Noname manuscript No.
(will be inserted by the editor)

Complexity iteration analysis for strongly convex multi-objective optimization using a Newton path-following procedure

El-Houcine Bergou · Youssef Diouane · Vyacheslav Kungurtsev

Received: date / Accepted: date

Abstract In this note, we consider the iteration complexity of solving strongly convex multi-objective optimization problems. We discuss the precise meaning of this problem, noting that its definition is ambiguous, and focus on the most natural notion of finding a set of Pareto optimal points across a grid

E. Bergou

KAUST, Thuwal, Saudi Arabia.

MaIAGE, INRAE, Université Paris-Saclay, 78350 Jouy-en-Josas, France.

E-mail: elhoucine.bergou@inrae.fr

Y. Diouane

ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse Cedex 4, France.

E-mail: youssef.diouane@isae-supaero.fr

V. Kungurtsev

Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague.

E-mail: vyacheslav.kungurtsev@fel.cvut.cz

of scalarized problems. We prove that, in most cases, performing sensitivity based path-following after obtaining one solution is the optimal strategy for this task in terms of iteration complexity.

Keywords Multi-objective optimization · strongly convex optimization · path-following · Newton method · complexity analysis.

Mathematics Subject Classification (2010) 49M05 · 49M15 · 90C06 · 90C60.

1 Introduction

Consider the following multi-objective optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x) := \{f_i(x)\}_{i=1,\dots,m} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$ is a strongly convex and twice continuously differentiable function. Our target is to find *weak Pareto-optimality* points for problem (1). We recall that *weak Pareto-optimality* holds at a point $\widetilde{x} \in \mathbb{R}^n$ if for all $d \in \mathbb{R}^n$, there exists an $i \in \{1, \dots, m\}$ such that

$$\nabla f_i(\widetilde{x})^\top d \geq 0.$$

For single objective optimization, *worst-case iteration complexity* quantifies the number of iterations that could be necessary, in the worst-case (i.e., for the most ill-behaved problems), for an algorithm to achieve a certain level of satisfaction of an approximate measure of optimality, typically a small norm for the gradient [9]. Classically, the multi-objective optimization community had

not considered attempting to derive bounds on iteration complexity for problems in vector optimization. Recently, however, works have appeared, see [6,3], which consider the iteration complexity of gradient descent for multi-objective optimization. In both papers rates were derived for obtaining some point satisfying approximate weak Pareto-optimality.

To the best of our knowledge, only [6] considers iteration complexity for specifically strongly convex objectives. However, in deriving their complexity result, convergence of the algorithm to some Pareto optimal point is assumed. Moreover, we believe that a much stronger and more meaningful result can be shown by considering the precise meaning of the problem. In particular, consider the so-called scalarized problem, parametrized by $\{\lambda_i\}_{i=1,..,m}$

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{m} \lambda_i f_i(x), \tag{2}$$

for any $\{\lambda_i\}_{i=1,..,m} \in \mathcal{D} = \{\{\lambda_i\}_{i=1,..,m} \in \mathbb{R}^m \ / \ 0 \leq \lambda_i \leq 1, \sum_{i=1}^{m} \lambda_i = 1\}$, the unit simplex of $\mathbb{R}^m$. A stationary point of the problem (2) is also Pareto optimal for (1). Thus, one can find a Pareto optimal point, at least for strongly convex multi-objective problems, by simply choosing any arbitrary convex combination $\{\lambda_i\}_{i=1,..,m} \in \mathcal{D}$ and solving the resulting mono-objective problem, thus the worst-case iteration complexity of finding *some* Pareto optimal point is already a known problem, it corresponds to the worst-case iteration complexity of solving a single objective strongly convex optimization problem.

In the multi-objective optimization literature, e.g., [8], scalarization is typically, at most, one step in the process of finding the solution of a multi-objective problem, where the ultimate definition of a solution can vary. In particular, it

can be that the goal of the optimization is (a) tracing the Pareto front itself, so in some sense finding all, or some adequate approximation to all, stationary points, or (b) finding an appropriately *best* point along the Pareto front through some secondary metrics, or (c) using an interactive environment with a human participant who grades potential solutions.

In this note, we shall concern ourselves with the first task: establish complexity bounds for some appropriate notion of finding the entire Pareto front. To this end, we define the problem as, for all $\lambda \in \Lambda$, finding

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \lambda_i f_i(x) \tag{3}$$

where $\Lambda \subset \mathcal{D}$ is some finite grid of elements. In the definition of $\mathcal{D}$, given the constraint on the sum, we can consider $m-1$ dimensions as free which in turn entirely determine the remaining λ_i. We thus divide each side of the hypercube $[0,1]$ by some desired width of the grid d, and thus there are $\lfloor \frac{1}{d} \rfloor^{m-1}$ total possible grid points, where $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a. Conversely, if there is some desired grid pre-defined by the user, we can define the quantity d denoting the maximum width between two neighbors on the grid, that is

$$d = \min_{(\lambda,\lambda') \in \Lambda \ , \ \lambda \neq \lambda'} \left(\max_{1 \leq i \leq m} |\lambda_i - \lambda_i'| \right).$$

We would like to emphasize that the procedure of obtaining a Pareto front by solving a set of scalarized single-objective reformulations is suited strictly to *strongly convex* objectives, in the non-convex case this strict correspondence between the two may be lost.

We organize this note as follows. In Section 2, we describe our algorithm to solve the strongly convex multi-objective optimization problem. We explain how to use a Newton path-following procedure to find the entire Pareto front. Section 3 addresses the convergence of our algorithm by characterizing its iteration complexity. A numerical illustration on a simple example demonstrating the efficiency of our approach is given in Section 4. Conclusions are given in Section 5.

2 Path-following for finding the entire Pareto front.

Recall that for a fixed $\lambda^0 \in \Lambda$, using only first order information one can solve a strongly convex optimization problem of the type (3) at best linearly (with a gradient descent based method, see for instance [2, Theorem 3.18]). Namely, in order to obtain a point ϵ distance from the solution of (3) for a fixed $\lambda^0 \in \Lambda$, $\mathcal{O}(\log(1/\epsilon))$ iterations must be taken, with each iteration involving the computation of one gradient vector. As a result, naively, one can obtain the entire Pareto front by solving each of the $\lfloor \frac{1}{d} \rfloor^{m-1}$ scalarized problems defined across the grid points independently with a variant of gradient descent, to obtain an overall complexity of $\mathcal{O}\left(\log(1/\epsilon) \lfloor \frac{1}{d} \rfloor^{m-1}\right)$.

In this note, we propose finding the entire Pareto front by performing path-following, a method based on the implicit function theorem. Later we will show that the proposed strategy will reduce the overall iteration complexity drastically relative to naively solving every scalarized problem separately. To start with, for some initial grid point $\lambda^{(0)} \in \Lambda$ we obtain the solution $x^{(0)} \in \mathbb{R}^n$

of the problem (3), (for instance, by using a gradient descent method with the following stopping criterion $\left\|\sum_{j=1}^{m} \lambda_j^{(0)} \nabla f_i(x)\right\| \leq \epsilon$). Note that such point $x^{(0)}$ gives the Pareto optimal point of the problem (1) associated with $\lambda^{(0)}$. Now, let $\lambda^{(1)}$ be one of the closest neighbors to $\lambda^{(0)}$ in the finite grid Λ, our goal is to apply a predictor-corrector scheme to compute a new $x^{(1)}$ corresponding to an approximate solution to the scalarized problem associated with $\lambda^{(1)}$.

Path-following, or tracing a set of solutions for a parametrized nonlinear system of equations across a range of parameters, is an important algorithmic tool, for which an introduction can be found in [1]. Closest to our work, a predictor-corrector path-following procedure for strongly convex optimization problems (interpreted as strongly regular variational inequalities) is given in [5]. In this work it is shown that, for this parametric problem, a property of uniform strong regularity holds and a procedure involving one tangential predictor (Euler) and one corrector (Newton) step result in a sequence of iterates whose distance to a set of solutions to the parametric variational inequality is of the order of d^4, where recall that d is, in this context, the grid spacing. Thus there exists C such that if $d \leq C(\epsilon)^{1/4}$, a set of solutions with approximate optimality ϵ across a set of parameters can be found. If applied to the multi-objective Pareto front context, the number of Euler-Newton continuation steps is the number of grid points, which corresponds to $d^{-1} \geq C^{-1}\epsilon^{-1/4}$. If the desired grid is already small enough, then it is clear that this path-following procedure outperforms the naive method of solving the standalone

Algorithm 1: A Newton path-following for finding the Pareto front.

Input: Let $\Lambda = \{\lambda^{(0)}, \ldots, \lambda^{(p-1)}\} \subset \mathbb{R}^m_+$ be some finite grid of cardinality p satisfying: for all $j = 0, \ldots, p-1$, $\sum_i^m \lambda_i^{(j)} = 1$, $\lambda^{(j+1)}$ is one of the closest neighbors of $\lambda^{(j)}$ not yet visited.

Output: The entire Pareto front by performing path-following associated with Λ: $x^{(0)}, x^{(1)}, \ldots, x^{(p-1)}$.

Compute an initial Pareto optimal point $x^{(0)}$, i.e.,

$$x^{(0)} = \arg\min_x f^{(0)}(x), \text{ where } \quad f^{(0)}(x) = \sum_{i=1}^m \lambda_i^{(0)} f_i(x). \tag{4}$$

Set $k = 0$.

Step 1: Compute a predictor $\bar{x}^{(k+1)}$, i.e.,

$$\bar{x}^{(k+1)} = x^{(k)} - \left[\sum_{j=1}^m \lambda_j^{(k)} \nabla^2 f_j\left(x^{(k)}\right)\right]^{-1} \left(\sum_{i=1}^m \left(\lambda_i^{(k+1)} - \lambda_i^{(k)}\right) \nabla f_i\left(x^{(k)}\right)\right) \tag{5}$$

Step 2: Apply a Newton correction to compute $x^{(k+1)}$, i.e., starting from $\bar{x}^{(k+1)}$ run the Newton method to find

$$x^{(k+1)} = \arg\min_x f^{(k+1)}(x), \text{ where } \quad f^{(k+1)}(x) = \sum_{i=1}^m \lambda_i^{(k+1)} f_i(x). \tag{6}$$

If $k = p-1$ **then** Stop, **otherwise** increment k by 1 and go to **Step 1**.

problem at every grid point. Otherwise, it depends on the magnitude of the desired number of additional grid points required to perform path-following.

We consider an alternative predictor-corrector scheme that is more aggressive in its use of potentially longer tangential steps and multiple Newton iterations. In particular, this is more suitable for obtaining the set of solutions across the Pareto front with the tightest iteration complexity bound. This predictor-corrector procedure will be repeated until we traverse the entire set Λ. A formal description of the algorithm is given as Algorithm 1.

The predictor step (**Step 1** of Algorithm 1) is motivated by the implicit function theorem. Therefore, first, let's recall the implicit function theorem adapted to our context.

Theorem 1 *Let $g : \mathbb{R}^{n+m} \to \mathbb{R}^n$ be a continuously differentiable function for a parametrized system of equations,*

$$g(x, \lambda) = 0, \quad \text{where } x \in \mathbb{R}^n \text{ and } \lambda \in \mathbb{R}^m.$$

Consider that there exists a solution satisfying $g(x_0, \lambda_0) = 0$. If the Jacobian matrix $J_{g,x}(x_0, \lambda_0)$ of g with respect to x is invertible, then there exists an open neighborhood $\mathcal{B} \subset \mathbb{R}^m$ such that there exists a unique continuously differentiable path $\widetilde{x}(\lambda)$ defined on $\lambda \in \mathcal{B}$ with $\widetilde{x}(\lambda_0) = x_0$ and $g(\widetilde{x}(\lambda), \lambda) = 0$ for all $\lambda \in \mathcal{B}$. Furthermore, it holds that the derivative of $\widetilde{x}(\lambda)$ over $\mathcal{B}$ is given by

$$\frac{\partial \widetilde{x}}{\partial \lambda}(\lambda) = -\left[J_{g,x}(\widetilde{x}(\lambda), \lambda)\right]^{-1} \frac{\partial g}{\partial \lambda}(\widetilde{x}(\lambda), \lambda). \tag{7}$$

We consider applying Theorem 1 to the optimality conditions of (2) given by the following parametrized system of equations

$$g(x, \lambda) = \sum_{i=1}^{m} \lambda_i \nabla f_i(x) = 0.$$

Precisely, for a given iteration index k, consider that we have a solution $x^{(k)}$ at some $\lambda^{(k)} \in \Lambda$, i.e.,

$$\sum_{j=1}^{m} \lambda_j^{(k)} \nabla f_i(x^{(k)}) = 0. \tag{8}$$

Since we consider strongly convex objectives, it holds that the matrix

$$\sum_{i=1}^{m} \lambda_i^{(k)} \nabla^2 f_i(x)$$

is invertible for all $x \in \mathbb{R}^n$, in particular the inverse norm is bounded by the inverse of the weighted sum of the strong convexity constants of $\{f_i\}$. Thus by Theorem 1 we have that there exists some ball $\mathcal{B}^{(k)}$ around $\lambda^{(k)}$ and a unique path $\widetilde{x}(\lambda)$ such that $\widetilde{x}(\lambda^{(k)}) = x^{(k)}$ and $\sum_{i=1}^m \lambda_i \nabla f_i(\widetilde{x}(\lambda)) = 0$, for all $\lambda \in \mathcal{B}^{(k)}$. Furthermore, the derivative of the path given by (7) is defined for all $\lambda \in \mathcal{B}^{(k)}$ to satisfy,

$$\frac{\partial \widetilde{x}}{\partial \lambda}(\lambda) = -\left[\sum_{j=1}^m \lambda_j \nabla^2 f_j\left(\widetilde{x}(\lambda)\right)\right]^{-1} \left[\nabla f_1\left(\widetilde{x}(\lambda)\right), \ldots, \nabla f_m\left(\widetilde{x}(\lambda)\right)\right].$$

Consider now a Taylor expansion of $\widetilde{x}(\lambda)$ along $\lambda \in \mathcal{B}^{(k)}$ from the base point $\tilde{x}(\lambda^{(k)}) = x^{(k)}$. This is given by

$$\widetilde{x}(\lambda) = x^{(k)} - \left[\sum_{j=1}^m \lambda_j^{(k)} \nabla^2 f_j\left(\left(x^{(k)}\right)\right)\right]^{-1} \sum_{i=1}^m \left(\lambda_i - \lambda_i^{(k)}\right) \nabla f_i\left(x^{(k)}\right) + \mathcal{O}\left(\left\|\lambda - \lambda^{(k)}\right\|^2\right).$$

Motivated by the discussion on Newton's method applied to path-following in [4, Chapter 5], we define a predictor $\bar{x}^{(k)}(\lambda)$ by computing the first order Taylor approximation,

$$\bar{x}^{(k)}(\lambda) = x^{(k)} - \left[\sum_{j=1}^m \lambda_j^{(k)} \nabla^2 f_j\left(\left(x^{(k)}\right)\right)\right]^{-1} \sum_{i=1}^m \left(\lambda_i - \lambda_i^{(k)}\right) \nabla f_i\left(x^{(k)}\right) \tag{9}$$

which is precisely the "tangent continuation method" with the order $p = 2$ as given in [4, Page 239]. Assuming that $\lambda^{(k+1)}$ is close enough to $\lambda^{(k)}$ (i.e., $\lambda^{(k+1)} \in \mathcal{B}^{(k)}$), the predictor step $\bar{x}^{(k+1)}$ given by (5) in Algorithm 1 is defined as

$$\bar{x}^{(k+1)} = \bar{x}^{(k)}\left(\lambda^{(k+1)}\right).$$

Let $\eta^{(k)}$ be the norm of the residual $\widetilde{x}(\lambda^{(k+1)}) - \bar{x}^{(k+1)}$, i.e.,

$$\eta^{(k)} = \|\widetilde{x}(\lambda^{(k+1)}) - \bar{x}^{(k+1)}\|.$$

There is a remaining algorithmic necessity before this becomes practical as the predictor $\bar{x}^{(k+1)}$ does not necessarily satisfy the desired level of stationarity. In fact, to achieve a point closer to the actual solution, we consider a "corrector" step $x^{(k+1)}$ using a sequence of Newton iterations. Under a set of conditions, the ordinary Newton method is quadratically convergent towards the solution starting from the predicted point if this point is sufficiently close to the solution. Thus, we require that the predictor $\bar{x}^{(k+1)}$ is sufficiently accurate and determine the size of the step $\lambda^{(k+1)} - \lambda^{(k)}$ appropriately. Note that by definition of $\widetilde{x}$ we have

$$x^{(k+1)} = \widetilde{x}(\lambda^{(k+1)}).$$

3 Characterizing the Complexity of Algorithm 1

Based on the ideas above, we can consider iteration complexity in a new sense. For a given iteration k, consider having an approximate solution to (3) for a particular $\lambda^{(k)}$, up to a desired optimality tolerance ϵ. Then consider path-following from $\lambda^{(k)}$ to some $\lambda^{(k+1)}$ where $\lambda^{(k+1)} - \lambda^{(k)}$ is small enough (in terms of desired grid-spacing d) to be able to determine the associated solution on the Pareto front. The same procedure is repeated across all the grid Λ until all solutions of the Pareto front have been found.

Before developing our complexity analysis, we formally state our working assumptions on the objective function f.

Assumption 31 *The objective function $f : \mathbb{R}^n \to \mathbb{R}^m$ is twice continuously differentiable and strongly convex. In particular, there exist two positive con-*

stants $c > 0$ and $L > 0$, such for all $i \in \{1, \ldots, m\}$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$,

$$c\|y\|^2 \leq y^\top \nabla^2 f_i(x) y \leq L\|y\|^2. \tag{10}$$

In other words, for all $i \in \{1, \ldots, m\}$, the eigenvalues of the Hessian of f_i are uniformly bounded from below by c, and above by L, everywhere.

This implies the following condition regarding scaling invariance properties appropriate for Newton methods [4].

Lemma 1 *Consider Assumption 31. For all $\lambda \in \Lambda$ and $x, y \in \mathbb{R}^n$, one has*

$$\left\| \left(\sum_{j=1}^m \lambda_j \nabla^2 f_j(x) \right)^{-1} \left(\sum_{j=1}^m \lambda_j \nabla f_j(y) - \sum_{j=1}^m \lambda_j \nabla f_j(x) \right) \right\| \leq \frac{L}{c} \|x - y\|. \tag{11}$$

In this case, the mapping $x \to \sum_{i=1}^p \lambda_i \nabla^2 f_i(x)$ is said to be affine covariant Lipschitz.

Proof Let $\lambda \in \Lambda$, then from Assumption 31, we conclude that the mapping $x \to \sum_{i=1}^m \lambda_i f_i(x)$ is L-smooth (i.e, its gradient is Lipschitz with constant L) and that it is c-strongly convex. In fact, for all x and $y \in \mathbb{R}^n$, we have that this mapping satisfies

$$\left\| \left(\sum_{j=1}^m \lambda_j \nabla f_j(y) - \sum_{j=1}^m \lambda_j \nabla f_j(x) \right) \right\| \leq L\|x - y\|,$$

and that the inverse of its Hessian for all x is bounded by $\frac{1}{c}$, i.e.,

$$\left\| \left(\sum_{j=1}^m \lambda_j \nabla^2 f_j(x) \right)^{-1} \right\| \leq \frac{1}{c}.$$

By combining these two inequalities with the Cauchy Schwarz inequality, we get (11). □

In the next Lemma, we will show that the remainder $\eta^{(k)}$ is at most of the same order as the distance between $\lambda^{(k+1)}$ and $\lambda^{(k)}$. This will be instrumental in giving sufficient conditions on the grid spacing necessary to ensure quadratic local convergence of the ordinary Newton method when it is applied to solve (6) starting from $\bar{x}^{(k+1)}$ (see Lemma 3).

Lemma 2 *Consider Assumption 31. Then there exists a constant $\eta > 0$ such that for all k, one has*

$$\eta^{(k)} \leq \eta \|\lambda^{(k+1)} - \lambda^{(k)}\|.$$

Proof In fact, one has

$$\begin{aligned}
\eta^{(k)} &= \|x^{(k+1)} - \bar{x}^{(k+1)}\| \\
&\leq \|x^{(k+1)} - x^{(k)}\| + \left\| \left[\frac{\partial \widetilde{x}}{\partial \lambda}(\lambda^{(k)}) \right]^{\top} (\lambda^{(k+1)} - \lambda^{(k)}) \right\| \\
&\leq \|\widetilde{x}(\lambda^{(k+1)}) - \widetilde{x}(\lambda^{(k)})\| + \max_{\lambda \in \Lambda} \left\| \frac{\partial \widetilde{x}}{\partial \lambda}(\lambda) \right\| \|\lambda^{(k+1)} - \lambda^{(k)})\|.
\end{aligned}$$

On the other hand, by Theorem 1, the function $\lambda \to \widetilde{x}(\lambda)$ is continuously differentiable for all $\lambda \in \mathcal{D}$, thus, it is bounded and Lipschitz continuous over the compact set $\mathcal{D}$, i.e., there exists $c_1 > 0$ and $c_2 > 0$ such that

$$\|\widetilde{x}(\lambda^{(k+1)}) - \widetilde{x}(\lambda^{(k)})\| \leq c_1 \|\lambda^{(k+1)} - \lambda^{(k)})\| \text{ and } \max_{\lambda \in \Lambda} \left\| \frac{\partial \widetilde{x}}{\partial \lambda}(\lambda) \right\| \leq c_2.$$

Hence,

$$\eta^{(k)} \leq (c_1 + c_2) \|\lambda^{(k+1)} - \lambda^{(k)})\|,$$

which completes the proof. □

The next result will show that to ultimately get an ϵ-Pareto optimal solution, applying the correction step will require a number of iterations of the

ordinary Newton method of order $\log\log\left(\frac{1}{\epsilon}\right)$. We will start by recalling some sufficient conditions concerning the local convergence of the Newton method adapted to our setting. These results are from [4]. In [4, Theorem 5.2], the authors gave sufficient conditions on the functions $\{f_j(\cdot)\}$ and the distance between $\lambda^{(k+1)}$ and $\lambda^{(k)}$ to ensure the convergence of the ordinary Newton method applied to solve (6) starting from $\bar{x}^{(k+1)}$, and in [4, Theorem 2.3] the authors showed the classic local quadratic convergence property of the Newton method when it is used to solve (6). In fact under the strong convexity assumption and as long as the distance between $\lambda^{(k+1)}$ and $\lambda^{(k)}$ is sufficiently small, the ordinary Newton method converges quadratically to the minimizer of (6) from the starting point $\bar{x}^{(k+1)}$.

Lemma 3 *Let Assumption 31 hold. For a given iteration index k, consider $\lambda^{(k)} \in \Lambda$ and $x^{(k)}$ such that*

$$\left\| \sum_{j=1}^{m} \lambda_j^{(k)} \nabla f_i\left(x^{(k)}\right) \right\| \leq \epsilon.$$

Let $\lambda^{(k+1)} \in \Lambda$ such that

$$\left\| \lambda^{(k+1)} - \lambda^{(k)} \right\| \leq \frac{c}{\eta L}, \tag{12}$$

where η is as in Lemma 2. Then the ordinary Newton method with the starting point $\bar{x}^{(k+1)}$ (as given by (5)) converges and the computational cost of achieving a solution point $x^{(k+1)}$ such that

$$\left\| \sum_{j=1}^{m} \lambda_j^{(k+1)} \nabla f_i\left(x^{(k+1)}\right) \right\| \leq \epsilon,$$

is of order $\log\log\left(\frac{1}{\epsilon}\right)$.

Proof Let $\left[x^{(k+1)}\right]_j$ be the j^{th} iterate produced by an ordinary Newton method starting from $\bar{x}^{(k+1)}$. By Lemma 1, the mapping $x \to \sum_{i=1}^{p} \lambda_i^{(k+1)} \nabla^2 f_i(x)$ is affine covariant Lipschitz. Hence, using [4, Theorem 5.2] and with $\bar{x}^{(k+1)}$ being the predictor step defined as (5), one can deduce that the sequence $\left\{\left[x^{(k+1)}\right]_j\right\}_{j\in\mathbb{N}}$ generated by the ordinary Newton method starting from $\bar{x}^{(k+1)}$ converges towards the solution $x^{(k+1)}$ (i.e., $x^{(k+1)} = \lim_{j\to\infty}\left[x^{(k+1)}\right]_j$). Using Lemma 2 and (12), one gets

$$\left\|\left[x^{(k+1)}\right]_0 - x^{(k+1)}\right\| = \left\|\bar{x}^{(k+1)} - x^{(k+1)}\right\| = \eta^{(k)} \leq \eta\|\lambda^{(k+1)} - \lambda^{(k)}\| \leq \frac{c}{L}. \tag{13}$$

In this case, using [4, Theorem 2.3], the Newton method converges quadratically, i.e.,

$$\left\|\left[x^{(k+1)}\right]_{j+1} - x^{(k+1)}\right\| \leq \frac{L}{2c}\left\|\left[x^{(k+1)}\right]_j - x^{(k+1)}\right\|^2.$$

Hence,

$$\left\|\left[x^{(k+1)}\right]_j - x^{(k+1)}\right\| \leq \left(\frac{L}{2c}\right)^{2^{j-1}}\left\|\left[x^{(k+1)}\right]_0 - x^{(k+1)}\right\|^{2^j}.$$

Thus, using (13), one deduces that

$$\left\|\left[x^{(k+1)}\right]_j - x^{(k+1)}\right\| \leq \frac{c}{L}2^{-2^{j-1}}.$$

Thus,

$$\begin{aligned}\left\|\sum_{j=1}^{m}\lambda_j^{(k)}\nabla f_i\left(\left[x^{(k+1)}\right]_j\right)\right\| &= \left\|\sum_{j=1}^{m}\lambda_j^{(k)}\nabla f_i\left(\left[x^{(k+1)}\right]_j\right) - \sum_{j=1}^{m}\lambda_j^{(k)}\nabla f_i\left(x^{(k+1)}\right)\right\| \\ &\leq L\left\|\left[x^{(k+1)}\right]_j - x^{(k+1)}\right\| \leq c2^{-2^{j-1}}.\end{aligned}$$

This implies that the computational cost of achieving the desired level of stationarity is of order $\log\log\left(\frac{1}{\epsilon}\right)$. □

Thus the iteration complexity of Algorithm 1 is just of the order of complexity for solving a standalone strongly convex problem (i.e., computing $x^{(0)}$)

added with $\frac{1}{d^{m-1}}$ multiplied by the cost of a predictor and the iterated Newton step. We formalize this with the following theorem,

Theorem 2 *Let Assumption 31 hold. Define N_ϵ to be the number of iterations required to obtain $x^{(0)}$, a point that has distance at most ϵ from the optimal point corresponding to (2) at $\lambda^{(0)}$. Assume that the maximum width between any two neighbors on the grid Λ is,*

$$d \leq \min\left(\frac{c}{\eta L}, \bar{d}\right), \tag{14}$$

with $\bar{d}$ the minimal desired distance between lattice points.

Then, the overall iteration complexity of Algorithm 1 is

$$N_\epsilon + \mathcal{O}\left(\left\lfloor\frac{1}{d}\right\rfloor^{m-1} \log\log\left(\frac{1}{\epsilon}\right)\right).$$

Proof First, note that, for each iteration k of Algorithm 1, the complexity of the predictor step is constant as its computational cost does not depend on ϵ. For the corrector Newton step, since one has

$$\|\lambda^{(k+1)} - \lambda^{(k)}\| \leq d \leq \frac{c}{\eta L},$$

Lemma 3 implies that the complexity of running the Newton iterations until approximate optimality is $O\left(\log\log\left(\frac{1}{\epsilon}\right)\right)$. The proof is thus completed since the total number of lattice points in the grid Λ is at most $\left\lfloor\frac{1}{d}\right\rfloor^{m-1}$.

Note that by using a gradient solver, the first term N_ϵ is of order $\log(1/\epsilon)$. Hence, one can see that the complexity is generally favorable compared to the naive method of solving the strongly convex problem at every grid point separately, as $\log\log\left(\frac{1}{\epsilon}\right) \ll \log(1/\epsilon)$ for small ϵ.

Remark 1 Note that both the naive method of solving every problem across the grid points and path-following are both about equally parallelizable with perfect speedup as long as the number of grid points is larger than the number of processors. We can split the grid into disjoint components, and each processor finds one point in its part of the convex hull of allowable $\{\lambda_i\}$ and proceeds to path-follow across section of the grid assigned to it.

4 Numerical Illustration

To show the numerical performance of our approach compared to the naive method (which corresponds to the Gradient Descent method applied sequentially to the set of problems (2) defined for varying λ), we consider a simple problem given in [7], defined by

$$f(x) = \left[(x_1 - 1)^2 + (x_1 - x_2)^2, (x_2 - 3)^2 + (x_1 - x_2)^2\right]^\top.$$

Since we have two objective functions, the vector λ has two components λ_1 and λ_2 where $\lambda_1 + \lambda_2 = 1$. In our experiment, we discretize λ in a uniform grid with a grid step-size d (the desired distance between the lattice points).

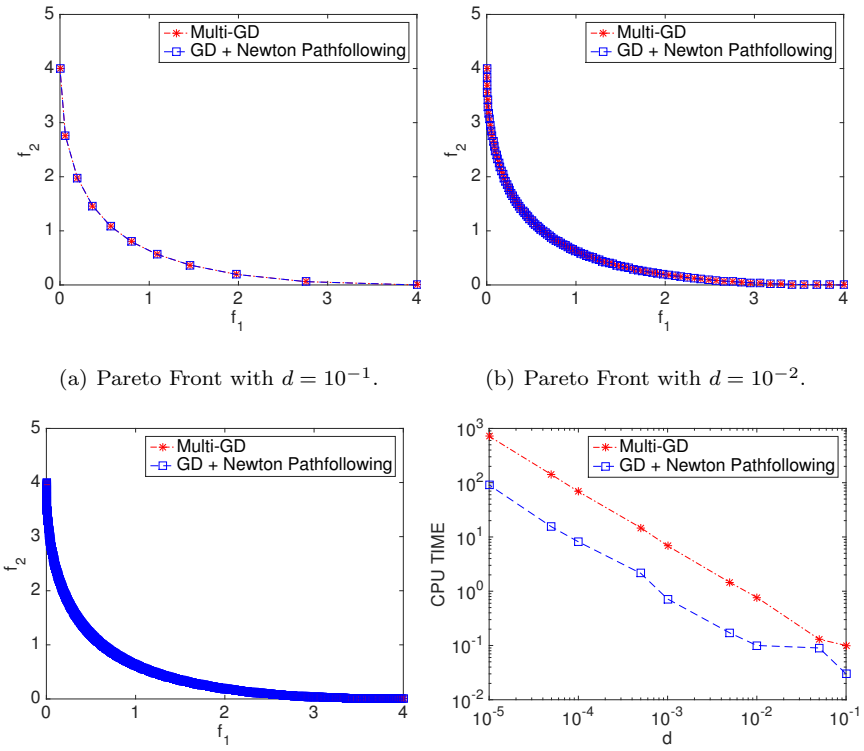
In our Matlab illustration, we will call **Multi-GD** the naive method and **GD+Newton Pathfollowing** the implementation of our Algorithm 1 (where we used the standard Gradient Descent method to find the first Pareto optimal point and then apply the Newton path-following procedure). For the Gradient Descent method, we used a random initial point x_0 and a stepsize equal to

$1/\lambda_{\max}$ where $\lambda_{\max}$ is the maximum eigenvalue for the Hessians of f_1 and f_2. We stopped the methods when the norm of the gradient was less than 10^{-7}.

The obtained results are shown in Figure 1. One can see that both methods are able to find a similar Pareto Front (independently of the value of the grid spacing d). In term of the elapsed CPU time to find the front, our proposed algorithm can be seen to be faster than the naive method. In particular, one can see that for some values of d, the method **GD+Newton Pathfollowing** can obtain the approximately optimal solution with up to 10 times faster total run-time than the **Multi-GD** method. We conducted other experiments (not reported here) on many toy problems and in all of them our method was outperforming the naive method in run-time while finding essentially the same front.

5 Conclusion

In this note, we studied the iteration complexity of a class of strongly convex multi-objective optimization problems. We observed that the notion of iteration complexity is not uniquely defined as there can be varying possible criteria of what it means to solve a multi-objective optimization problem. By working with the most context-independent criterion (namely, finding the set of all Pareto optimal points on a front), we demonstrated that finding the solution of one scalarized problem and then path-following across the grid to obtain the others is superior to finding the solution of every problem independently.



(a) Pareto Front with $d = 10^{-1}$.

(b) Pareto Front with $d = 10^{-2}$.

(c) Pareto Front with $d = 10^{-3}$.

(d) CPU time in seconds.

Fig. 1 Pareto Front and CPU time comparison, using **Multi-GD** and **GD+ Newton Path-following**, for different values of d.

Acknowledgements

We would like to thank two anonymous referees for their careful readings and corrections that helped us to improve our manuscript significantly. E. Bergou received support from the AgreenSkills+ fellowship programme which has received funding from the EU's Seventh Framework Programme under grant agreement No FP7-609398 (AgreenSkills+ contract). V. Kungurtsev received support from the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

References

1. E. L. Allgower and K. Georg. *Introduction to numerical continuation methods*, volume 45. SIAM, 2003.
2. S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8:231–357, 2015.
3. L. Calderòn, M. A. Diniz-Ehrhardt, and J. M. Martinez. On high-order model regularization for multiobjective optimization. *Optim. Methods Softw.*, 2020, https://doi.org/10.1080/10556788.2020.1719408.
4. P. Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.
5. A. L. Dontchev, M. I. Krastanov, R. T. Rockafellar, and V. M. Veliov. An Euler–Newton continuation method for tracking solution trajectories of parametric variational inequalities. *SIAM J. Control Optim.*, 51:1823–1840, 2013.
6. J. Fliege, A. I. F. Vaz, and L. N. Vicente. Complexity of gradient descent for multiobjective optimization. *Optim. Methods Softw.*, 34:949–959, 2019.
7. S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comp.*, 10:477–506, 2006.
8. R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Struct. Multidisciplinary Optim.*, 26:369–395, 2004.
9. Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.