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

Adaptive multi-gradient methods for quasiconvex vector optimization and applications to multi-task learning

Tran Ngoc Thang · **Nguyen Anh Minh**

February 25, 2023

Abstract For solving a broad class of nonconvex multiobjective programming problems on an unbounded constraint set, we provide an adaptive step-size strategy that does not include line-search techniques and establish convergence of a generic approach under mild assumptions. Specifically, the objective function may not satisfy the convexity condition. It does not need a previously established Lipschitz constant for determining an initial step-size, as is the case with descent line-search algorithms. The crucial feature of this process is the steady reduction of the step size until a certain condition is fulfilled. In particular, it may be used to provide a novel multi-gradient projection approach to unbounded constrained optimization problems. The correctness of the method is verified by preliminary results from some computational examples. To demonstrate the effectiveness of the proposed technique for large scale problems, we apply it to some experiments on multi-task learning.

Keywords nonconvex multi-objective programming · gradient descent algorithms · quasiconvex functions · pseudoconvex functions · adaptive step-sizes

Mathematics Subject Classification (2000) 90C25 · 90C26 · 68Q32 · 93E35

1 Introduction

Gradient descent methods are a common tool for a wide range of programming problems, from convex to nonconvex, for both scalar and vector optimization, and have numerous practical applications (see [2], [5], [11] and references therein). At each iteration, gradient descent algorithms provide an iterative series of solutions based on gradient directions and step sizes. For a long time, researchers have focused on finding the direction to improve the

Tran Ngoc Thang
School of Applied Mathematics and Informatics, Hanoi University of Science and Technology, 1st Dai Co Viet street, Hanoi, Viet Nam
E-mail: thang.tranngoc@hust.edu.vn
Nguyen Anh Minh
School of Applied Mathematics and Informatics, Hanoi University of Science and Technology, 1st Dai Co Viet street, Hanoi, Viet Nam
E-mail: minh.na194117@hust.edu.vn

convergence rate of techniques, while the step-size was determined using one of the few well-known approaches (see [2], [15]).

Multicriteria optimization algorithms that do not scalarize have recently been developed (see, e.g., [?] for an overview on the subject). Some of these techniques are extensions of scalar optimization algorithms, such as notably the steepest descent algorithm [?] with at most linear convergence based on the gradient descent method, while others borrow heavily from ideas developed in heuristic optimization [?]. For the latter, no convergence proofs are known, and empirical results show that convergence generally is, as in the scalar case, quite slow [?].

Recently, new major areas of machine learning applications with high dimensionality and nonconvex objective functions have required the development of novel step-size choosing procedures to reduce the method's overall computing cost (see [5], [11]). The exact or approximate one-dimensional minimization line-search incurs significant computational costs per iteration, particularly when calculating the function value is nearly identical to calculating its derivative and requires the solution of complex auxiliary problems (see [2]). To avoid the line-search, the step-size value may be calculated using prior information such as Lipschitz constants for the gradient. However, this requires using just part of their inexact estimations, which leads to slowdown convergence. This is also true for the well-known divergent series rule (see [9], [15]).

In this research, we propose a novel and line-search-free adaptive step-size algorithm for a broad class of multiobjective programming problems where the objective function is nonconvex smooth and the constraint set is unbounded closed convex. A crucial component of this procedure is gradually decreasing the step size until a predetermined condition is fulfilled. Although the Lipschitz continuity of the gradient of the objective function is required for convergence, the approach does not make use of predetermined constants. The proposed change has been shown to be effective in preliminary computational tests. We perform various machine learning experiments, including multi-task learning and neural networks for classification, to show that the proposed method performs well on large-scale tasks.

The rest of this paper is structured as follows. Section 2 provides preliminaries and details the problem formulation. Section 3 summarizes the primary results, including the proposed algorithms. Section 4 depicts numerical experiments and analyzes computational outcomes. Section 5 presents applications to certain machine learning problems. The final section makes some conclusions.

2 Preliminaries

2.1 Notations and definitions

Denote by $\mathbb{R}$ the set of real numbers, by $\mathbb{R}_+$the set of non-negative real numbers, and by $\mathbb{R}_{++}$the set of strictly positive real numbers. Assume that $U \subset \mathbb{R}^n$ is a nonempty, closed and convex set, and $F : U \longrightarrow \mathbb{R}^m$ is a given function.

Definition 1 The problem is to find *an efficient point* or *Pareto optimum* of F, i.e., a point $x^* \in U$ such that $\nexists y \in U, F(y) \leq F(x^*)$, and $F(y) \neq F(x^*)$, where the inequality sign $\leq$ between vectors is to be understood in a componentwise sense.

In effect, we are employing the partial order induced by $\mathbb{R}^m_+ = \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$ (the nonnegative orthant or Paretian cone of $\mathbb{R}^m$) defined by

$$F(y) \leq F(z) \Longleftrightarrow F(z) - F(y) \in \mathbb{R}^m_+,$$

and we are searching for minimal points induced by such partial order.

Definition 2 A point x^* is *weakly efficient* or *weak Pareto optimum* if

$$\nexists y \in U, \quad F(y) < F(x^*),$$

where the vector strict inequality $F(y) < F(x^*)$ is to be understood componentwise too.

This relation is induced by $\mathbb{R}^m_{++} = \mathbb{R}_{++} \times \cdots \times \mathbb{R}_{++}$, the interior of the Paretian cone $(F(y) < F(z)$ if, and only if, $F(z) - F(y) \in \mathbb{R}^m_{++})$. Later on, we will make use of the negative of $\mathbb{R}^m_{++}$, i.e,

$$-\mathbb{R}^m_{++} = \left\{-v : v \in \mathbb{R}^m_{++}\right\}.$$

Definition 3 A point $x^* \in U$ is *locally efficient* (respectively, *locally weakly efficient*) if there is a neighborhood $V \subseteq U$ of x^* such that the point x^* is efficient (respectively, weakly efficient) for F restricted to V. Locally efficient points are also called *local Pareto optimal*, and locally weakly efficient points are also called *local weak Pareto optimal*.

Note that if U is convex and F is $\mathbb{R}^m_+$-convex (i.e., if F is componentwise-convex), then each local Pareto optimal point is globally Pareto optimal. Clearly, every locally efficient point is locally weakly efficient.

Definition 4 We say that a point $x^* \in U$ is a *Pareto stationary* point (or a *Pareto critical* point) of F if

$$DF(x^*)(U - x^*) \cap \left(-\mathbb{R}^m_{++}\right) = \emptyset, \tag{1}$$

where $DF(x^*)$ is the Jacobian matrix of F at x^* given by $DF(x) = (\nabla f_1(x), \cdots, \nabla f_m(x))^\top$. This notion is necessary (but in general not sufficient) for a point to be weak Pareto efficient and was first used in [15] to investigate a steepest descent algorithm. Note that in the case when $m = 1$, (1) is reduced to the classical optimality condition in scalar optimization.

Definition 5 [14] The differentiable function $f : \mathbb{R}^m \to \mathbb{R}$ is said to be

i) convex on U if for all $x, y \in U$, $\lambda \in [0,1]$, it holds that

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

ii) pseudoconvex on U if for all $x, y \in U$, it holds that

$$\langle \nabla f(x), y - x \rangle \geq 0 \Rightarrow f(y) \geq f(x).$$

iii) quasiconvex on U if for all $x, y \in U$, $\lambda \in [0;1]$, it holds that

$$f(\lambda x + (1-\lambda)y) \leq \max\{f(x); f(y)\}.$$

Proposition 1 *[6] The differentiable function f is quasiconvex on U if and only if*

$$f(y) \leq f(x) \Rightarrow \langle \nabla f(x), y - x \rangle \leq 0.$$

It is worth mentioning that "f is convex" $\Rightarrow$ "f is pseudoconvex" $\Rightarrow$ "f is quasiconvex" [14].

Proposition 2 *When F is pseudoconvex, i.e. the components of f are pseudoconvex, then* (1) *is a necessary and sufficient condition for a point to be weakly efficient.*

Indeed, suppose to the contrary, that there exists $y \in U$ such that $F(y) \prec F(x)$. Since F_i is pseudoconvex for each $i = 1,\ldots,m$, it follows that $\langle \nabla F_i(x), (y-x)\rangle < 0$ and therefore $DF(x)(y-x) \in -\mathbb{R}^m_{++}$, contradicting (2). This result, in general, does not hold for quasi-convex functions; see [?].

The range, or image space, of a matrix $M \in \mathbb{R}^{m\times n}$ will be denoted by $R(M)$ and $I \in \mathbb{R}^{n\times n}$ will denote the unit matrix. For two matrices $A, B \in \mathbb{R}^{n\times n}, B \leq A (B < A)$ will mean $A - B$ positive semidefinite (definite). In what follows, the Euclidean norm in $\mathbb{R}^n$ will be denoted by $\|\cdot\|$, and $B[x,r]$ denotes the closed ball of radius r with center $x \in \mathbb{R}^n$. We will use the same notation $\|\cdot\|$ for the induced operator norms on the corresponding matrix spaces.

For $x \in \mathbb{R}^m$, denote by $P_U(x)$ the projection of x onto U, i.e.,

$$P_U(x) := \operatorname{argmin}\{\|z-x\| : z \in U\}.$$

Proposition 3 [1] *It holds that*

(i) $\|P_U(x) - P_U(y)\| \leq \|x-y\|$ *for all* $x, y \in \mathbb{R}^m$,
(ii) $\langle y - P_U(x), x - P_U(x)\rangle \leq 0$ *for all* $x \in \mathbb{R}^m$, $y \in U$.

Proposition 4 *[6] Suppose that ∇f is L-Lipschitz continuous on U. For all $x, y \in U$, it holds that*

$$|f(y) - f(x) - \langle \nabla f(x), y-x\rangle| \leq \frac{L}{2}\|y-x\|^2.$$

Lemma 1 *[16] Let $\{a_k\}$; $\{b_k\} \subset (0;\infty)$ be sequences such that*

$$a_{k+1} \leq a_k + b_k \ \forall k \geq 0; \quad \sum_{k=0}^{\infty} b_k < \infty.$$

Then, there exists the limit $\lim_{k\to\infty} a_k = c \in \mathbb{R}$.

2.2 Problem statements

Throughout the paper, unless explicitly mentioned, we will assume that F is a differentiable function on an open set containing U and the Jacobian $DF \in \mathbb{R}^{m\times n}$ of F is Lipschitz continuous, that is the gradient $\nabla F_j \in \mathbb{R}^n$ of the function F_j is Lipschitz continuous for all $j = 1, ..., m$. We consider the optimization problem:

$$\operatorname{Min}_{x\in U} F(x). \qquad \text{(MOP}(F,U)\text{)}$$

Assume that the solution set of (MOP(F,U)) is not empty.

3 The adaptive multi-gradient methods

We now proceed by defining the adaptive steepest multi-gradient methods for the multi-objective problem under consideration.

For $x \in U$, we define $s(x)$, the steepest direction at x, as the optimal solution of

$$\begin{cases} \min \ \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T s \\ \text{s.t.} \ \ s \in U(x), \end{cases} \tag{2}$$

where $U(x) = U - x = \{s \in \mathbb{R} \mid s = u - x \text{ for some } u \in U\}$. If $U = \mathbb{R}^n$ then $U(x) = \mathbb{R}^n$.

First of all, observe that problem (2) always has a unique minimizer, since the functions $\nabla F_j(x)^T s + \frac{1}{2} s^T s$ are strongly convex for $j = 1, \ldots, m$. Also note that for $m = 1$, the direction $s(x)$ is the "classical" steepest direction for scalar optimization, which is $-\nabla F(x)$.

Here, we are approximating

$$\max_{j=1,\ldots,m} F_j(x+s) - F_j(x)$$

by the maximum of the local quadratic models at x of each F_j. Some comments are in order. First, note that in a standard scalarization approach, the multicriteria problem is replaced by a problem of minimizing a scalar function $x \mapsto \varphi(F_1(x), \ldots, F_m(x))$, where φ might depend on some parameters. This is not the case in the approach taken here: We solve a scalar optimization problem to find a direction of descent for all objective functions involved. After a corresponding descent step, another, usually different, scalar optimization problem will be solved, and so on.

It is also important to remark that the idea of minimizing the maximum of quadratic approximations of the component functions variations in order to obtain a search direction has already been proposed in an algorithm for order-value optimization problems by Andreani et al. in [2], where at each iteration only the "ε-active" component functions are considered.

Search directions obtained by the minimization of the maximum of linear approximations (regularized by a quadratic term) of the components variations were previously considered for defining steepest descent-like methods for multiobjective optimization [**?**] and for vector optimization [**?**]. The optimal value of problem (2) will be denoted by $\theta(x)$. Hence,

$$\theta(x) = \inf_{s \in U(x)} \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T s, \tag{3}$$

and

$$s(x) = \arg\min_{s \in U(x)} \max_{j=1,\ldots,m} \nabla F_j(x)^T s + \frac{1}{2} s^T s \tag{4}$$

Although (2) is a nonsmooth problem, it can be framed as a convex quadratic optimization problem and so, it can be effectively solved. Indeed, (2) is equivalent to

$$\begin{cases} \min & g(t,s) = t \\ \text{s.t.} & \nabla F_j(x)^T s + \frac{1}{2} s^T s - t \leq 0 \quad (1 \leq j \leq m) \\ & (t,s) \in \mathbb{R} \times U(x). \end{cases} \tag{5}$$

The Lagrangian of this problem is

$$L((t,s),\lambda) = t + \sum_{j=1}^{m} \lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s - t \right).$$

Direct calculation of the Karush-Kuhn-Tucker conditions yields

$$\sum_{j=1}^{m} \lambda_j = 1, \quad \sum_{j=1}^{m} \lambda_j (\nabla F_j(x) + s) = 0, \tag{6}$$

$$\lambda_j \geq 0, \quad \nabla F_j(x)^T s + \frac{1}{2} s^T s \leq t \quad (1 \leq j \leq m), \tag{7}$$

$$\lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s - t \right) = 0 \quad (1 \leq j \leq m). \tag{8}$$

Problem (5) has a unique solution, $(\theta(x), s(x))$. As this is a convex problem and has a Slater point (e.g., $(1,0)$), there exists a KKT multiplier $\lambda = \lambda(x)$, which, together with $s = s(x)$ and $t = \theta(x)$, satisfies conditions (6)-(8). In particular, from (6) we obtain

$$s(x) = -\sum_{j=1}^{m} \lambda_j(x) \nabla F_j(x). \tag{9}$$

So the steepest direction defined in this paper is a steepest direction for a standard scalar optimization problem, implicitly induced by weighting the given objective functions by the (nonnegative) a priori unknown KKT multipliers. As a consequence, the standard weighting factors [23], well known in multiobjective programming, do show up in our approach, albeit a posteriori and 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 KKT multipliers in the corresponding direction search program.

Existence of the KKT multipliers for the convex problem (5) implies that there is no duality gap, and so apply (9) we have:

$$\begin{aligned}
\theta(x) &= \sup_{\lambda \geq 0} \inf_{s \in \mathbb{R}^n} L((t,s), \lambda) \\
&= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} \inf_{s \in \mathbb{R}^n} \sum_{j=1}^{m} \lambda_j \left(\nabla F_j(x)^T s + \frac{1}{2} s^T s \right) \\
&= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} \inf_{s \in \mathbb{R}^n} \left(\sum_{j=1}^{m} \lambda_j(x) \nabla F_j(x)^T . s + \frac{1}{2} s^T s \right) \\
&= \sup_{\substack{\lambda \geq 0 \\ \sum \lambda_j = 1}} -\frac{1}{2} \| \sum_{j=1}^{m} \lambda_j(x) \nabla F_j(x) \|^2
\end{aligned} \tag{10}$$

Let us now study some properties of function θ and analyze its relation with $s(x)$ and stationarity of x.

Lemma 2 *Take $x \in U$, then*

$$\theta(x) = -\frac{1}{2} s(x)^T s(x) \text{ and } \|\theta(x)\| = \frac{1}{2} \|s(x)\|^2.$$

Lemma 3 *Under our general assumptions, we have:*

1. *For any $x \in U, \theta(x) \leq 0$.*
2. *The following conditions are equivalent.*
   (a) The point x is noncritical.
   (b) $\theta(x) < 0$.
   (c) $s(x) \neq 0$. In particular, $x \in U$ is Pareto stationary if and only if $\theta(x) = 0$.
3. *The function $s : U \to \mathbb{R}^n$, given by* (4)*, is bounded on compact sets and $\theta : U \to \mathbb{R}$, given by* (3)*, is continuous in U.*

Now we sketch the Steepest algorithm for multi-criteria optimization. At each step, at a non-stationary point, we minimize the maximum of all local models as in (2) to obtain the Steepest step (4), which is a descent direction. After that, the step length is determined by means of an Adaptive rule coupled with a backtracking procedure. Under suitable local

assumptions, full Steepest steps are always accepted and the generated sequence converges super-linear (or quadratically) to a local solution. Formally, the algorithm for finding a Pareto point is the following.

Adaptive Steepest Algorithm for Multi-criteria Optimization

Algorithm 1 ***Step 1****. Choose* $\kappa \in [0,1], \sigma \in [0,1], \alpha_1 \in (0,+\infty)$*, and* $x^1 \in \mathbb{R}^n$*, set* $k := 1$.

Step 2. *(Main loop)*

(a) Solve the direction search program of this problem:

$$\min_{\substack{\lambda \geq 0 \\ \sum \lambda_j^k = 1}} || \sum_{j=1}^{m} \lambda_j^k \nabla F_j(x^k)^T ||^2 \tag{11}$$

to obtain the optimal solution $\lambda^k = (\lambda_j^k)$, $s(x^k) = -\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)$ *and* $\theta(x^k) = -\frac{1}{2}||s(x^k)||^2$.

(b) If $\theta(x^k) = 0$*, then stop. Otherwise, proceed to step 2(c).*

(c) Set $x^{k+1} := P_U(x^k + \alpha_k s(x^k))$.

(d) Compute a step length:

If $\sum_{j=1}^m \lambda_j^k F_j(x^{k+1}) \leq \sum_{j=1}^m \lambda_j^k F_j(x^k) + \sigma \left\langle s(x^k), x^{k+1} - x^k \right\rangle$
Then $\alpha_{k+1} = \alpha_k$ *else set* $\alpha_{k+1} := \kappa \alpha_k$

Step 3. *Set* $k := k+1$*, and goto Step 2 .*

Remark 1 Define for $x \in U$

$$G(x) = \sum_{j=1}^{m} \lambda_j F_j(x).$$

We also have

$$G(x^k) = \sum_{j=1}^{m} \lambda_j^k F_j(x^k) \text{ and } G(x^{k+1}) = \sum_{j=1}^{m} \lambda_j^k F_j(x^{k+1}).$$

Then $s(x^k) = -\nabla G(x^k) = -\sum_{j=1}^m \lambda_j^k \nabla F_j(x^k)$. If Algorithm 1 stops at step k, then x^k is a Pareto stationary point of the problem MOP(F,U). Indeed, since $x^{k+1} = P_U\left(x^k - \alpha_k \nabla G(x^k)\right)$, applying Proposition 3-(ii), we have

$$\left\langle z - x^{k+1}, x^k - \alpha_k \nabla G(x^k) - x^{k+1} \right\rangle \leq 0 \; \forall z \in U. \tag{12}$$

If $x^{k+1} = x^k$, we get

$$\left\langle \nabla G(x^k), z - x^k \right\rangle \geq 0 \quad \forall z \in U. \tag{13}$$

If $U = \mathbb{R}^n$ **then** $s(x^k) = \nabla G(x^k) = 0$, which means x^k is a Pareto stationary point of the problem. If, in addition, F is pseudoconvex, from Proposition 2, it implies that x^k is a weakly efficient solution of MOP(F,U).

Now, suppose that the algorithms generates an infinite sequence. We will prove that this sequence converges to a Pareto solution of the problem MOP(F,U).

Theorem 1 *Assume that the sequence* $\{x^k\}$ *is generated by Algorithm 1. Then, each limit point (if any) of the sequence* $\{x^k\}$ *is a Pareto stationary point of the problem MOP(*F,U*). Moreover,*

- *if F is quasiconvex on U, then the sequence $\{x^k\}$ converges to a Pareto stationary point of the problem.*
- *if F is pseudoconvex on U, then the sequence $\{x^k\}$ converges to a weakly efficient solution of the problem.*

Proof Applying Proposition 4, we get

$$G(x^{k+1}) \leq G(x^k) + \left\langle \nabla G(x^k), x^{k+1} - x^k \right\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2. \tag{14}$$

In (12), taking $z = x^k \in U$, we arrive at

$$\left\langle \nabla G(x^k), x^{k+1} - x^k \right\rangle \leq -\frac{1}{\alpha_k}\|x^{k+1} - x^k\|^2. \tag{15}$$

Combining (14) and (15), we obtain

$$G(x^{k+1}) \leq G(x^k) - \sigma \left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle - \left(\frac{1-\sigma}{\alpha_k} - \frac{L}{2}\right)\|x^{k+1} - x^k\|^2. \tag{16}$$

We now claim that $\{\alpha_k\}$ is bounded away from zero, or in other words, the step size changes finite times. Indeed, suppose, by contrary, that $\alpha_k \to 0$. From (16), there exists $k_0 \in \mathbb{N}$ satisfying

$$G(x^{k+1}) \leq G(x^k) - \sigma \left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle \ \forall k \geq k_0.$$

According to the construction of α_k, the last inequality implies that $\alpha_k = \alpha_{k_0}$ for all $k \geq k_0$. This is a contradiction. And so, there exists $k_1 \in \mathbb{N}$ such that for all $k \geq k_1$, we have $\alpha_k = \alpha_{k_1}$ and

$$G(x^{k+1}) \leq G(x^k) - \sigma \left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle. \tag{17}$$

Noting that $\left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle \geq 0$, we infer that the sequence $\{G(x^k)\}$ is convergent and

$$\sum_{k=0}^{\infty} \left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle < \infty; \quad \sum_{k=0}^{\infty} \|x^{k+1} - x^k\|^2 < \infty. \tag{18}$$

From (12), for all $z \in U$, we have

$$\begin{aligned} \|x^{k+1} - z\|^2 &= \|x^k - z\|^2 - \|x^{k+1} - x^k\|^2 + 2\left\langle x^{k+1} - x^k, x^{k+1} - z \right\rangle \\ &\leq \|x^k - z\|^2 - \|x^{k+1} - x^k\|^2 + 2\alpha_k \left\langle \nabla G(x^k), z - x^{k+1} \right\rangle. \end{aligned} \tag{19}$$

Let $\bar{x}$ be a limit point of $\{x^k\}$. There exists a subsequence $\{x^{k_i}\} \subset \{x^k\}$ such that $\lim_{i\to\infty} x^{k_i} = \bar{x}$. In (19), let $k = k_i$ and take the limit as $i \to \infty$. Noting that $\|x^k - x^{k+1}\| \to 0$, ∇G is continuous, we get

$$\langle \nabla G(\bar{x}), z - \bar{x} \rangle \geq 0 \quad \forall z \in U.$$

If $U = \mathbb{R}^n$ **then** $s(x^k) = \nabla G(x^k) = 0$, which means $\bar{x}$ is a Pareto stationary point of the problem MOP(F,U).

Now, suppose that F is quasiconvex on U. Denote

$$U := \left\{ x \in U : F(x) \leq F(x^k) \quad \forall k \geq 0 \right\}.$$

Note that U contains the solution set of MOP(F,U), and hence, is not empty. Take $\hat{x} \in U$. Since $F(x^k) \geq F(\hat{x})$ for all $k \geq 0$, it implies that

$$\left\langle \nabla F(x^k), \hat{x} - x^k \right\rangle \leq 0, \ \forall k \geq 0. \tag{20}$$

Therefore,

$$\left\langle \nabla G(x^k), \hat{x} - x^k \right\rangle \leq 0 \quad \forall k \geq 0. \tag{21}$$

Combing (19) and (21), we get

$$\|x^{k+1} - \hat{x}\|^2 \leq \|x^k - \hat{x}\|^2 - \|x^{k+1} - x^k\|^2 + 2\alpha_k \left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle. \tag{22}$$

Applying Lemma 1 with $a_k = \|x^{k+1} - \hat{x}\|^2$, $b_k = 2\alpha_k \left\langle \nabla G(x^k), x^k - x^{k+1} \right\rangle$, we deduce that the sequence $\{\|x^k - \hat{x}\|\}$ is convergent for all $\hat{x} \in U$. Since the sequence $\{x^k\}$ is bounded, there exist a subsequence $\{x^{k_j}\} \subset \{x^k\}$ such that $\lim_{i\to\infty} x^{k_i} = \bar{x} \in U$. From (17) and (20), we know that the sequence $\left\{F(x^k)\right\}$ is nonincreasing and convergent. It implies that $\lim_{k\to\infty} F(x^k) = F(\bar{x})$ and $F(\bar{x}) \leq F(x^k)$ for all $k \geq 0$. This means $\bar{x} \in U$ and the sequence $\left\{\|x^k - \bar{x}\|\right\}$ is convergent. Thus,

$$\lim_{k\to\infty} \|x^k - \bar{x}\| = \lim_{i\to\infty} \|x^{k_i} - \bar{x}\| = 0.$$

Note that each limit point of $\left\{x^k\right\}$ is a Pareto stationary point of the problem. Then, the whole sequence $\left\{x^k\right\}$ converges to $\bar{x}$ - a Pareto stationary point of the problem.

Moreover, by Proposition 2, when F is pseudoconvex, this Pareto stationary point becomes a weakly efficient solution of MOP(F,U). □

Next, we estimate the convergence rate of Algorithm 1 in solving unconstrained optimization problems.

Corollary 1 *Assume that F is convex, $C = \mathbb{R}^m$ and $\{x^k\}$ is the sequence generated by Algorithm 1. Then,*

$$G(x^k) - G(x^*) = O\left(\frac{1}{k}\right),$$

where x^* is a solution of the problem.

Proof Let x^* be a solution of the problem. Denote $\Delta_k := G(x^k) - G(x^*)$. From (17), noting that $x^k - x^{k+1} = \alpha_k \nabla G(x^k)$, we have

$$\Delta_{k+1} \leq \Delta_k - \sigma\alpha_{k_1} \|\nabla G(x^k)\|^2 \quad \forall k \geq k_1. \tag{23}$$

On the other hand, since the sequence $\left\{x^k\right\}$ is bounded and f is convex, it holds that

$$\begin{aligned} 0 \leq \Delta_k &\leq \left\langle \nabla G(x^k), x^k - x^* \right\rangle \\ &\leq M\|\nabla G(x^k)\|, \end{aligned} \tag{24}$$

where $M := \sup\left\{\|x^k - x^*\| : k \geq k_1\right\} < \infty$. From (23) and (24), we arrive at

$$\Delta_{k+1} \leq \Delta_k - Q\Delta_k^2 \quad \forall k \geq k_1, \tag{25}$$

where $Q := \frac{\sigma\alpha_{k_1}}{M^2}$. Noting that $\Delta_{k+1} \leq \Delta_k$, from (25), we obtain

$$\frac{1}{\Delta_{k+1}} \geq \frac{1}{\Delta_k} + Q \geq \ldots \geq \frac{1}{\Delta_{k_1}} + (k - k_1)Q,$$

which implies

$$G(x^k) - G(x^*) = O\left(\frac{1}{k}\right).$$

4 Numerical experiments

Example 1 First, let's look at a simple nonconvex problem (P):

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{x_1^2 + x_2^2 + 3}{1 + 2x_1 + 8x_2} \\ &\text{subject to} \quad x \in U, \end{aligned}$$

where $C = \{x = (x_1, x_2)^\top \in \mathbb{R}^2 | g_1(x) = -x_1^2 - 2x_1x_2 \leq -4;\ x_1, x_2 \geq 0\}$. It is quite evident that for this problem, the objective function f is pseudoconvex on the convex feasible set (Example 5.2 [12], Liu et al., 2022).

Fig. 1 illustrates the temporary solutions of the proposed method for various initial solutions. It demonstrates that the outcomes converge to the optimal solution $x^* = (0.8922, 1.7957)$ of the given problem. The objective function value generated by Algorithm GDA is 0.4094, which is better than the optimum value 0.4101 of the neural network in Liu et al. (2022) [12].

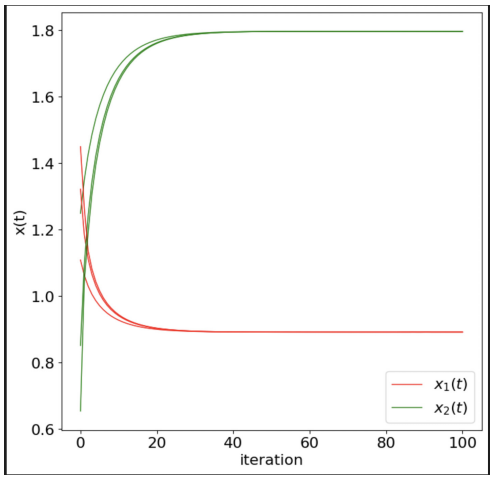

Fig. 1 Computational results for Example 1.

Example 2 Consider the following nonsmooth pseudoconvex optimization problem with nonconvex inequality constraints (Example 5.1 [12], Liu et al., 2022):

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{e^{|x_2 - 3|} - 30}{x_1^2 + x_3^2 + 2x_4^2 + 4} \\ &\text{subject to} \quad g_1(x) = (x_1 + x_3)^3 + 2x_4^2 \leq 10, \\ &\qquad\qquad\ g_2(x) = (x_2 - 1)^2 \leq 1, \\ &\qquad\qquad\ 2x_1 + 4x_2 + x_3 = -1, \end{aligned}$$

where $x = (x_1, x_2, x_3, x_4)^\top \in \mathbb{R}^4$. The objective function $f(x)$ is nonsmooth pseudoconvex on the feasible region $\mathscr{X}$, and the inequality constraint g_1 is continuous and quasiconvex on $\mathscr{X}$, but not pseudoconvex (Example 5.1 [12]). Fig 2 shows that Algorithm GDA converges to an optimal solution $x^* = (-1.0649, 0.4160, -0.5343, 0.0002)^\top$ with the optimal value -3.0908, which is better than the optimal value -3.0849 of the neural network model in [12].

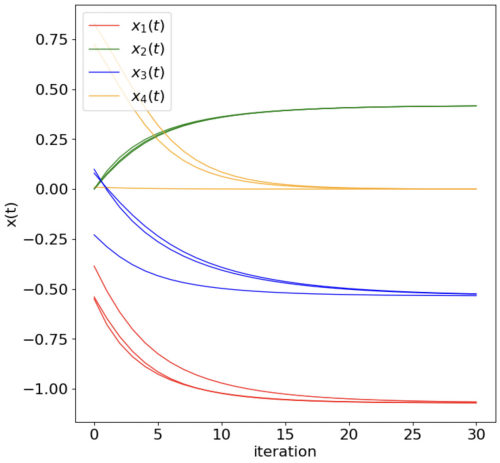

Fig. 2 Computational results for Example 2.

Example 3 Consider the nonsmooth, nonconvex optimization problem:

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{e^{|x_1-1|} + |x_1 - x_2| + x_2^2 - 30}{(x_1+x_2)^2 + 1} \\ &\text{subject to} \quad x \in U, \end{aligned}$$

where

$$C = \left\{x \in \mathbb{R}^2 : x_1 + x_1^2 + e^{x_1+x_2} \leq 4, \|x\|^2 \leq 1, x_1 + 2x_2 = 1\right\}.$$

Since it has been demonstrated that the objective function $f(x)$ is pseudoconvex in Liu et al.(2012), network(3) may be utilized to solve this constrained nonsmooth and pseudoconvex optimization problem.

Example 4 Let $e := (1,\ldots,n) \in \mathbb{R}^n$ be a vector, $\alpha > 0$ and $\beta > 0$ be constants satisfying the parameter condition $2\alpha > 3\beta^{3/2}\sqrt{n}$. Consider Problem (P) (Example 4.5 [7], Ferreira et. al, 2022) with the associated function

$$f(x) := a^T x + \alpha x^T x + \frac{\beta}{\sqrt{1+\beta x^T x}} e^T x,$$

with $a \in \mathbb{R}^n_{++}$ is convex and the nonconvex constraint is given by $C := \left\{x \in \mathbb{R}^n_{++} : 1 \leq x_1 \ldots x_n\right\}$. This example is implemented to compare Algorithm GDA with the original gradient descent algorithm (GD). We choose a random number $\beta = 0.741271, \alpha = 3\beta^{3/2}\sqrt{n+1}$ fulfilled the parameter condition and Lipschitz coefficient $L = \left(4\beta^{3/2}\sqrt{n} + 3\alpha\right)$ suggested in [7]. The step size of Algorithm GD is $\lambda = 1/L$ and the initial step size of Algorithm GDA is $\lambda_0 = 5/L$. Table 2 shows the optimal value, number of loops, computational time of two algorithms through the different dimensions. From this result, Algorithm GDA is more efficient than GD at the computational time with the same optimal output value, specially for the large scale dimensions.

Example 5 Consider the following large-scale pseudoconvex optimization problem (Bian et al., 2018): minimize $f(x) = -\exp\left(-\sum_{i=1}^{600} \frac{x_i^2}{e_i^2}\right)$ subject to $Ax = b, g(x) \leq 0$, where $x \in$

n	Algorithm GDA			Algorithm GD		
	$f(x^*)$	loop	time	$f(x^*)$	loop	time
10	79.3264	9	0.9576	79.3264	15	1.5463
20	220.5622	10	6.0961	220.5622	67	34.0349
50	857.1166	12	2.8783	857.1166	16	4.6824
100	2392.5706	12	17.2367	2392.5706	17	30.8886
150	4903.1452	85	2543.1163	4805.9582	500	12421.8229
200	7065.9134	65	525.1199	7179.3542	200	1610.6560
500	26877.7067	75	2273.0011	27145.6292	500	14113.5003

Table 1 Computational results for Example 3.

$\mathbb{R}^{600}, \rho = (\rho_1, \rho_2, \ldots, \rho_{600})^\top$ with $\rho_i > 0, A \in \mathbb{R}^{1\times 600}$ with the first half entries of A are 1 , the rest are 3 , and $b = 16$.

The inequality constraints are

$$g_i(x) = x^2_{10\cdot(i-1)+1} + x^2_{10\cdot(i-1)+2} + \cdots + x^2_{10\cdot(i-1)+10} - 20$$
$$i = 1, 2, \cdots, 60$$

Note that the objective function f in (33) is locally Lipschitz and pseudoconvex on $\mathbb{R}^{600}$. Fig. 5 depicts the convergent state $x(t)$ of neural network (7) in solving problem (33). Fig. 6 depicts the continuous descending value of $-\ln(-f(x(t)))$ resulted from neural network (7) until reaching the minimum 650.8807.

n	Algorithm GDA			Algorithm GD		
	$f(x^*)$	loop	time	$f(x^*)$	loop	time
10	79.3264	9	0.9576	79.3264	15	1.5463
20	220.5622	10	6.0961	220.5622	67	34.0349
50	857.1166	12	2.8783	857.1166	16	4.6824
100	2392.5706	12	17.2367	2392.5706	17	30.8886
150	4903.1452	85	2543.1163	4805.9582	500	12421.8229
200	7065.9134	65	525.1199	7179.3542	200	1610.6560
500	26877.7067	75	2273.0011	27145.6292	500	14113.5003

Table 2 Computational results for Example 3.

5 Applications to machine learning

5.1 Supervised feature selection

We consider the classification problem with m classes, n samples, and p features. Given a training set $\{(x_i, y_i) \mid i = 1, \ldots, n\}$, where $x_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ is a p-dimensional feature vector of the ith sample and $y_i \in \{1, \ldots, m\}$ represents the corresponding labels indicating classes or target values. Let $y = (y_1, \ldots, y_n)^T$ be a vector of class labels for the n samples. The matrix dataset $X = \left(x_{(1)}, \ldots, x_{(p)}\right)$ of size $n \times p$, where $x_{(j)} = (x_{1j}, \ldots, x_{nj})^{\mathrm{T}}$ is the j th column vector and also the jth predictor. Centered features can be obtained by $F_j = C_n x_{(j)}$, where $C_n = I_n - ee^T/n$ is the centering matrix (Wang, Liu, Nie and Huang, 2015), I_n is the $n \times n$ identity matrix, and e is a vector of 1's. $\mathscr{F} = \left\{F_1, \ldots, F_p\right\}$ is a full set of features of the matrix dataset X, where F_j indicates the jth feature in $\mathscr{F}(j = 1, \ldots, p)$.

Feature selection aims to select an optimal subset of $k(k < \mathrm{p})$ features $\{F_1, \ldots, F_k\} \subseteq \mathscr{F}$ with the least correlation with each other and the highest relevancy to the target class y. The smaller the number k, the more likely important features are to be selected by a better feature selection method.

To obtain high-quality features, feature redundancy minimiza- tion and relevancy maximization need to be carried out together. Based on the positive-semidefinite redundancy matrix Q and the feature relevance measures in Eqs. (7)-(8), we propose a holistic feature selection problem formulation as follows:

$$\begin{array}{ll} \text{minimize} & \frac{w^T Q w}{\rho^T w} \\ \text{subject to} & e^T w = 1 \\ & w \geq 0 \end{array}$$

where $w = (w_1, \ldots, w_p)^T$ is the feature score vector to be determined, Q is the similarity coefficient matrix in (14), $\rho = (\rho_1, \ldots, \rho_p)^T$ is a feature relevancy vector (e.g., $\rho_{IG}(7)$ or $\rho_{FS}(8)$), and $e = (1, \ldots, 1)^T$ is a vector of ones.

The quadratic $w^T Q w$ in the numerator of the objective function in (15) measures global and supervised redundancy, while $\rho^T w$ in the denominator quantifies feature relevancy. It is known that the objective function $f(w)$ in problem (15) is pseudoconvex in view of the convexity of the numerator and concavity of the denominator (Cambini, Crouzeix, & Martein, 2002; Liu et al., 2012).

5.2 Multi-variable logistic regression

The experiments are performed with the dataset including $\mathbf{N}$ observations $(\mathbf{a}_i, b_i) \in \mathbb{R}^d \times \mathbb{R}, i = 1, \ldots, n$. The cross-entropy loss function for multi-variable logistic regression is given by $J(x) = -\sum_{i=1}^{\mathbf{N}} \left(b_i \log\left(\sigma\left(-x^T \mathbf{a}_i\right)\right) + (1 - b_i) \log\left(1 - \sigma\left(-x^T \mathbf{a}_i\right)\right)\right)$, where σ is the sigmoid function. Associated with ℓ_2-regularization, we get the regularized loss function $\bar{J}(x) = J(x) + \frac{1}{2\mathbf{N}}\|x\|^2$. The Lipschitz coefficient L is estimated by $\frac{1}{2\mathbf{N}}(\|A\|^2/2 + 1)$, where $A = \left(a_1^\top, \ldots, a_n^\top\right)^\top$. We compare the algorithms for training the logistic regression problem by using datasets Mushrooms and W8a [13]. The computational results are shown in Fig. 3 and Fig. 5 respectively. The figures suggest that Algorithm GDA outperforms Algorithm GD in terms of objective function values and step sizes during iterations.

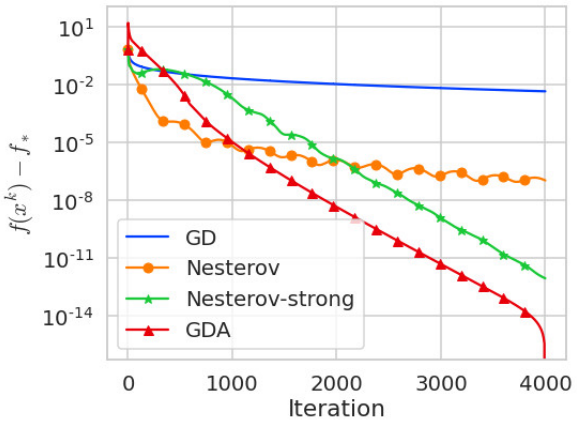

Fig. 3 The computational results for logistic regression with dataset Mushrooms.

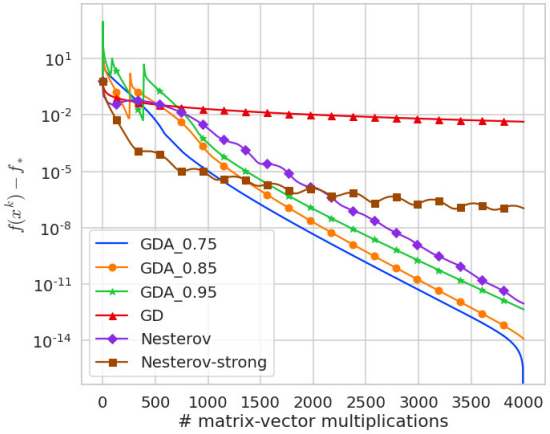

Fig. 4 The computational results for logistic regression with dataset W8a.

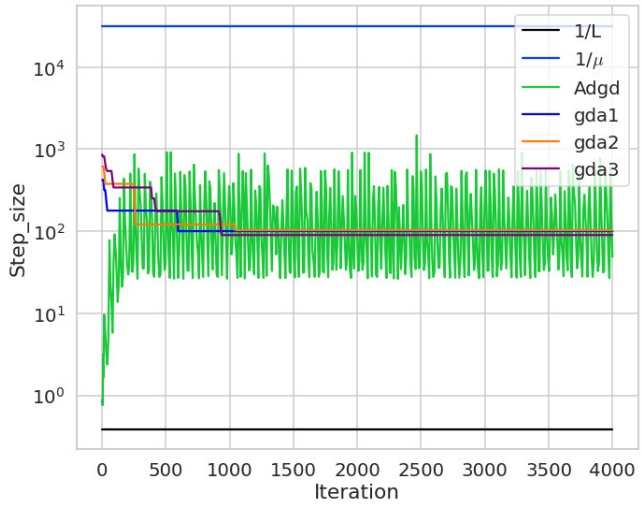

Fig. 5 The computational results for logistic regression with dataset W8a.

5.3 Neural networks for classification

In order to provide an example of how the proposed algorithm can be implemented into a neural network training model, we will use the standard ResNet-18 architectures that have been implemented in PyTorch and will train them to classify images taken from the Cifar10 dataset while taking into account the cross-entropy loss. In the studies with ResNet-18, we made use of Adam's default settings for its parameters. The computational outcomes shown in Fig. 7 reveal that Algorithm GDA outperforms Algorithm GD in terms of testing accuracy and train loss over iterations.

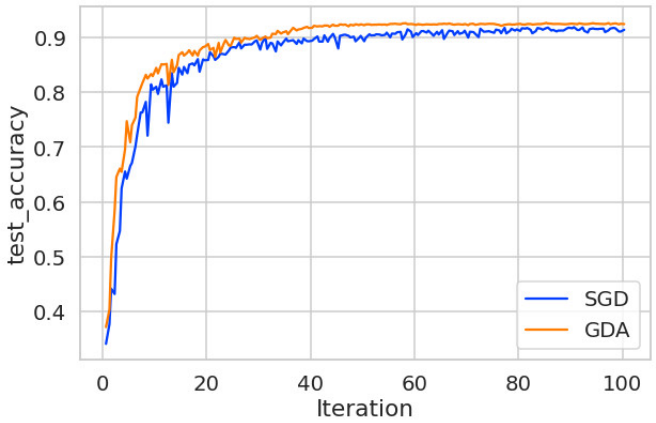

Fig. 6 The training outcomes for ResNet-18 model.

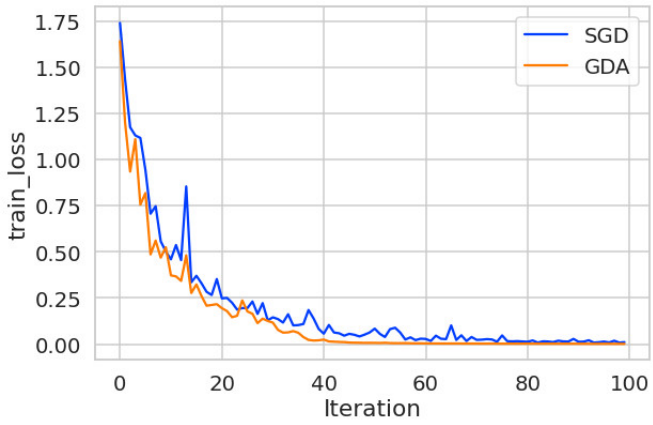

Fig. 7 The training outcomes for ResNet-18 model.

6 Conclusion

We proposed a novel easy adaptive step-size process in a wide family of solution methods for optimization problems with non-convex objective functions. This approach does not need any line-searching or prior knowledge, but rather takes into consideration the iteration sequence's behavior. As a result, as compared to descending line-search approaches, it significantly reduces the implementation cost of each iteration. We demonstrated technique convergence under simple assumptions. We demonstrated that this new process produces a generic foundation for optimization methods. The preliminary results of computer experiments demonstrated the new procedure's efficacy.

References

1. Bauschke, H.H., Combettes, P.L.: Convex analysis and monotone operator theory in hilbert spaces. Springer, New York (2011)
2. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2009)
3. Rockafellar, R.T.: Convex analysis. Princeton University Press, Princeton, New Jersey (1970)
4. W. Bian, L. Ma, S. Qin, X. Xue: Neural network for nonsmooth pseudoconvex optimization with general convex constraints, Neural Networks 101, 1-14 (2018).
5. Cevher, V., Becker, S., Schmidt, M.: Convex optimization for big data. Signal Process. Magaz. 31, 32–43 (2014)
6. J.E. Dennis, R.B. Schnabel: Numerical methods for unconstrained optimization and nonlinear equations, Prentice-Hall, New Jersey, 1983.
7. O. P. Ferreira, W. S. Sosa, On the Frank–Wolfe algorithm for non-compact constrained optimization problems, Optimization, 71:1, 197-211 (2022).
8. Y. Hu , J. Li, C. K. Yu, Convergence Rates of Subgradient Methods for Quasiconvex Optimization Problems, Computational Optimization and Applications, 77, 183–212 (2020).
9. K. C. Kiwiel, Convergence and efficiency of subgradient methods for quasiconvex minimization, Math. Program., Ser. A 90: 1–25 (2001).
10. I. V. Konnov, Simplified versions of the conditional gradient method, Optimization, 67(12), 2275-2290 (2018).
11. Guanghui Lan, First-order and Stochastic Optimization Methods for Machine Learning, Springer Series in the Data Sciences, Springer Nature Switzerland (2020)
12. N. Liu, J. Wang, S. Qin: A one-layer recurrent neural network for nonsmooth pseudoconvex optimization with quasiconvex inequality and affine equality constraints, Neural Networks 147, 1-9 (2022).

13. Yura Malitsky, Konstantin Mishchenko, Adaptive Gradient Descent without Descent, Proceedings of Machine Learning Research, 119:6702-6712 (2020).
14. O. Mangasarian, Pseudo-convex functions, Siam Control, 8, 281-290 (1965)
15. Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.
16. Xu, H.K.: Iterative algorithms for nonlinear operators. J. London Math. Soc. 66, 240-256 (2002)
17. C.K. Yu, Y. Hu, X. Yang & S. K. Choy, Abstract convergence theorem for quasi-convex optimization problems with applications, Optimization, 68(7), 1289-1304, 2019.
18. Yura Malitsky, Konstantin Mishchenko, Adaptive Gradient Descent without Descent, Proceedings of Machine Learning Research, 119:6702-6712 (2020).
19. O. Mangasarian, Pseudo-convex functions, Siam Control, 8, 281-290 (1965)
20. Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.
21. Xu, H.K.: Iterative algorithms for nonlinear operators. J. London Math. Soc. 66, 240-256 (2002)
22. C.K. Yu, Y. Hu, X. Yang & S. K. Choy, Abstract convergence theorem for quasi-convex optimization problems with applications, Optimization, 68(7), 1289-1304, 2019.