ISSN: 2664-9640

Generalized Conditional Gradient Method for Multi-objective Optimization with Applications

Jun Yuan*, Yangdong Xu

School of Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

* Corresponding Author

Abstract

This paper is devoted to investigating a multi-objective Generalized Conditional Gradient Method for multi-objective composite optimization problems. The proposed algorithm utilizes multiple-step size strategies and is evaluated through numerical experiments. Moreover, the effectiveness of the method is demonstrated in multi-task learning scenarios by testing it on the Multi-MNIST, Multi-Fashion, and Multi-Fashion-MNIST datasets. The results demonstrate that the method performs superbly and may enhance the model's generalizability in multi-task learning.

Keywords

Multi-objective optimization, Generalized conditional gradient method, Multi-task learning.

1. Introduction

Multi-objective optimization (MOO) aims to simultaneously minimize (or maximize) several objective functions while taking into certain constraints. There will frequently be conflicts between different objective functions. There is no solitary solution that concurrently maximizes some of the objectives while they are in conflict. Finding Pareto optimal solutions (also known as efficient points) in these circumstances is the purpose of MOO. Multi-objective optimization has found successful applications in a variety of fields, including engineering, logistics, transportation, and medicine.

Recent years have seen a large number of approaches to multi-objective optimization problems presented. The scalarization strategy [1-2] is one of the most often used techniques. It converts the original multi-objective problem into a scalar-valued objective function. However, choosing the appropriate parameters (or weights) in advance can be challenging. On the other hand, for some non-convex situations, scalarization approaches may produce unanticipated mistakes. [3] is an illustration of the inability of the scalarization approach to address issues.

Heuristic algorithms are another prominent method [4-5]. Heuristic algorithms are a category of stochastic optimization techniques that replicate the process of natural development. These algorithms can handle most multi-objective optimization problems even problems with black-box objective functions. However, they can cause high computational effort when expensive functions are considered since they are based on the search strategy of sampling by changing or recombining former points. Therefore, if there are some very large-scale problems, it is almost impossible to solve them using heuristic algorithms. Furthermore, such algorithms do not include any intrinsic measures of distance to convergence, such as a step length and descending direction, and therefore there is no clear stopping criterion.

Many scholars pay much attention to extending single-objective optimization based on descent methods to multi-objective optimization in recent years [6-10]. An ordering of the importance of the components of the objective function vector need not be provided in these algorithms.

The common descent direction (i.e. this descending direction can decrease all objective functions) is solved in the iteration of the algorithm. In most instances, it is possible to prove convergence to a first-order Pareto stationary point.

In this paper, we consider the problem

$$\begin{aligned} \min \quad & F(x)+G(x) \\ \text{s.t.} \quad & x \in C, \end{aligned} \tag{1}$$

Where $F:\mathbb{R}^n \to \mathbb{R}^m$ is continuous differentiable function with $F=(F_1,F_2,\cdots,F_m)$, $G:\mathbb{R}^n \to \mathbb{R}^m$ is proper close convex function with $G=(G_1,G_2,\cdots,G_m)$, and $C \subseteq \mathbb{R}^n$ is compact convex set.

We present a generalized conditional gradient method for multi-objective optimization problems to solve (1). The suggested approach simply requires the solution of a common descent direction through the subproblem in each iteration, followed by the solution of an acceptable step size for the original problem.

This paper is structured as follows. Section 2 introduces the required conditions for solving multi-objective optimization problems and proposes a generalized conditional gradient method for multi-objective optimization problems. Section 3 reports the results of numerical experiments conducted for four test problems and multi-task learning. Finally, Section 4 presents a summary and discusses future research directions.

2. Algorithm Framework

In problem (1), we define the Jacobian JF of function $F \coloneqq (F_1,\cdots,F_\mathrm{m})$ that has full rank and Lipschitz continuous, if there exist constants $L_1,L_2,\cdots,L_m>0$ such that

$$\|F_i(x)-F_i(y)\| \le L_i\|x-y\|, \forall x,y \in C \text{ and } \forall i=1,\cdots m.$$

Here, we set $L \coloneqq \max\{L_i: i=1,\cdots,m\}$.

Let $H(x)=F(x)+G(x)$, then the problem (1) can be rewritten as follows:

$$\begin{aligned} \min \quad & H(x) \\ \text{s.t} \quad & x \in C. \end{aligned} \tag{2}$$

The search direction of generalized conditional gradient method at a given $x \in C$ is defined as

$$d(x)=q(x)-x,$$

where $q(x)$ is an optimal solution of subproblem

$$\min_{v\in C}\max_{i=1,\cdots,m}\{\nabla F_i(x)^\top(v-x)+G_i(v)-G_i(x)\}, \tag{3}$$

and we define that

$$q(x) \in \arg\min_{v\in C}\max_{i=1,\cdots,m}\{\nabla F_i(x)^\top(v-x)+G_i(v)-G_i(x)\}, \tag{4}$$

Since (3) is convex function and C is compact convex set, (4) has an optimal solution and, as a result, $q(x)$ is well defined. Now, we convert (3) into the following problem

$$\begin{aligned} \min_{v,a} \quad & a \\ \text{s.t.} \quad & \nabla F_i(x)^\top(v-x)+G_i(v)-G_i(x) \le a, i=1,\cdots,m \\ & v \in C. \end{aligned} \tag{5}$$

Let $\theta_x(q)$ be the optimal value of (3) given by

$$\theta_x(q) \coloneqq \max_{i=1,\cdots,m}\{\nabla F_i(x)^\top(q(x)-x)+G_i(q(x))-G_i(x)\} \tag{6}$$

Subsequently, we introduce the generalized conditional gradient method for multi-objective optimization (GCGMMO) algorithm for addressing problem (1). The algorithm is presented in the following framework:

```
Generalized Conditional Gradient Method for Multi-objective Optimization (GCGMMO)
Input: initial point $x_0 \in \mathbb{R}^n$, iteration precision $\varepsilon$.
Output: $x^k, H(x^k)$.
for $k = 0,1,\cdots,$ do
    Compute an optimal solution $q(x^k)$ and the optimal value $\theta_{x^k}(q)$ as:
        $q(x^k) \in \arg\min_{v \in C} \max_{i=1,\cdots,m}\{\nabla F_i^\top (v - x^k) + G_i(v) - G_i(x^k)\}$;
        $\theta_{x^k}(q) := \max_{i=1,\cdots,m}\{\nabla F_i^\top (q(x^k) - x^k) + G_i\left(q(x^k)\right) - G_i(x^k)\}$.
    Let $d^k = q(x^k) - x^k$.
    $\left|\theta_{x^k}(q)\right| \leq \varepsilon$ then
        return $x^k, H(x^k)$.
    end if
        Compute the step size $t_k$ by line search and update:
            $x^{k+1} = x^k + t_k d^k$.
end for
```

We assume that $\theta_{x^k}(q) \leq 0$ for all $k = 0,1,\cdots$, i.e., Algorithm GCGMMO generates an infinite sequence $\{x^k\}$. Owing to $x^k \in C, q(x^k) \in C$ and $t_k \in (0,1]$ for all $k = 0,1,\ldots, C$ is a compact convex set, $x^{k+1} = x^k + t_k(q(x^k) - x^k) \in C$. it can be concluded that $\{x^k\} \subseteq C$ In the case of single-objective optimization, we can determine the exact optimal step size. Yet, it is extremely difficult in the case of multi-objective optimization. As a result, we investigate the convergence qualities of the sequence created by Algorithm GCGMMO using five distinct, well-defined, and realistic step sizes listed below.

Armijo stepsize: Let $t_{k_0} \in (0,1]$, $\beta \in (0,1)$, $\lambda \in (0,1)$, for $j = 0,1,2,\cdots,$ if

$$H\left(x^k + t_{k_j} d^k\right) \leq H(x^k) + \beta t_{k_j} \theta_{x^k}(q) e,$$

return $t_k = t_{k_j}$, else return that

$$t_{k_{j+1}} = \lambda t_{k_j}.$$

Adaptive stepsize: Define the stepsize as

$$t_k := \min\left\{1, \frac{\theta_{x^k}(q)}{L} \|d^k\|^2\right\} = \arg\min_{t \in (0,1]}\left\{\theta_{x^k}(q)t + \frac{L}{2}\|d^k\|^2 t^2\right\}.$$

Diminishing stepsize: Defined the diminishing stepsize as

$$t_k = \frac{2}{k+2}.$$

Max-type stepsize: Let $t_{k_0} \in (0,1]$, $\beta \in (0,1)$, $\lambda \in (0,1)$. Choose a nonnegative integer M, let $m(0) = 0$ and $0 \leq m(k) \leq \min\{m(k-1)+1, M\}$, then we apply the Armijo step size rule to find a $t_k > 0$ that satisfieins

$$H(x^k + t_k d^k) \leqq c_k + \beta t_k \theta_{x^k}(q) e,$$

where $c_k = \max_{0 \leq l \leq m(k)} H(x^{k-l})$.

Average-type stepsize: The average-type step size criterion is similar to the max-type step size criterion except that let $p_0 = 1$, $\eta \in [0,1]$ and update c_k and p_k as follows:

$$p_{k+1} = \eta p_k + 1,$$

$$c_{k+1} = \frac{\eta p_k c_k + H(x^{k+1})}{p_{k+1}}$$

Subsequently, we propose a Selection-type algorithm that deals specifically with sequences of the list, in which the selection process is derived from the genetic algorithm [4]. Let $I \subseteq \{1,\cdots,m\}$, then the descent direction of the algorithm is as follows:

$$q(x) \in \arg\min_{v\in C} \max_{i\in I}\{\nabla F_i(x)^\top (v-x) + G_i(v) - G_i(x)\}.$$

The main algorithm framework is as follows:

```
Selection-type GCGMMO
Input: initial points $L_0 \in \mathbb{R}^n$, popsize $N$, iteration precision $\varepsilon$.
for $k = 0,1,\cdots,$ do
    Set $L_{temp} = L_k$;
    for each $x^k$ in the list $L_k$ do
        for $I \in 2^{\{1,\dots,m\}}$ do
            Compute $q(x) \in \arg\min_{v\in C} \max_{i\in I}\{\nabla F_i(x)^\top (v-x) + G_i(v) - G_i(x)\}$;
            Compute the step size $t_k$ by Armijo line search and update:
            $x^{k+1} = x^k + t_k(q(x^k) - x^k)$
            Add $x^{k+1}$ into $L_{temp}$;
        end for
    end for
    Fast-non-dominated-sort $L_{temp}$ and choose best $N$ points;
    Set $L_{k+1} = L_{temp}$;
    if each $|\theta_{x^{k+1}}(q)| \le \varepsilon$ in the list $L_{k+1}$
        return $L_{k+1}$;
    end if
end for
```

Each x^k in L_k will generate $2^m - 1$ descent directions, which will make the population size larger after the Armijo line search. Then, the iterative points are ranked using fast non-dominated sort in L_{k+1}, and the top N points with the best performance are selected to enter the next iteration. Then the quality of the solution generated by the algorithm may be better through the non-dominated sorting and selection process.

3. Numerical Experiment

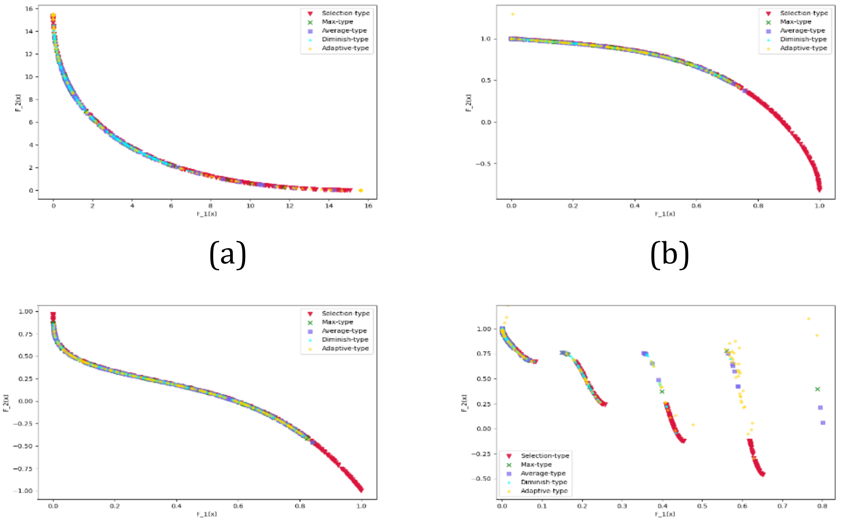
3.1. Compare with different stepsizes

In this subsection, the proposed methods are compared to each other with some test problems. Note that Table 1 presents key information such as the source, variable dimensions, number of objectives, and geometry types of the Pareto front for each of the four selected problems. The approximate Pareto fronts of test problems are shown in Figure 1.

Table 1: List of test problems

Problem	Source	n	m	Geometry
Lov1	[11]	2	2	Convex
JOS2	[12]	30	2	Concave
JOS4	[12]	30	2	Mixed
ZDT3	[13]	30	2	Disconnected

Some results can be obtained from the approximate Pareto front of the above four test questions: The non-dominant points obtained by the Selection-type method are the most uniform and can be spread throughout the approximate Pareto front. The approximate Pareto front distribution of other methods is worse than that of the selection method. However, some points of the adaptive method are not convergent. Therefore, adding a non-dominated sorting process into the algorithm can improve the generation of the Pareto front.



(a) (b)

(c) (d)

Figure 1: Approximate Pareto front: (a) Lov1; (b) JOS2; (c) JOS4; (d) ZDT3

3.2. Multitasking learning

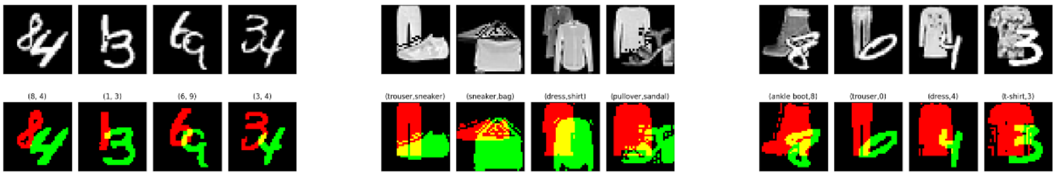
This subsection applies the suggested method to Multi-Tasking Learning (MTL). An MTL problem is made up of m or related tasks as a loss vector:

$$min_{\theta^{sh},\theta^1,\dots,\theta^m} L(\theta^{sh}, \theta^1, \dots, \theta^m) = \left(L_1(\theta^{sh}, \theta^1), L_2(\theta^{sh}, \theta^2), \dots, L_m(\theta^{sh}, \theta^m)\right),$$

where $L_i\left(\theta^{sh}, \theta^i\right)$ is the loss of i-th task. An MTL algorithm optimizes all tasks at the same time by using shared structure and information. The gradient of the model may then be updated in the decreasing direction of solving the subproblems.

We created Multi-MNIST datasets from the [14] to examine the effectiveness of our method on Multi-Task Learning situations with different task relations. We select two pictures with distinct digits at random from the original MNIST dataset [15] and then combine them into a new image by placing one digit in the top-left corner and the other in the bottom-right corner. Each digit may be moved by four pixels in each direction. We may create a Multi-Fashion-MINST (we call it Multi-Fashion) dataset with overlap Fashion-MNIST items [16], as well as a Multi-Fashion + MNIST (we call it Multi-Fashion-MNIST) dataset with overlap MNIST and Fashion-MNIST items, using the same method. We have a two-objective MTL problem for each dataset: classify the item on the top-left (task 1) and classify the item on the bottom-right (task 2). We construct a LeNet [15] based MTL neural network that is comparable to the one used in [17]. Before training, we set the maximum epoch to 100, and then record the final training accuracy and test accuracy. It can be seen from Table 2 that the training accuracy of the first single-task training baseline in the Multi-Fashion dataset is lower than that of the conditional gradient method, while others are higher than that of the conditional gradient method. In the test set, the accuracy of all single-task training baselines is lower than that of the proposed method. This is because the suggested method incorporates regular terms into the model, improving the model's generalization ability and successfully alleviating the overfitting problem. And the proposed algorithm performs well while solving multi-objective optimization problems if the objective functions are convex and non-differentiable. As a result, while fitting the data set, we recommend using regular terms in the model to increase test accuracy.

Table 2 displays the acquired results.



(a) (b) (c)

Figure 2: Dataset:(a) Multi-MNIST; (b) Multi-Fashion; (c) Multi-Fashion-MNIST

Before training, we set the maximum epoch to 100, and then record the final training accuracy and test accuracy. It can be seen from Table 2 that the training accuracy of the first single-task training baseline in the Multi-Fashion dataset is lower than that of the conditional gradient method, while others are higher than that of the conditional gradient method. In the test set, the accuracy of all single-task training baselines is lower than that of the proposed method. This is because the suggested method incorporates regular terms into the model, improving the model's generalization ability and successfully alleviating the overfitting problem. And the proposed algorithm performs well while solving multi-objective optimization problems if the objective functions are convex and non-differentiable. As a result, while fitting the data set, we recommend using regular terms in the model to increase test accuracy.

Table 2: Train result of datasets

Dataset	task	Train-base	Train-acc	Test-base	Test-acc
MNIST	Task1	0.995	0.988	0.973	0.983
	Task2	0.996	0.983	0.974	0.978
Fashion	Task1	0.852	0.874	0.828	0.852
	Task2	0.912	0.888	0.860	0.868
Fashion-MNIST	Task1	0.886	0.875	0.853	0.856
	Task2	0.995	0.978	0.969	0.975

4. Conclusion

This paper aims to study a generalized conditional gradient method for multi-objective optimization designed for multi-objective composite optimization problems. On this basis, Selection-type GCGMMO is constructed to generate a better quality Pareto front. Additionally, The proposed step size criteria are compared with the selection-type method. The results show that the Pareto front generated by the selection-type method is more uniform and complete. In addition, the proposed algorithm is applied to multi-task learning. The experimental results demonstrate the feasibility of the proposed algorithm. Further work is to investigate versions of the proposed algorithm for solving problems involving non-convex functions.

References

[1] I. Das, J.E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems, Structural Optimization, Vol. 14 (1997) No. 1, p.63-69.

[2] G. Eichfelder. Scalarizations for adaptively solving multi-objective optimization problems, Computational Optimization and Applications, Vol. 44 (2009), p.249-273.

[3] J. Fliege, L.G Drummond, B.F. Svaiter. Newton's method for multiobjective optimization, SIAM Journal on Optimization, Vol. 20 (2009) No. 2, p. 602-626.

[4] K. Deb, A. Pratap, S. Agarwal, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on evolutionary computation, Vol. 6 (2002) No. 2, p.182-197.

[5] L. Ke, Q. Zhang, R. Battiti. MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antColony, IEEE Transactions on Cybernetics, Vol. 43 (2013) No. 6, p.1845-1859.

[6] P.B. Assunção, P.F. Orizon, L.F. Prudente. Conditional gradient method for multiobjective optimization, Computational Optimization and Applications, Vol. 78 (2021), p.741-768.

[7] J. Fliege, B.F. Svaiter. Steepest descent methods for multicriteria optimization, Mathematical methods of operations research, Vol. 51 (2000), p.479-494.

[8] H. Tanabe, E.H. Fukuda, N. Yamashita. Proximal gradient methods for multiobjective optimization and their applications, Computational Optimization and Applications, Vol. 72 (2019), p.339-361.

[9] H. Tanabe, E.H. Fukuda, N. Yamashita. Convergence rates analysis of a multiobjective proximal gradient method, Optimization Letters, Vol. 17 (2023) No. 2, p.333-350.

[10] G. Cocchi, M. Lapucci. An augmented Lagrangian algorithm for multi-objective optimization, Computational Optimization and Applications, Vol. 77 (2020) No. 1, p.29-56.

[11] S. Huband, P. Hingston, L. Barone, et al. A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation, Vol. 10 (2006) No. 5, p.477-506.

[12] Y. Jin, M. Olhofer, B. Sendhoff. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? Proceedings of the genetic and evolutionary computation conference (San Francisco, USA, July 7-11, 2001), Vol. 1, p.1042-1049.

[13] E. Zitzler, K. Deb, L. Thiele. Comparison of multiobjective evolution algorithms: empirical results, Evolutionary Computation, Vol. 8 (2000) No. 2, p.173-195.

[14] S. Sara, F. Nicholas, E.H. Geoffrey. Dynamic routing between capsules. In Advances in Neural Information Processing Systems (Long Beach, CA, USA, December 4-9, 2017), p.3856–3866.

[15] L. Yann, B. Léon, B. Yoshua, et al. Gradient-based learning applied to document recognition, Proceedings of the IEEE, Vol. 86 (1998) No. 11, p.2278–2324.

[16] H. Xiao, K. Rasul, R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint, 2017, doi: 10.48550/arXiv.1708.07747.

[17] S. Ozan, K. Vladlen. Multi-task learning as multi-objective optimization. In Advances in Neural Information Processing Systems (Montréal, Canada, December 3-8, 2018), p.525–536.