space of Problem No. 2 for 150 random starting points. Circled points indicate local efficient points that are not globally efficient.
Figure 5: Value space of Problem No. 6 for 150 random starting points. Circled points indicate local efficient points that are not globally efficient.

points, the method is able to identify the set of local, nonglobal Pareto points as well as the set of global Pareto points for a problem (Problem No. 6) in which both cases occur.

Finally we tested the algorithm on a convex problem with three criteria and
<table>
<thead>
<tr>
<th>$n$</th>
<th>iter</th>
<th>feval</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.74</td>
<td>14.02</td>
</tr>
<tr>
<td>10</td>
<td>3.29</td>
<td>11.90</td>
</tr>
<tr>
<td>20</td>
<td>3.37</td>
<td>12.26</td>
</tr>
<tr>
<td>50</td>
<td>13.78</td>
<td>76.97</td>
</tr>
</tbody>
</table>

Figure 6: Average number of iterations (iter) and average number of function evaluations (feval) for the three-criteria problem defined by (18)–(20).

variable dimension $n$ defined by

\[
F_1(x) := \sum_{k=1}^{n} k(x_k - k)^4, \quad (18)
\]

\[
F_2(x) := \exp \left( \sum_{k=1}^{n} \frac{x_k}{n} \right) + \|x\|_2^2, \quad (19)
\]

\[
F_3(x) := \frac{1}{n(n+1)} \sum_{k=1}^{n} k(n-k+1)e^{x_k} \quad (20)
\]

with $x \in [0, 1]^n$. Results for various dimensions $n$ can be found in Table 6

7 Final Remarks

7.1 Variations

7.2 Extensions

The constrained case, as well as the multicriteria case where the order is induced by a general convex pointed cone is a topic of further research.

References


    1968.

[22] M. Gravel, I. M. Martel, R. Madeau, W. Price, and R. Tremblay. A mul-
    ticriterion view of optimal resource allocation in job-shop production.

    Application in Multicriteria Optimization (in german)*. PhD thesis, De-
    partment of Mathematics, University of Dortmund, Germany, 2005.

    Homotopy Approach*, volume 25 of *International Series of Numerical

    of antennas for treatment planning in hyperthermia. Preprint 265,
    Institut für Angewandte Mathematik, Universität Erlangen-Nürnberg,

[26] Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. Dynamic weighted
    aggregation for evolutionary multi-objective optimization: Why does it
    work and how? In *Proceedings of the Genetic and Evolutionary Com-

[27] A. Jüschke, J. Jahn, and A. Kirsch. A bicriterial optimization problem of
    antenna design. *Computational Optimization and Applications*, 7:261–

[28] Renata Kasperska. Bi-criteria Optimization Of Open Cross-Section Of
    The Thin-Walled Beams With Flat Flanges. Presentation at 75th An-
    nual Scientific Conference of the Gesellschaft für Angewandte Mathe-
    matik und Mechanik e. V., Dresden, Germany, March 2004.

    and diversity in evolutionary multiobjective optimization. *Evolutionary


