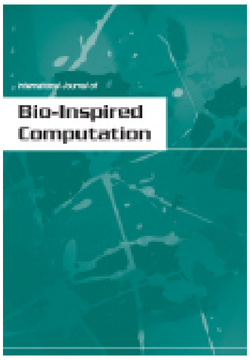


International Journal of Bio-Inspired Computation

ISSN online: 1758-0374 - ISSN print: 1758-0366
https://www.inderscience.com/ijbic

Decomposition gradient descent method for bi-objective optimisation

Jingjing Chen, Genghui Li, Xi Lin

DOI: 10.1504/IJBIC.2023.10056897

Article History:

Received:	17 November 2022
Last revised:	25 December 2022
Accepted:	06 January 2023
Published online:	22 January 2024

Decomposition gradient descent method for bi-objective optimisation

Jingjing Chen*

Department of Computer Science,
City University of Hong Kong,
Hong Kong, China
Email: jingjche2-c@my.cityu.edu.hk
*Corresponding author

Genghui Li

School of System Design and Intelligent Manufacturing,
Southern University of Science and Technology,
Shenzhen, 518055, China
Email: genghuili2-c@my.cityu.edu.hk

Xi Lin

Department of Computer Science,
City University of Hong Kong,
Hong Kong, China
Email: Xi.Lin@my.cityu.edu.hk

Abstract: Population-based decomposition methods decompose a multi-objective optimisation problem (MOP) into a set of single-objective subproblems (SOPs) and then solve them collaboratively to produce a set of Pareto optimal solutions. Most of these methods use heuristics such as genetic algorithms as their search engines. As a result, these methods are not very efficient. This paper investigates how to do a gradient search in multi-objective decomposition methods. We use the NBI-style Tchebycheff method to decompose a MOP since it is not sensitive to the scales of objectives. However, since the objectives of the resultant SOPs are non-differentiable, they cannot be directly optimised by the classical gradient methods. We propose a new gradient descent method, decomposition gradient descent (DGD), to optimise them. We study its convergence property and conduct numerical experiments to show its efficiency.

Keywords: multi-objective optimisation; decomposition strategy; NBI-style Tchebycheff method; gradient descent method.

Reference to this paper should be made as follows: Chen, J., Li, G. and Lin, X. (2024) 'Decomposition gradient descent method for bi-objective optimisation', *Int. J. Bio-Inspired Computation,* Vol. 23, No. 1, pp.28–38.

Biographical notes: Jingjing Chen received her MSc in Applied Mathematics from the Xidian University, Xi'an, China, in 2019. She is currently pursuing her PhD with the Department of Computer Science, City University of Hong Kong, Hong Kong. Her research interests include optimisation algorithms and machine learning.

Genghui Li received his MSc in Computer Science and Technology from the Shenzhen University, Shenzhen, China, in 2016, and PhD in Computer Science from the City University of Hong Kong, Hong Kong, China, in 2021. He is currently a Post-doctoral Fellow with the Southern University of Science and Technology, Shenzhen, China. His research interests include evolutionary computation, computational intelligence, and machine learning.

Xi Lin received his BS in Mathematics from the South China University of Technology, Guangzhou, China, in 2013, MS in Statistics from the Columbia University, New York, NY, USA, in 2015, and PhD in the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2021. He is currently a Post-doctoral Research Fellow with the Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include multi-objective optimisation, Bayesian optimisation, and machine learning.

1 Introduction

A multi-objective optimisation problem (MOP) can be formulated as follows:

$$\begin{aligned} &\text{minimise } F(x) = (f_1(x), ..., f_m(x))^T \\ &\text{subject to } x \in \Omega, \end{aligned} \tag{1}$$

where $\Omega \subset R^n$ is the decision space, $x \in \Omega$ is the decision vector, and $f_i(x) : \Omega \to R,\ i = 1, 2, ..., m$ are continuously differentiable objective functions. Since these objectives often conflict with one another, no solution in Ω can minimise all the objectives simultaneously. *Pareto optimality* is commonly used to define the best trade-off solutions. A decision vector $x^* \in \Omega$ is called *weakly Pareto optimal* if there does not exist another decision vector $x \in \Omega$ such that $f_i(x) < f_i(x^*)$ for all $i \in \{1, ..., m\}$. A decision vector $x^* \in \Omega$ is called *Pareto optimal* if no $x \in \Omega$ such that $f_i(x) \le f_i(x^*)$ for all $i \in \{1, ..., m\}$, and $f_j(x) < f_j(x^*)$ for at least one index $j \in \{1, ..., m\}$. The set of all Pareto optimal decision vectors is called the *Pareto set* (PS). An objective vector is Pareto optimal if its corresponding decision vector is Pareto optimal and the set of all Pareto optimal objective vectors is called the *Pareto front* (PF) (Miettinen, 2012; Ojha et al., 2019).

Population-based evolutionary algorithms (Deb et al., 2002; Zhou et al., 2011; Coello et al., 2020; Jiang and Yang, 2015) are efficient in finding multiple solutions simultaneously, and decomposition-based multi-objective evolutionary algorithms (MOEA/D) (Zhang and Li, 2007; Trivedi et al., 2016; Wu et al., 2018; Li et al., 2021) are widely used to find a set of Pareto optimal solutions to approximate the PS and/or PF. MOEA/D and its variants convert the MOP into a number of single-objective optimisation subproblems (SOPs) (Li and Zhang, 2008; Zhou and Zhang, 2015; Chen et al., 2021), and then solve them simultaneously in a collaborative manner. Any single-objective optimisation algorithm can be used to solve each subproblem. Most existing MOEA/D algorithms use derivative-free heuristics as their search engines, which could be inefficient for problems with valid gradient information.

Recently, gradient-based multi-objective optimisation methods have attracted growing research efforts in machine learning (Sener and Koltun, 2018; Milojkovic et al., 2019; Mahapatra and Rajan, 2020). For example, these gradient methods have been used in multi-objective reinforcement learning (Xu et al., 2020), recommendation systems (Milojkovic et al., 2019; Mitrevski et al., 2020) and multi-task learning (MTL) (Martín and Schütze, 2018; Lin et al., 2019). The multiple-gradient descent algorithm (MGDA) (Fliege and Svaiter, 2000; Gebken et al., 2017) is one of the most widely used multi-objective gradient descent methods in machine learning. It uses the Karush-Kuhn-Tucker (KKT) condition and finds a descent direction for all the objectives. Sener and Koltun (2018) cast an MTL problem as a MOP and then use MGDA to solve it. Lin et al. (2019) decomposes an MTL problem into several multi-objective optimisation subproblems and uses MGDA as its optimiser. A significant drawback of MGDA is that it cannot produce a Pareto optimal solution for a given preference. It cannot generate a set of uniformly distributed solutions for approximating the PF.

In this paper, we propose a novel gradient-based method to directly optimise the subproblems in MOEA/D such that it can generate Pareto optimal solutions for any given preferences. Specifically, due to the drawbacks of classical scalarisation methods, we use the NBI-style Tchebycheff method proposed by Zhang et al. (2010) as the scalarisation method, which is insensitive to the scales of objectives. Since the corresponding SOP is non-differentiable, scalar optimisation methods cannot be directly adopted. Hence, we propose a new gradient descent method, the decomposition gradient descent (DGD) method, to optimise these SOPs. The DGD method combines the gradient descent method and the improved MGDA to solve the non-differentiable subproblem. In order to get an adequate objective value decrease, the line search based on the Armijo rule is employed. Under some mild assumptions, there is a decreasing sequence for each NBI-style Tchebycheff problem. The numerical experiments show that the DGD method can obtain a well-distributed Pareto optimal set.

Our major contributions to this paper are as follows:

- Firstly, we give the theoretical results of the NBI-style Tchebycheff method.
- Secondly, we propose a gradient-based method for the NBI-style Tchebycheff problem and give some theoretical analysis.
- Thirdly, the proposed DGD method performs well in both convergence and diversity of the obtained PS than traditional gradient-based algorithms.

The rest of this paper is organised as follows. Section 2 introduces the decomposition-based and gradient-based methods, including the MGDA and the increment central descent method (ICDM). Section 3 presents the NBI-style Tchebycheff problem and some theoretical results. Section 4 proposes a DGD method to solve the NBI-style Tchebycheff problem. Section 5 gives the theoretical analysis of the DGD method. In Section 6, we conduct numerical experiments. Finally, Section 7 concludes this paper.

2 Related works

2.1 Decomposition-based approaches

The scalarisation method plays a crucial role in MOEA/D (Zhang and Li, 2007) and its variants (Liu et al., 2013; Lin et al., 2019). It transforms the MOP into a sequence of single-objective optimisation subproblems. An evenly distributed PF can be approximately obtained by solving these subproblems. Classical scalarisation methods include the weighted sum (WS) method (Zadeh, 1963), the weighted Tchebycheff (TCH) method (Bowman, 1976), the penalty-based boundary intersection (PBI) method (Zhang and Li, 2007), and the normal boundary intersection (NBI) method (Das and Dennis, 1998), etc.

However, most of the existing decomposition methods (Eichfelder, 2009) have shortcomings in approximating the PF of the MOP. The WS method fails to obtain the non-convex PF (Boyd et al., 2004). Moreover, the WS and the weighted TCH methods are sensitive to the scales of the objectives. The PBI method is sensitive to the penalty coefficient. Compared to these three approaches mentioned above, the NBI approach is relatively insensitive to scales of objective functions and has no extra parameters. However, the NBI method can not be applied directly since the extra constraints. Since most scalarisation methods rely heavily on extra parameters (e.g., weighting factors, penalty coefficient) or constraints, Zhang et al. (2010) proposed an NBI-style Tchebycheff method, which could get rid of these drawbacks. The following section will introduce the NBI-style Tchebycheff method and present some theoretical results.

2.2 Gradient-based methods

2.2.1 Multi-objective gradient descent method

In MGDA (Fliege and Svaiter, 2000; Désidéri, 2012), a necessary condition for a point $x \in \Omega \subset R^n$ to be locally Pareto optimal is

$$\text{Im}(JF(x)) \cap (-R^m_{++}) = \varnothing, \tag{2}$$

where $JF(x)$ is the $m \times n$ Jacobian matrix of F at x and $\text{Im}(JF(x)) = \{JF(x)d | d \in R^n\}$. A point x that satisfies equation (2) is called Pareto critical. This means that there does not exist a direction d at the Pareto critical point x, such that $\nabla f_i(x)^T d < 0$, for all $i \in \{1, ..., m\}$. In other words, if a point x is not Pareto critical, then there exists a direction $d \in R^n$ at x, such that $\nabla f_i(x)^T d < 0$, for all $i \in \{1, ..., m\}$. The descent direction d can be obtained by solving the following optimisation problem:

$$\begin{aligned} &\min \ \beta + \frac{1}{2} \|d\|_2^2 \\ &\text{s.t. } \nabla f_i(x)^T d \le \beta \quad \forall i \in \{1, ..., m\}. \end{aligned} \tag{3}$$

Since the optimisation problem (3) is a convex quadratic problem with linear inequality constraints, it always has a unique optimal solution d. Namely, there exists ω_1, ω_2, ..., $\omega_m \ge 0$, such that $d = -\sum_{i=1}^m \omega_i \nabla f_i(x)$, with $\sum_{i=1}^m \omega_i = 1$. It is the convex combination of negative gradient descent direction of all objectives. If $d = \mathbf{0}$, the solution x satisfies the KKT condition and is called Pareto stationary point. It is the necessary condition for Pareto optimality.

Based on the descent direction d, the Armijo rule is employed to compute the step length t. Let $\sigma \in (0, 1)$, the condition to accept t is

$$F(x + td) \le F(x) + \sigma t JF(x)d. \tag{4}$$

Starting from an initial point $x^0 \in \Omega$, the MGDA generates a sequence $\{x^k\}$ with $F(x^{k+1}) < F(x^k)$, and each accumulation point of the sequence $\{x^k\}$ is Pareto critical (Vieira et al., 2012; Gebken et al., 2017). However, these iterative methods can not guarantee the diversity of Pareto optimal solutions since MGDA can only converge to one arbitrary Pareto optima. Moreover, they cannot obtain the pre-specified Pareto optimal solution.

2.2.2 The increment central descent method

The increment central descent method (ICDM) (Oliveira and Takahashi, 2022) was established under the assumption that objective functions have L-Lipschitz continuous gradients, that is, there exists $L > 0$, such that $\|\nabla f_i(x) - \nabla f_i(y)\| \le \|x - y\|, \ \forall x, y \in \Omega, \ i = 1, 2, ..., m$. They obtain the multi-objective descent direction by solving the following problem:

$$\begin{aligned} &\min \ \frac{1}{2} \|d\|_2^2 \\ &\text{s.t. } \nabla f_i(x)^T d \le -\|\nabla f_i(x)\| \quad \forall i \in \{1, ..., m\}. \end{aligned} \tag{5}$$

ICDM achieves a faster convergence rate than MGDA since the calculation of the gradient information in ICDM is less accurate than that of MGDA. Specifically, the constraints of the quadratic programs (5) are more relaxed than (3). The descent direction solution set obtained by ICDM is larger than that of MGDA. Though the gradient descent direction obtained by ICDM is not as accurate as MGDA, it has a faster convergence speed than MGDA.

Many other classical scalar optimisation methods have been generalised to solve MOPs, such as the Newton method (Fliege et al., 2009; Wang et al., 2019), the trust-region method (Thomann and Eichfelder, 2019), and the SQP method (Fliege and Vaz, 2016; Ansary and Panda, 2021) and so on. These methods can not only solve the unconstrained MOP but have also been applied to MOP with constraints and real-world applications (Lin et al., 2022). On the other hand, gradient information or the Hessian matrix can also integrate with evolutionary algorithms (Bosman, 2011; Hernández et al., 2018; Wang et al., 2022; Nedjah and Mourelle, 2015) to boost the performance of classical MOEAs in solving the real-valued MOPs.

3 NBI-style Tchebycheff method

In this work, we consider the bicriteria case ($m = 2$) of the NBI-style Tchebycheff problem as an illustration. Let $F : \Omega \to R^m$ be a continuously differentiable mapping, $F^1 = (f_1(x_1^*), \ f_2(x_1^*))$ and $F^2 = (f_1(x_2^*), \ f_2(x_2^*))$ be the two extreme points of the MOP. Let the line segment linking F^1 and F^2 be the convex hull of individual minima (CHIM):

$$H = \{\alpha F^1 + (1 - \alpha) F^2 | 0 \le \alpha \le 1\}.$$

The NBI-style Tchebycheff method is used to solve the following problem:

$$\min_{x\in\Omega} g^{tn}(x|r,\lambda)$$
$$= \max\{\lambda_1(f_1(x)-r_1), \lambda_2(f_2(x)-r_2)\},$$

where $r=(r_1,r_2)^T \in H$. $\lambda=(\lambda_1,\lambda_2)^T$ is the normal vector to the CHIM. This method takes advantage of both the NBI and the weighted TCH method, which is relatively insensitive to scales of objectives and performs well in obtaining a well-distributed PF of the bi-objective optimisation problem.

In order to generalise the NBI-style Tchebycheff method into m objectives, we simplify the NBI-style Tchebycheff problem by removing the weight vector. Let z^{nad} be the nadir objective vector, which consists of the respective global maximum: $z_i^{nad} = \max_{x\in\Omega} f_i(x), i = 1,...,m$. Let $\mathbf{1} = (1,...,1)^T$ be the all-one vector, and define a hyperplane H:

$$H = \{r \in R^m | \mathbf{1}^T r = -b\},$$

where $b = \max_{i=1,\dots,m} z_i^{nad}$. Then the NBI-style Tchebycheff method aims to solve the problem given as follows:

$$\min_{x\in\Omega} g^{tn}(x|r) = \max\{f_1(x)-r_1,...,f_m(x)-r_m\} \quad (6)$$

where $r=(r_1,...,r_m)^T \in H$. Compared to the WS and weighted TCH methods, the NBI-style Tchebycheff method is independent of the scales of objectives since there are no weighting factors in equation (6).

We present some theoretical analyses concerning the NBI-style Tchebycheff problem in the following.

Theorem 1: The solution of the NBI-style Tchebycheff problem (6) is weakly Pareto optimal.

Theorem 2: The NBI-style Tchebycheff problem (6) has at least one Pareto optimal solution.

The proofs of Theorems 1 and 2 can refer to Miettinen (2012).

Theorem 3: Let x^* be Pareto optimal, define a hyperplane $H = \{r \in R^m | \mathbf{1}^T r = -b\}$. Then there exists a reference point $r \in H$ such that x^* is a solution of the NBI-style Tchebycheff problem (6).

Proof: Let x^* be Pareto optimal. Let us assume that there does not exist a reference point r such that x^* is a solution to the NBI-style Tchebycheff problem. Let $\hat{t} := \frac{\mathbf{1}^T F(x^*)+b}{\mathbf{1}^T\mathbf{1}}$ and $r = F(x^*) - \mathbf{1}\hat{t}$. Then we have $r \in H$ and $r_i = f_i(x^*) - \hat{t}, i = 1,...,m$.

If x^* is not the minimal solution of the NBI-style Tchebycheff problem, then there exists another point $x' \in \Omega$ that is a solution of the NBI-style TCH problem, meaning that

$$\max_{i=1,\dots,m}\{f_i(x')-r_i\} < \max_{i=1,\dots,m}\{f_i(x^*)-r_i\}$$
$$= \max_{i=1,\dots,m}\{f_i(x^*)-(f_i(x^*)-\hat{t})\} = \hat{t}.$$

Thus $f_i(x') - r_i < \hat{t}$ for all $i = 1,...,m$. This means that

$$f_i(x') - r_i = f_i(x') - (f_i(x^*) - \hat{t}) < \hat{t},$$

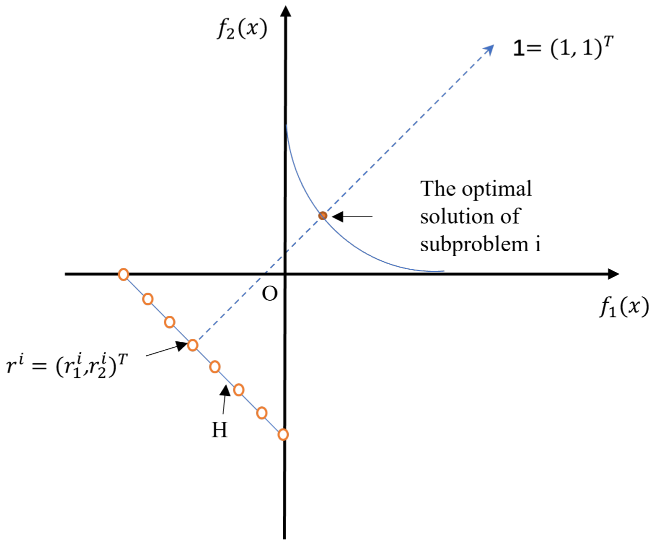
and after simplifying the expression, we have

$$f_i(x') < f_i(x^*).$$

for all $i = 1,...,m$. Here we have a contradiction with the Pareto optimality of x^*, which completes the proof.

The theoretical results show that it is sufficient to obtain all Pareto optimal solutions by varying reference points on the hyperplane H. Figure 1 illustrates the NBI-style Tchebycheff problem with two objectives.

Figure 1 Illustration of the NBI-style Tchebycheff problem (see online version for colours)



4 The DGD method for bi-objective optimisation

This article assumes that $f_i(x)$ $(i = 1,2)$ are continuously differential and supposes that all the gradients of objectives are bounded, i.e., there exists a constant $G > 0$ such that $\|\nabla f_i(x)\| \le G_i$, for any $x \in \Omega$, $i = 1, 2$. Let $G = \max_{1\le i\le 2} G_i^2$. For simplicity's sake, all r values in equation (6) are assumed to be zero. Then the NBI-style problem of the bi-objective optimisation problem has the following form:

$$\min_{x\in\Omega} g^{tn}(x|r) = \max\{f_1(x), f_2(x)\} \quad (7)$$

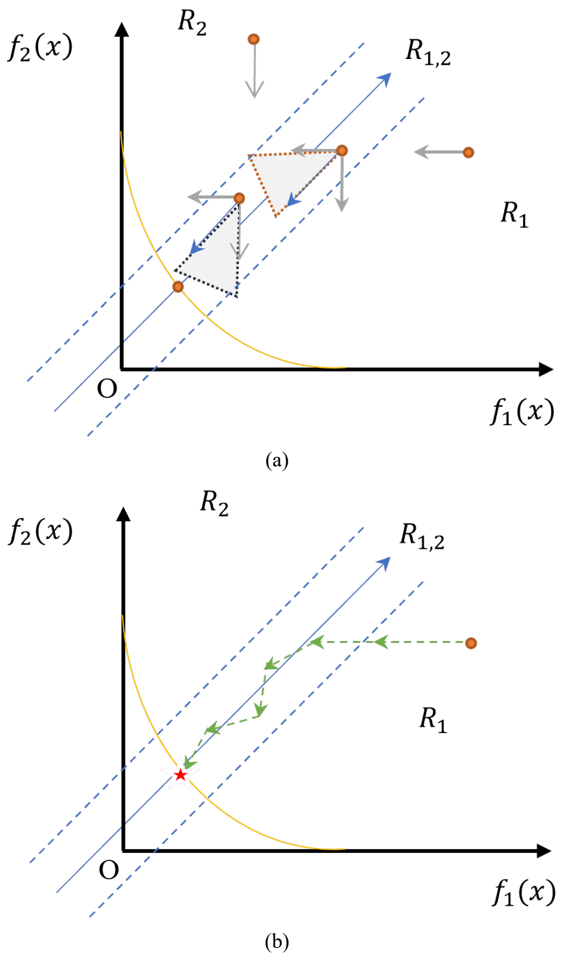
Since the NBI-style TCH problem (7) is non-differentiable, applying the gradient descent method to solve it directly is infeasible. In order to utilise the gradient information to solve the NBI-style TCH problem, we classify the variables in the decision space into two categories:

$$\Omega_I = \{x | f_I(x) - f_i(x) > \delta, i = \{1,2\} \setminus I\},$$
$$\Omega_{\bar{I}} = \{x || f_1(x) - f_2(x)| \le \delta\}, \quad (8)$$

where $I := \arg\max_{1\le i\le 2} f_i(x) \in \{1,2\}$. I implies that the corresponding function value $f_I(x)$ is the maximum between $f_1(x)$ and $f_2(x)$. δ is a predefined hyperparameter to divide the objective space better, which is related to G.

Under this decomposition mechanism, if $x \in \Omega_I \subset \Omega$, $f_I(x)$ is relatively greater than other terms in the NBI-style Tchebycheff problem. While the point locates in the region $\Omega_{\bar{I}}$, the elements in the NBI-style Tchebycheff problem are approximately equal to each other. To better understand this partition, we use the NBI-style TCH problem with two objective functions as an example. The objective space can be divided into three regions R_1, R_2 and $R_{1,2}$. These regions imply that the first term in the NBI-style Tchebycheff problem (7) is relatively greater, less than, or approximately equal to the second one, respectively. See Figure 2 for an illustration.

Figure 2 The division to the objective space, (a) descent direction in different regions (b) a descent sequence (see online version for colours)



4.1 Computing the descent direction

If the point x locates in region $\Omega_{\bar{I}}$, the difference between $f_1(x)$ and $f_2(x)$ in the NBI-style Tchebycheff problem is no greater than δ. It is infeasible to apply the scalar steepest descent method to calculate the descent direction in this area. Because of the conflicting property among objective functions, the decrease of one objective function may cause the increase of the other. Therefore, minimising one objective function at each iteration is inefficient since it may cause vibration among the terms in equation (7) that are approximately equal. Under this circumstance, MGDA is directly applied to minimise two objectives in the NBI-style Tchebycheff problem, that is,

$$\min_{x \in \Omega} \{f_1(x), f_2(x)\}. \tag{9}$$

As mentioned in Section 2, the gradient descent direction of one differential MOP can be obtained by solving the optimisation problem (3). The descent direction is the convex combination of all negative gradients. In order to restrict the succeeding point not exceeding the region $\Omega_{\bar{I}}$, we narrow the range of the possible descent direction d. There are m kinds of possibilities of the direction d once the point drops in the region $\Omega_{\bar{I}}$. As the shadow triangles in Figure 2(a) show, the descent direction d needs to ensure the decrease of two objectives simultaneously in the region $R_{1,2}$, and is required to restrict the succeeding point not exceeding the region. Therefore, the gradient descent direction in the region $\Omega_{\bar{I}}$ can be obtained by solving the following optimisation problem:

$$\begin{aligned} &\min \ \nabla f_s(x)^T d + \frac{1}{2}\|d\|_2^2 \\ &\text{s.t. } \nabla f_i(x)^T d \le \nabla f_s(x)^T d, \quad \forall i \in \{1,2\} \setminus \{s\}, \end{aligned} \tag{10}$$

where $s = \arg\min_{1 \le i \le 2} f_i(x)$. Compared to equation (3), this quadratic programming with linear inequality constraints restricted the range of the gradient descent direction. The linear inequality constraints in equation (10) restrict the range of the descent direction. These inequalities imply that once a point falls into the region $\Omega_{\bar{I}}$, according to the gradient update rule, the subsequent iteration process will be restricted in this area. In the next section, we will give proof that the succeeding points will not exceed the area $\Omega_{\bar{I}}$ under some mild assumptions.

The Lagrange function of the constrained optimisation problem (10) is given below:

$$\begin{aligned} L(d,\lambda) &= \nabla f_s(x)^T d + \frac{1}{2}\|d\|_2^2 \\ &\quad + \sum_{i \ne s} \lambda_i (\nabla f_i(x) - \nabla f_s(x))^T d. \end{aligned} \tag{11}$$

$\lambda_i \ge 0$ are Lagrangian multipliers. And the KKT conditions satisfied by the vector d and λ are

$$\begin{aligned} &\frac{\partial L(d,\lambda)}{\partial d} = d + \sum_{i \ne s} \lambda_i \nabla f_i(x) \\ &+ \left(1 - \sum_{i \ne s} \lambda_i\right) \nabla f_s(x) = 0, \\ &\frac{\partial L(d,\lambda)}{\partial \lambda_i} = (\nabla f_i(x) - \nabla f_s(x))^T d = 0, \text{ for } i = 1,2. \end{aligned}$$

Let $\omega_s = 1 - \lambda_s, \omega_i = \lambda_i, i \in \{1,2\} \setminus \{s\}$. Then we get the gradient descent direction of equation (10):

$$d = -\sum_{i=1}^{2} \omega_i \nabla f_i(x), \sum_{i=1}^{2} \omega_i = 1, \omega_i \geq 0, i = 1, 2. \quad (12)$$

On the other hand, if one point x lies in the region Ω_I, the greatest objective $f_I(x)$ will be optimised with the gradient descent method. In other words, the NBI-style Tchebycheff problem (7) is reformulated as:

$$\min_{x \in \Omega} \; f_I(x). \quad (13)$$

Hence, as Figure 2(a) shows, the gradient descent direction at current point x is

$$d = -\nabla f_I(x). \quad (14)$$

4.2 Computing the stepsize

After getting the descent direction d, we compute the step length t. Let $\sigma \in (0, \frac{1}{2})$ be a prespecified constant. The improved Armijo rule to accept t is

$$\begin{aligned} & f_I(x + td) \leq f_I(x) + \sigma t \nabla f_I(x)^T d, \\ & f_i(x + td) \leq f_I(x + td) + \sigma t \left\| \nabla f_I(x) \right\|^2, \\ & I = \arg\max_{1 \leq i \leq 2} f_i(x), i \in \{1, 2\} \setminus \{I\}. \end{aligned} \quad (15)$$

The second inequality in the improved Armijo rule (15) implies that a point will not move directly from Ω_I to another region Ω_i, where $i = \{1, 2\} \setminus \{I\}$. In other words, once one point drops in one region Ω_I, it will still move within this area or move to the region $\Omega_{\bar{I}}$. If $\{I\} = \{1, 2\}$, it means that the point drops into the region $\Omega_{\bar{I}}$. Then the improved Armijo rule degenerates into the Armijo rule.

The proposed algorithm is described in Algorithm 1.

Algorithm 1 The DGD method

for $l = 1$ to N **do**
  Set $k := 0$, choose $x^0 \in \Omega$ randomly.
  while not terminate **do**
    Calculate $\delta_k = |f_1(x^k) - f_2(x^k)|$,
    if $\delta_k \leq \delta$ **then**
      calculate the descent direction d^k as equation (10),
    else
      calculate the descent direction d^k as equation (14).
    end if
    (Line search) Compute the step size t_k. Choose t_k that satisfies
    $t_k := \max\{t : f_I(x^k + td^k) \leq f_I(x^k)$
    $+\sigma t \nabla f_I(x^k)^T d^k,$
    $f_i(x^k + td^k) \leq f_I(x^k + td^k)$
    $+\sigma t \left\| \nabla f_I(x^k) \right\|^2,$
    $I = \arg\max_{1 \leq i \leq 2} f_i(x^k),$
    $i \in \{1, 2\} \setminus \{I\}\}, t \in (0, 0.5].$
    $x^{k+1} := x^k + t_k d^k,$
    $\Delta = |g_l^{tn}(x^{k+1}|r^l) - g_l^{tn}(x^k|r^l)|$, set $k := k + 1$.
  end while
end for

5 Theoretical analysis

The following lemma indicates that once one point falls into region $\Omega_{\bar{I}}$, the following points will remain in this region.

Lemma 5.1: For the bi-objective problem, if $x^k \in \Omega_{\bar{I}}$, then $x^{k+1} \in \Omega_{\bar{I}}$.

Proof: Suppose x^k falls into the region $\Omega_{\bar{I}}$, then we have

$$-\delta < \delta_k := f_1(x^k) - f_2(x^k) < \delta.$$

Firstly, suppose $f_1(x^k) < f_2(x^k)$, that is $-\delta < \delta_k < 0$. Under this assumption, we can obtain the gradient descent direction d^k, which satisfies $\nabla f_2(x)^T d - \nabla f_1(x)^T d \leq 0$. According to the differentiability of the objective functions, we have $f_i(x^{k+1}) = f_i(x^k) + t_k \nabla f_i(x^k)^T d^k + o(\left\| t_k d^k \right\|^2)$, i = 1, 2, and we get

$$\begin{aligned} \delta_{k+1} &:= f_1(x^{k+1}) - f_2(x^{k+1}) \\ &= \delta_k + t_k(\nabla f_1(x^k)^T d^k - \nabla f_2(x^k)^T d^k) \\ &+ R(\left\| t_k d^k \right\|^2) \geq \delta_k > -\delta \end{aligned}$$

On the other hand, the gradient descent direction d^k at x^k can be obtained by solving (10). The general expression of the gradient descent direction at point x^k is $d^k = -(\omega \nabla f_1(x^k) + (1 - \omega) \nabla f_2(x^k))$. According to Cauchy-Schwarz inequality $| < u, v > | \leq \|u\| * \|v\|$, we have

$$\begin{aligned} \delta_{k+1} &= f_1(x^{k+1}) - f_2(x^{k+1}) \\ &= \delta_k + t_k[\nabla f_1(x^k)^T d^k - \nabla f_2(x^k)^T d^k] \\ &+ R(\left\| t_k d^k \right\|^2) \\ &\leq -t_k(\sigma \nabla f_1(x^k) - \nabla f_2(x^k))^T (\omega \nabla f_1(x^k) \\ &+ (1 - \omega) \nabla f_2(x^k)) - o(\left\| t_k d^k \right\|^2) \\ &\leq t_k \Bigg[-\sigma\omega \left\| \nabla f_1(x^k) \right\|^2 + (1 - \omega) \left\| \nabla f_2(x^k) \right\|^2 \\ &+ (\omega + \sigma(1 - \omega)) \frac{\left\| \nabla f_1(x^k) \right\|^2 + \left\| \nabla f_2(x^k) \right\|^2}{2} \Bigg] \\ &- o(\left\| t_k d^k \right\|^2) < \delta. \end{aligned}$$

Therefore, we have proved that $-\delta < \delta_{k+1} < \delta$. For the other circumstance $f_2(x^k) < f_1(x^k)$, we have similar proof. Hence, we complete the proof that all subsequent points will not exceed this area once the point moves to region $\Omega_{\bar{I}}$. Therefore, we have completed the proof that once the point moves to the region $\Omega_{\bar{I}}$, all subsequent points will not exceed this area.

In the following theorem, we will show the component of each subproblem‘s iteration sequence $\{x^k\}$.

Theorem 4: For each subproblem (7), the sequence $\{x^k\}$ generated by Algorithm 1 has the following form:

$$\{\underbrace{x^1, x^2, ..., x^p}_{\Omega_I}, \underbrace{x^{p+1}, x^{p+2}, ..., x^{p+q}}_{\Omega_{\bar{I}}}\}, p, q \geq 0, \quad (16)$$

Table 1 Test problems

Instance	*Objective functions*	*Variables*
F1	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = 1 + \frac{9}{n-1} * (\sum_{i=2}^{n}(x_i - x_1)^2)$ PF: $f_2(x) = 1 - \sqrt{f_1(x)}$	$[0, 1]^{30}$
F2	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + \frac{9}{n-1} * (\sum_{i=2}^{n}(x_i - x_1)^2)$ PF: $f_2(x) = 1 - f_1(x)^2$	$[0, 1]^{30}$
F5	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = 1 + \frac{9}{n-1} * (\sum_{i=2}^{n}(x_i^2 - x_1)^2)$ PF: $f_2(x) = 1 - \sqrt{f_1(x)}$	$[0, 1]^{30}$
F6	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + \frac{9}{n-1} * (\sum_{i=2}^{n}(x_i^2 - x_1)^2)$ PF: $f_2(x) = 1 - f_1(x)^2$	$[0, 1]^{30}$
RF1	$f_1(x) = x_1 + \frac{2}{\lvert J_1 \rvert} \sum_{i \in J_1} (x_i - x_1 + \frac{i}{n})^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert} \sum_{i \in J_2} (x_i - x_1 + \frac{i}{n})^2$ where $J_1 = \{j \lvert j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j \lvert j$ is even and $2 \le j \le n\}$ PF: $\{(x_1, 1 - \sqrt{x_1}) \in R^2 \lvert 0 \le x_1 \le 1\}$	$[0, 1] \times [-1, 1]^{29}$
RF2	$f_1(x) = x_1 + \frac{2}{\lvert J_1 \rvert} \sum_{i \in J_1} (x_i - x_1^2)^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert} \sum_{i \in J_2} (x_i - x_1^2)^2$ where $J_1 = \{j \lvert j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j \lvert j$ is even and $2 \le j \le n\}$ PF: $\{(x_1, 1 - \sqrt{x_1}) \in R^2 \lvert 0 \le x_1 \le 1\}$	$[0, 1]^{30}$

p, q represent the number of iterations in region Ω_I and $\Omega_{\bar{I}}$, respectively.

Proof: On the one hand, from the assumption and the second inequality in the improved Armijo rule (15), we know that points located in the region Ω_I will not move directly to another region apart from $\Omega_{\bar{I}}$. On the other hand, from Lemma 5.1, we know that once points lie in the region $\Omega_{\bar{I}}$, the following points will stay in this area. Therefore, the sequence x^k generated by the DGD method is composed of at most two kinds of variables.

In other words, the obtained sequence $\{x^k\}$ consists of $p + q$ components. These points can be classified into at most two groups: the first p terms and the last q terms are located in the region Ω_I and $\Omega_{\bar{I}}$, respectively.

Given the Lemma 5.1 and the sequences (16) mentioned above, we conclude that a proper parameter δ can ensure the sequence $\{F(x^k)\}$ varies from region R_I to the region $R_{\bar{I}}$, as the Figure 2(b) shows.

Based on Lemma 5.1 and Theorem 4, we have the following theorem.

Theorem 5: There exists a monotone decreasing sequence $\{g^{tn}(x^k|r)\}$.

Proof: Based on Lemma 5.1 and Theorem 4, we know that the generated sequence $\{x^k\}$ can only be the form of equation (16). Therefore, there are three possibilities of the sequence $\{x^k\}$:

Case 1 If $p = 0$, $q \ge 0$, the iteration sequence is $\{x^1, x^2, ..., x^q\}$. It means that all the objective functions decrease simultaneously in the region $\Omega_{\bar{I}}$, that is $f_i(x^{k+1}) < f_i(x^k)$, $i = 1, ..., m$. Then we have

$$\begin{aligned} g^{tn}(x^{k+1}|r) &= \max\{f_1(x^{k+1}), f_2(x^{k+1})\} \\ &< \max\{f_1(x^k), f_2(x^k)\} \\ &= g^{tn}(x^k|r), \end{aligned}$$

Thus $\{g^{tn}(x^k|r)\}$ is a decreasing sequence.

Case 2 If $p > 0$, $q = 0$, the iteration sequence is $\{x^1, x^2, ..., x^p\}$. For $x^k, x^{k+1} \in \{x^1, x^2, ..., x^p\}$, we have

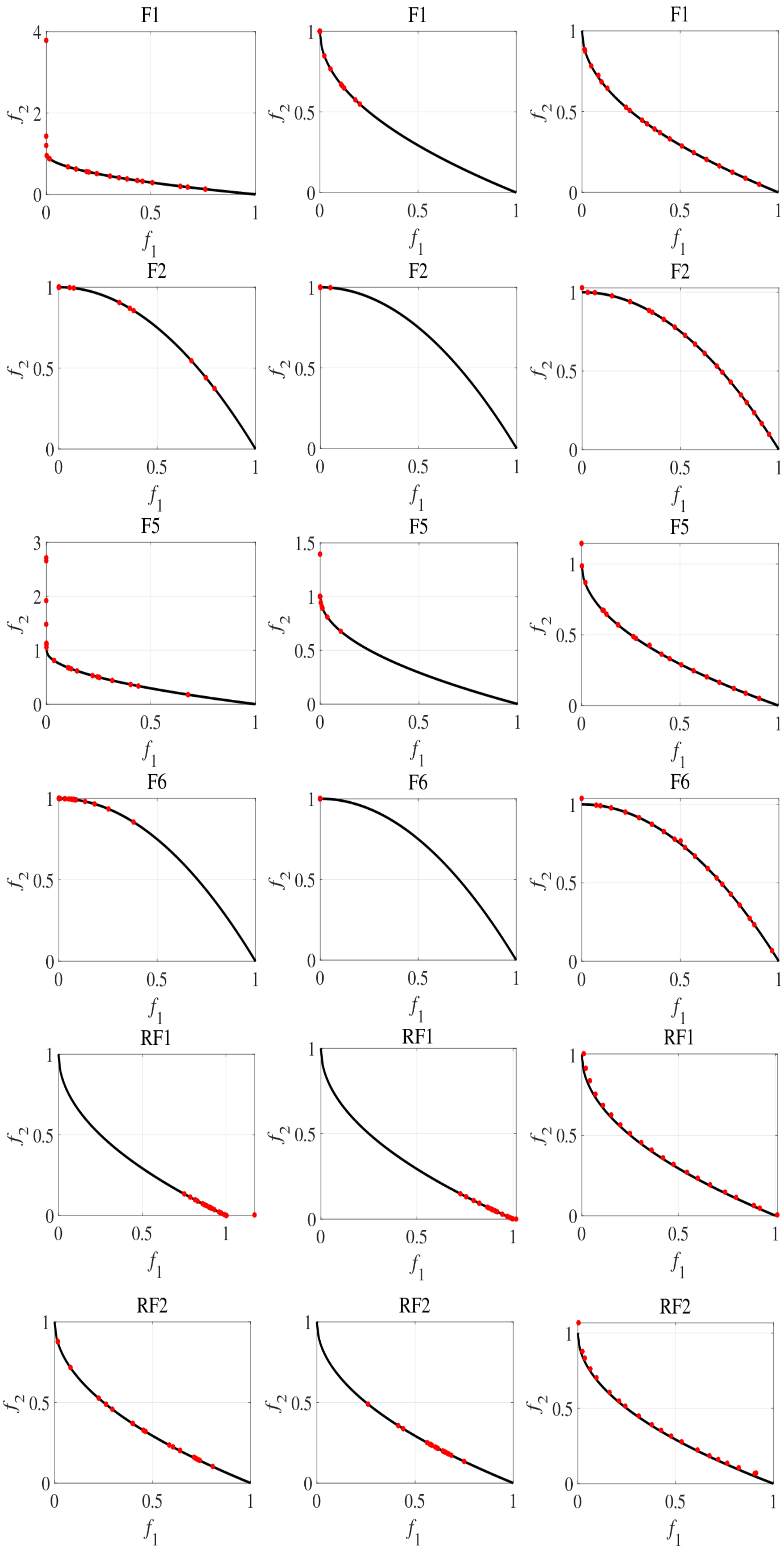
$$g^{tn}(x^k|r) = \max\{f_1(x^k), f_2(x^k)\} = f_I(x^k).$$

After one step iteration, the I^{th} objective function decreases. It means that

$$\begin{aligned} g^{tn}(x^{k+1}|r) &= \max\{f_1(x^{k+1}), f_2(x^{k+1})\} \\ &= f_I(x^{k+1}) < f_I(x^k) \\ &= g^{tn}(x^k|r). \end{aligned}$$

Therefore, $\{g^{tn}(x^k|r)\}$ is a monotone decreasing sequence.

Figure 3 The PFs found by, (a) MGDA (b) ICDM (c) DGD method on the test problems (see online version for colours)



(a) (b) (c)

Case 3 If $p > 0$, $q > 0$, the iteration sequence $\{x^1, x^2, ..., x^p, x^{p+1}, x^{p+2}, ..., x^{p+q}\}$ is a mixed sequence, we only need to prove that the value is decreasing once the point moves from the region Ω_I to the region $\Omega_{\bar{I}}$. We have proved that the sequence with points located in the same region is decreasing. Therefore, we only need to prove $g^{tn}(x^p|r) \le g^{tn}(x^{p+1}|r)$. In fact,

$$g^{tn}(x^p|r) = f_I(x^p),$$
$$g^{tn}(x^{p+1}|r) = \max\{f_1(x^{p+1}), f_2(x^{p+1})\}$$
$$= f_J(x^{p+1}),$$

According to the gradient update rule, we get $x^{p+1} = x^p + t_k(-\nabla f_I(x^p))$, and have the following two cases:

Case 3a If $J = I$, we obtain

$$g^{tn}(x^{p+1}|r) = f_I(x^{p+1}) < f_{\bar{I}}(x^p) = g^{tn}(x^p|r).$$

Case 3b If $J \ne I$, according to the improved Armijo rule (15) and the inequality $f_I(x^{k+1}) \le f_I(x^k) + t_k \nabla f_I(x^k)^T d^k$, we know that the following point of x^p would not move to other regions except for original region Ω_I or the region $\Omega_{\bar{I}}$.

$$\begin{aligned} g^{tn}(x^{p+1}|r) &= f_J(x^{p+1}) \\ &\le f_I(x^{p+1}) \\ &+ \sigma t_k \left\|\nabla f_I(x^p)\right\|^2 \\ &\le f_I(x^p) = g^{tn}(x^p|r). \end{aligned}$$

Based on the two cases discussed above, we know that $g^{tn}(x^{p+1}|r) \le g^{tn}(x^p|r)$, which completes the proof.

From the proof mentioned above, we conclude that, given a random initial point x^0 for each subproblem, there exists a monotone decreasing sequence that satisfies $g^{tn}(x^{k+1}|r) < g^{tn}(x^k|r)$.

Thus we have completed the proof. Under some mild assumption, one immediately obtains a monotone decreasing sequence for each subproblem.

6 Experimental results and analysis

To prove the effectiveness of our method, in this article, we employ test instances (F1–F2, F5–F6) proposed in (Zhang et al., 2008). Moreover, we modify existing test instances given in Li and Zhang (2008), and name them RF1 and RF2. As shown in Table 1, these test instances are bi-objective optimisation problems with linear or nonlinear variable linkages. Meanwhile, these benchmark problems do not have any local Pareto solutions. The shape of their PFs is either convex or concave. We evaluate the performance of MGDA, ICDM, and the proposed DGD method on these test problems.

All these algorithms were implemented through MATLAB R2019a. The quadratic programs (3), equations (5) and (10) are solved by the CVX optimisation toolbox via MATLAB R2019a. The parameter settings, performance metrics, experimental results and analysis for MGDA, ICDM, and DGD method are presented.

6.1 Parameter setting

MGDA, ICDM, and DGD conducted ten independent runs on each test instance. σ = 0.2, the maximum step size is $t_{\max}$ = 0.1. And there are N = 20 initial points for each algorithm. The error of neighboring points obtained by MGDA and ICDM is $\Delta = \|F(x^{k+1}) - F(x^k)\|$, while the error in DGD is $\Delta = |g_l^{tn}(x^{k+1}|r^l) - g_l^{tn}(x^k|r^l)|$, $l = 1, ..., N$. The stopping criterion is that when the algorithm reaches the maximum iteration number ($k_{\max}$ = 500), or the error Δ is less than $\epsilon = 10^{-4}$. $\delta = 0.1 * (\sum_{i=1}^{2} \|\nabla f_i(x)\|^2)$. δ varies with the variable x during iteration.

6.2 Performance metrics

The inverted generational distance (IGD) (Zitzler et al., 2003) is employed to evaluate the performance of these algorithms. The IGD is defined as follows:

$$\text{IGD}(PF^*, PF) = \frac{1}{|PF^*|} \sum_{\mathbf{x} \in PF^*} \min_{\mathbf{y} \in PF} \|\mathbf{x} - \mathbf{y}\|_2, \quad (17)$$

where PF^* is the true PF and PF is the approximated PF obtained by the algorithm.

6.3 Experimental results and analysis

The mean and standard deviation of the IGD of all compared algorithms are presented in Table 2. As shown in Table 2, the IGD of the DGD method is the best on all test problems. Therefore, the DGD gets a better approximation to PFs of test instances than that of MGDA and ICDM.

Table 2 IGD of MGDA, ICDM and DGD

Test problem	*MGDA*	*ICDM*	*DGD*
F1	$0.3291_{0.0499}$	$0.4011_{0.0755}$	$\mathit{0.0309}_{0.0012}$
F2	$0.5138_{0.0876}$	$0.5554_{0.0779}$	$\mathit{0.0218}_{0.0157}$
F5	$0.4994_{0.0808}$	$0.6543_{0.0452}$	$\mathit{0.0258}_{0.0559}$
F6	$0.9169_{0.1302}$	$0.6105_{0.0327}$	$\mathit{0.0256}_{0.0134}$
RF1	$0.3116_{0.0693}$	$0.3095_{0.0432}$	$\mathit{0.0221}_{0.0010}$
RF2	$0.0475_{0.0003}$	$0.1195_{0.0142}$	$\mathit{0.0321}_{0.0064}$

The obtained best PFs in ten independent runs of MGDA, ICDM, and DGD are illustrated in Figure 3. Figure 3(a) shows that MGDA converges to (weakly) Pareto optimal solutions. Some of the obtained solutions are weakly

Pareto optimal for the test instances with convex PF, such as F1 and F5. It is a waste of computational resources because these solutions are not helpful for decision-makers. Moreover, for all test instances with either convex or concave PF, solutions obtained by MGDA distribute unevenly along the PF. For test instances RF1 and RF2, which have a more complex gradient for both objective functions, MGDA converges to the PF of RF2 while performing poorly on RF1. While MGDA performs relatively well in converging to the Pareto optimal solutions of RF1, it could not get an evenly distributed PF. The lousy performance on RF2 may be because the descent direction obtained by MGDA moves quickly toward the direction with the maximum decrease of objectives. Therefore, it is unsuitable for MGDA to solve test instances to obtain a well-distributed PF.

Figure 3(b) shows that ICDM could only converge to one or multiple Pareto solutions. ICDM performs worse than MGDA and DGD on all test instances apart from RF1. ICDM converges to the PFs of all test instances with Pareto solutions packed together. The central descent direction obtained by equation (5) is the bisecting of the angle between $-\nabla f_1(x)$ and $-\nabla f_2(x)$. Therefore, it obtains the crowded approximated PF. Moreover, the bad performance in these instances may owe to the particularity of ICDM. As mentioned in Section 2, ICDM is designed for MOPs with the objective functions having L-Lipschitz continuous gradients.

Figure 3(c) illustrates that the DGD method can obtain a uniformly distributed PF for each test instance. For each NBI-style Tchebycheff subproblem, the iteration sequence converges to a point on the PF irrespective of the starting point. After a finite number of iterations, a well-distributed PF can be obtained. Therefore, the DGD method can obtain an evenly distributed PF within the limited number of iterations for the MOP with differentiable objective functions. Experimental results show that the DGD method, compared to MGDA and ICDM, could converge to an evenly distributed PF. The experimental results demonstrate the effectiveness of the proposed method.

7 Conclusions

This work has investigated the gradient search methods in solving MOPs. Motivated by the shortcomings of the existing gradient descent algorithms, we introduce the NBI-style Tchebycheff method and use it to decompose the MOP into several subproblems. A decreasing sequence for each subproblem is obtained under regularity assumptions by solving the NBI-style Tchebycheff problem with the proposed DGD method. The numerical experiments have illustrated the efficiency of the DGD method in obtaining a well-distributed PF.

In the future, we will concentrate on the MOP with three or more objective functions to illustrate the efficiency of the proposed algorithm.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 62206120.

References

Ansary, M.A.T. and Panda, G. (2021) ‘A globally convergent SQCQP method for multiobjective optimization problems’, *SIAM Journal on Optimization*, Vol. 31, No. 1, pp.91–113.

Bosman, P.A. (2011) ‘On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization’, *IEEE Transactions on Evolutionary Computation*, Vol. 16, No. 1, pp.51–69.

Bowman Jr., V.J. (1976) ‘On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives’, in *Multiple Criteria Decision Making*, pp.76–86, Springer, Berlin, Heidelberg.

Boyd, S., Boyd, S.P. and Vandenberghe, L. (2004) *Convex Optimization*, Cambridge University Press.

Chen, J., Zhang, Q. and Li, G. (2021) ‘MOEA/D for multiple multi-objective optimization’, in *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, pp.152–163.

Coello, C.A.C., Brambila, S.G., Gamboa, J.F., Tapia, M. G.C. and Gómez, R.H. (2020) ‘Evolutionary multiobjective optimization: open research areas and some challenges lying ahead’, *Complex & Intelligent Systems*, Vol. 6, No. 2, pp.221–236.

Désidéri, J-A. (2012) ‘Mutiple-gradient descent algorithm for multiobjective optimization’, in *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*.

Das, I. and Dennis, J.E. (1998) ‘Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems’, *SIAM Journal on Optimization*, Vol. 8, No. 3, pp.631–657.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) ‘A fast and elitist multiobjective genetic algorithm: NSGA-II’, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.

Eichfelder, G. (2009) ‘An adaptive scalarization method in multiobjective optimization’, *SIAM Journal on Optimization*, Vol. 19, No. 4, pp.1694–1718.

Fliege, J. and Svaiter, B.F. (2000) ‘Steepest descent methods for multicriteria optimization’, *Mathematical Methods of Operations Research*, Vol. 51, No. 3, pp.479–494.

Fliege, J. and Vaz, A.I.F. (2016) ‘A method for constrained multiobjective optimization based on SQP techniques’, *SIAM Journal on Optimization*, Vol. 26, No. 4, pp.2091–2119.

Fliege, J., Drummond, L.G. and Svaiter, B.F. (2009) ‘Newton’s method for multiobjective optimization’, *SIAM Journal on Optimization*, Vol. 20, No. 2, pp.602–626.

Gebken, B., Peitz, S. and Dellnitz, M. (2017). ‘A descent method for equality and inequality constrained multiobjective optimization problems’, in *Numerical and Evolutionary Optimization – NEO 2017*, pp.29–61, Springer International Publishing.

Hernández, V.A.S., Schütze, O., Wang, H., Deutz, A. and Emmerich, M. (2018) ‘The set-based hypervolume newton method for bi-objective optimization’, *IEEE Transactions on Cybernetics*, Vol. 50, No. 5, pp.2186–2196.

Jiang, S. and Yang, S. (2015) 'An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts', *IEEE Transactions on Cybernetics*, Vol. 46, No. 2, pp.421–437.

Li, H. and Zhang, Q. (2008) 'Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, pp.284–302.

Li, K., Nie, H., Gao, H. and Yao, X. (2021) 'Posterior decision-making based on decomposition-driven knee point identification', *IEEE Transactions on Evolutionary Computation*, Vol. 26, No. 6, pp.1409–1423.

Lin, X., Zhen, H-L., Li, Z., Zhang, Q-F. and Kwong, S. (2019) 'Pareto multi-task learning', in *Advances in Neural Information Processing Systems*, Vol. 32, pp.12060–12070.

Lin, X., Yang, Z. and Zhang, Q. (2022) *Pareto Set Learning for Neural Multi-Objective Combinatorial Optimization*, arXiv preprint arXiv:2203.15386.

Liu, H-L., Gu, F. and Zhang, Q. (2013) 'Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems', *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 3, pp.450–455.

Mahapatra, D. and Rajan, V. (2020) 'Multi-task learning with user preferences: gradient descent with controlled ascent in Pareto optimization', in *International Conference on Machine Learning, PMLR*, pp.6597–6607.

Martín, A. and Schütze, O. (2018) 'Pareto tracer: a predictor-corrector method for multi-objective optimization problems', *Engineering Optimization*, Vol. 50, No. 3, pp.516–536.

Miettinen, K. (2012) *Nonlinear Multiobjective Optimization*, Vol. 12, Springer Science & Business Media.

Milojkovic, N., Antognini, D., Bergamin, G., Faltings, B. and Musat, C. (2019) *Multi-Gradient Descent for Multi-Objective Recommender Systems*, arXiv preprint arXiv:2001.00846.

Mitrevski, B., Filipovic, M., Antognini, D., Glaude, E.L., Faltings, B. and Musat, C. (2020) *Momentum-based Gradient Methods in Multi-Objective Recommendation*, arXiv preprint arXiv:2009.04695.

Nedjah, N. and Mourelle, L.d.M. (2015) 'Evolutionary multi-objective optimisation: a survey', *International Journal of Bio-Inspired Computation*, Vol. 7, No. 1, pp.1–25.

Ojha, M., Singh, K.P., Chakraborty, P. and Verma, S. (2019) 'A review of multi-objective optimisation and decision making using evolutionary algorithms', *International Journal of Bio-Inspired Computation*, Vol. 14, No. 2, pp.69–84.

Oliveira, I.F. and Takahashi, R.H. (2022) 'An incremental descent method for multi-objective optimization', *Optimization Methods and Software*, Vol. 38, No. 2, pp.312–331.

Sener, O. and Koltun, V. (2018) 'Multi-task learning as multi-objective optimization', *Advances in Neural Information Processing Systems*, Vol. 31, pp.527–538.

Thomann, J. and Eichfelder, G. (2019) 'A trust-region algorithm for heterogeneous multiobjective optimization', *SIAM Journal on Optimization*, Vol. 29, No. 2, pp.1017–1047.

Trivedi, A., Srinivasan, D., Sanyal, K. and Ghosh, A. (2016) 'A survey of multiobjective evolutionary algorithms based on decomposition', *IEEE Transactions on Evolutionary Computation*, Vol. 21, No. 3, pp.440–462.

Vieira, D.A., Takahashi, R.H. and Saldanha, R.R. (2012) 'Multicriteria optimization with a multiobjective golden section line search', *Mathematical Programming*, Vol. 131, No. 1, pp.131–161.

Wang, J., Hu, Y., Wai Yu, C.K., Li, C. and Yang, X. (2019) 'Extended newton methods for multiobjective optimization: majorizing function technique and convergence analysis', *SIAM Journal on Optimization*, Vol. 29, No. 3, pp.2388–2421.

Wang, Y., Gao, W., Gong, M., Li, H. and Xie, J. (2022) 'A new two-stage based evolutionary algorithm for solving multi-objective optimization problems', *Information Sciences*, Vol. 611, pp.649–659.

Wu, M., Li, K., Kwong, S., Zhang, Q. and Zhang, J. (2018) 'Learning to decompose: a paradigm for decomposition-based multiobjective optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 23, No. 3, pp.376–390.

Xu, J., Tian, Y., Ma, P., Rus, D., Sueda, S. and Matusik, W. (2020) 'Prediction-guided multi-objective reinforcement learning for continuous robot control', in *International Conference on Machine Learning*, PMLR, pp.10607–10616.

Zadeh, L. (1963) 'Optimality and non-scalar-valued performance criteria', *IEEE Transactions on Automatic Control*, Vol. 8, No. 1, pp.59–60.

Zhang, Q. and Li, H. (2007) 'MOEA/D: a multiobjective evolutionary algorithm based on decomposition', *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 6, pp.712–731.

Zhang, Q., Zhou, A. and Jin, Y. (2008) 'RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm', *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 1, pp.41–63.

Zhang, Q., Li, H., Maringer, D. and Tsang, E. (2010) 'MOEA/D with NBI-style tchebycheff approach for portfolio management', in *IEEE Congress on Evolutionary Computation*, IEEE, pp.1–8.

Zhou, A. and Zhang, Q. (2015) 'Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms', *IEEE Transactions on Evolutionary Computation*, Vol. 20, No. 1, pp.52–64.

Zhou, A., Qu, B-Y., Li, H., Zhao, S-Z., Suganthan, P.N. and Zhang, Q. (2011) 'Multiobjective evolutionary algorithms: a survey of the state of the art', *Swarm and Evolutionary Computation*, Vol. 1, No. 1, pp.32–49.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M. and Da Fonseca, V.G. (2003) 'Performance assessment of multiobjective optimizers: an analysis and review', *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, pp.117–132.