

# Hypervolume Indicator Gradient Ascent Multi-objective Optimization

Hao Wang, André Deutz, Thomas Bäck, and Michael Emmerich

Leiden Institute of Advanced Computer Science  
Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands  
`{h.wang, a.h.deutz, t.h.w.baeck, m.t.m.emmerich}@liacs.leidenuniv.nl`

**Abstract.** Many evolutionary algorithms are designed to solve black-box multi-objective optimization problems (MOPs) using stochastic operators, where neither the form nor the gradient information of the problem is accessible. In some real-world applications, e.g. surrogate-based global optimization, the gradient of the objective function is accessible. In this case, it is straightforward to use a gradient-based multi-objective optimization algorithm to achieve fast convergence speed and the stability of the solution. In a relatively recent approach, the hypervolume indicator gradient in the decision space is derived, which paves the way for the method for maximizing the hypervolume indicator of a fixed size population. In this paper, several mechanisms which originated in the field of evolutionary computation are proposed to make this gradient ascent method applicable. Specifically, the well-known non-dominated sorting is used to help steering the dominated points. The principle of the so-called cumulative step-size control that is originally proposed for evolution strategies is adapted to control the step-size dynamically. The resulting algorithm is called Hypervolume Indicator Gradient Ascent Multi-objective Optimization (HIGA-MO). The proposed algorithm is tested on ZDT problems and its performance is compared to other methods of moving the dominated points as well as to some evolutionary multi-objective optimization algorithms that are commonly used.

**Keywords:** Set-based scalarization, hypervolume indicator, gradient ascent, non-dominated sorting, cumulative step-size control

## 1 Introduction

Different multi-objective optimization algorithms have been proposed and exploited in real-world problem over the years, e.g. NSGA-II [2], SPEA2 [20] and SMS-EMOA [1]. These evolutionary multi-criteria optimization (EMO) algorithms employ heuristic operators (e.g. random variation and selection operators), instead of using the gradient information of the objective functions. For a large subclass of such problems, that is the **continuous** multi-objective optimization problem, gradient-based algorithms are of interest due to the fact that they are generally fast, precise and stable with respect to local convergence. Various gradient-based approaches have been proposed for the multi-objective

optimization task [5, 9, 7, 13]. A relatively new idea is proposed by [4, 3], in which the gradient of the hypervolume indicator with respect to a set of decision vectors is computed. In this paper, we adopt the definition and the computation of the *hypervolume indicator gradient* to steer the search points within the decision space. By using the hypervolume indicator gradient [19], the search points are moved in the direction of steepest ascent in the hypervolume indicator. Therefore, the proposed algorithm is termed *hypervolume indicator gradient ascent multi-objective optimization* (HIGA-MO). The major benefit of exploiting hypervolume gradients are 1) the points in the objective space will be well distributed on the Pareto front, 2) it is almost free of control parameters, and 3) the algorithm has a high precision of convergence to the Pareto front.

However, the first implementation of this idea showed numerical problems. As a remedy, ideas that were developed in the field of evolutionary multi-criterion optimization are adopted in this paper. Firstly, the hypervolume indicator may have zero gradient components at some decision vectors, e.g., the dominated points. The well-known non-dominated sorting technique is adopted and combined with the hypervolume indicator gradient computation, in order to equip each decision vector with a multi-layered gradient. Secondly, the normalization of the hypervolume indicator *sub-gradient* is used to overcome the “creepiness” phenomenon observed in earlier versions of hypervolume gradient ascent, and caused by an imbalance in the length of sub-gradients which leads to a slow convergence speed [14]. Thirdly, the usage of constant step-sizes is no longer appropriate if precise convergence to the Pareto front is aimed for. Instead, a cumulative step-size control inspired by the optimal gradient ascent is proposed to dynamically adapt the step-size. Such a cumulative step-size control resembles the step-size adaptation mechanism in the well-known CMA-ES [6], an evolutionary algorithm for single objective continuous optimization. The resulting algorithm is tested on problems named ZDT1-4 and ZDT6 from [18]. Its performance is compared to three evolutionary algorithms: NSGA-II [2], SPEA2 [20] and SMS-EMOA [1], as well as the other methods for steering the dominated points.

This paper is organized as follows. In section 2, the multi-objective optimization problem and some notations used in this paper are introduced. In section 3, derivations of the hypervolume indicator gradient are revisited with simplification of notations. The method for steering the dominated points is discussed in section 4. The cumulative step-size control method is illustrated in section 5. In section 7, an experimental study of the resulting algorithm is conducted on the ZDT problems. Finally, we conclude the paper and suggest potential future improvements on HIGA-MO.

## 2 Background and Notations

In this section, a brief introduction of the problem and terminology that will be used later is given. In multi-objective optimization problems (MOPs), a collection of functions, represented as the  $m$ -tuple below, are optimized simultane-

ously:

$$(f_1 : S_1 \rightarrow \mathbb{R}, f_2 : S_2 \rightarrow \mathbb{R}, \dots, f_m : S_m \rightarrow \mathbb{R}), \quad S_1, S_2, \dots, S_m \subseteq \mathbb{R}^d$$

where  $d$  denotes the dimension of the domain of each function and  $m$  denotes the number of objective functions. Without loss of generality, we assume all the functions above are to be maximized (minimization problem can be transformed from maximization). In this work, it is assumed that each objective function  $f_i$  is continuous differentiable *almost everywhere* in  $S_i$ . Thus, the MOP can be formulated as follows:

$$\max_{\mathbf{x} \in S} \mathbf{f}(\mathbf{x}), \quad S = \bigcap_{i=1}^m S_i \subseteq \mathbb{R}^d,$$

where  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]^\top$  is a vector-valued function composed of  $m$  objective functions:

$$\mathbf{f} : S \rightarrow \mathbb{R}^m$$

Due to the continuous differentiability assumption on each objective function,  $\mathbf{f}$  is again continuous differentiable almost everywhere in  $S$ . The gradient information is expressed as transpose of the Jacobian matrix as follows:

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = [\nabla f_1(\mathbf{x}), \nabla f_2(\mathbf{x}), \dots, \nabla f_m(\mathbf{x})], \quad \nabla f_i(\mathbf{x}) : S \rightarrow \mathbb{R}^d, \quad i = 1, 2, \dots, m$$

In addition, it is assumed that each gradient vector above (column vector) can be computed either analytically or numerically. In MOPs, a set of decision vectors are moved in *decision space*  $S$  to approximate the Pareto efficient set, which is the so-called Pareto efficient set approximation:

$$X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\mu)}\}, \quad \mathbf{x}^{(i)} \in S, \quad i = 1, 2, \dots, \mu$$

with corresponding Pareto front approximation set (objective vectors) in the *objective space*:

$$Y = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(\mu)}\}, \quad \mathbf{y}^{(i)} = \mathbf{f}(\mathbf{x}^{(i)}) \in \mathbb{R}^m, \quad i = 1, 2, \dots, \mu$$

In order to measure and compare the quality among Pareto front approximation sets  $Y$ , one approach is to quantify the quality by constructing a proper indicator. The most common one is the hypervolume indicator  $H$  [21, 22]. Given a reference point  $\mathbf{r} \in \mathbb{R}^m$ , the hypervolume indicator of the Pareto front approximation set  $Y$  can be expressed as:

$$H(Y; \mathbf{r}) = \lambda \left( \bigcup_{\mathbf{y} \in Y} [\mathbf{r}, \mathbf{y}] \right),$$

where  $\lambda$  denotes the Lebesgue measure on  $\mathbb{R}^m$ , which is the size of the hypervolume dominated by the approximation set  $Y$  with respect to the reference space. Note that the reference point  $\mathbf{r}$  will be assumed to be a given constant and thus omitted in the following notations for brevity.

### 3 Hypervolume Indicator Gradient

The hypervolume indicator gradient is defined as the gradient of the hypervolume indicator with respect to the approximation of the Pareto efficient set, which is proposed in [3, 4]. In this work, the derivation of the hypervolume indicator gradient is reformulated and the notation is simplified. In the following, we shall use matrix calculus notations with denominator layout, meaning that the derivative of a vector/matrix is laid out according to the denominator.

Intuitively, the hypervolume indicator can be expressed as a function of the Pareto efficient set approximation  $X$ , which allows for the differentiation of hypervolume indicator with respect to decision vectors. More specifically, by concatenation of all the vectors in this set, we obtain a so-called  $\mu \cdot d$ -vector:

$$\mathbf{X} = \left[ \mathbf{x}^{(1)\top}, \mathbf{x}^{(2)\top}, \dots, \mathbf{x}^{(\mu)\top} \right]^\top \in S^\mu \subseteq \mathbb{R}^{\mu \cdot d}$$

and its corresponding Pareto front approximation vector can be written as a  $\mu \cdot m$ -vector:

$$\mathbf{Y} = \left[ \mathbf{y}^{(1)\top}, \mathbf{y}^{(2)\top}, \dots, \mathbf{y}^{(\mu)\top} \right]^\top \in \mathbb{R}^{\mu \cdot m}$$

In order to establish a connection between  $\mu \cdot d$ -vectors and  $\mu \cdot m$ -vectors, we define a mapping  $\mathbf{F} : S^\mu \rightarrow \mathbb{R}^{\mu \cdot m}$ :

$$\mathbf{F}(\mathbf{X}) := \left[ \mathbf{f}(\mathbf{x}^{(1)})^\top, \mathbf{f}(\mathbf{x}^{(2)})^\top, \dots, \mathbf{f}(\mathbf{x}^{(\mu)})^\top \right]^\top$$

Now consider that the hypervolume indicator, that is normally defined in the objective space, can be re-written as a function of  $\mu \cdot d$ -vectors by composition:

$$\mathcal{H}_{\mathbf{F}}(\mathbf{X}) := H(\mathbf{F}(\mathbf{X})),$$

which is a continuous mapping from  $S^\mu$  to  $\mathbb{R}$ , for which under certain regularity conditions the gradient is defined (in case of differentiable objective functions only for a zero measure subset of  $\mathbb{R}^{\mu \cdot d}$  the gradient is undefined, in which case one-sided derivatives still exist). Given  $\mathcal{H}_{\mathbf{F}}$ , its derivatives are defined (given they exist) by:

$$\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{X}} = \left[ \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(1)}}^\top, \dots, \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(\mu)}}^\top \right]^\top, \quad (1)$$

where each of the term in the RHS of the equation above is called *sub-gradient*, which is the local hypervolume change rate by moving each decision vector infinitesimally. It has been shown in [3] that the hypervolume indicator gradient is the concatenation of the hypervolume contribution gradients. Moreover, the sub-gradients can be calculated by applying the chain rule:

$$\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}} = \frac{\partial \mathbf{y}^{(i)}}{\partial \mathbf{x}^{(i)}} \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{y}^{(i)}} \quad (2)$$

$$= \sum_{k=1}^m \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial f_k(\mathbf{x}^{(i)})} \nabla f_k(\mathbf{x}^{(i)}) \quad (3)$$

The first partial derivative in Eq. 2 is the gradient of  $\mathcal{H}_{\mathbf{F}}$  in the objective space while the second one is the transpose of the Jacobian matrix of the mapping  $\mathbf{F}$ . Eq. 3 is the detailed form. From it, it is clear that the hypervolume indicator gradient is a *linear combination of gradient vectors of objective functions*, where the weight for an objective function is the partial derivative of the hypervolume indicator at this objective value. We omit the calculation for gradients of  $\mathcal{H}_{\mathbf{F}}$  in the objective space for simplicity, noting that in the bi-objective case they correspond to the length of the steps of the attainment curve. For the high dimensional case and efficient computation, see [3].

Note that in practice the length of the sub-gradients usually differ by orders of magnitude, leading to the “creepiness” behavior [14] that some decision vectors move much faster than the rest. Such a behavior results in a very slow convergence speed and points might get dominated by others. As a remedy, it is suggested to normalize all the sub-gradients.

## 4 Steering Dominated Points

The difficulty increases when applying the hypervolume indicator gradient direction for steering the decision vectors: the hypervolume indicator can either be zero or only one-sided at decision vectors. For example, at every strictly dominated search point, the hypervolume indicator sub-gradient is zero, because the Pareto front and thus the hypervolume indicator remain unchanged if it is moved locally in an infinitesimally small neighborhood. For every weakly dominated point, the hypervolume indicator sub-gradient at this point, even does not exist due to the fact that only one-sided partial derivatives exist. Consequently, such decision vectors will become stationary in gradient ascent method. One obvious solution to such a problem is to apply evolutionary operators (mutation and crossover) on those search points (decision vectors) until they become non-dominated. However, as we are aiming for a fully deterministic multi-objective optimization algorithm, randomized operators are not adopted in this work.

Some methods have been proposed to steer dominated points [11, 17, 9]. The most prominent one, proposed by Lara [9], computes the gradient at dominated points as follows (for bi-objective problems):

$$\frac{\nabla f_1(\mathbf{x}^{(i)})}{\|\nabla f_1(\mathbf{x}^{(i)})\|} + \frac{\nabla f_2(\mathbf{x}^{(i)})}{\|\nabla f_2(\mathbf{x}^{(i)})\|}, \quad \mathbf{x}^{(i)} \text{ is dominated}$$

which is a sum of normalized gradients of each objective function. It guarantees that dominated decision vectors move into the *dominance cone* [17]. However, such a method only considers the movement of single points, instead of a set of search points and it does not generalize to more than two dimensions. We shall call this method **Lara’s direction** in the following experiments, where it is compared with the method proposed in this work. Another method for steering the dominated points is proposed by the authors in [17]. It steers dominated points towards the nearest gap on the non-dominated set. The search direction is determined as the gradient of the distance of the dominated objective vector

to the center of its nearest gap. Again, this method steers dominated points independently and is termed as **Gap-filling** in this paper. In the above methods, dominated points are steered widely independent of each other, which might result in diversity loss.

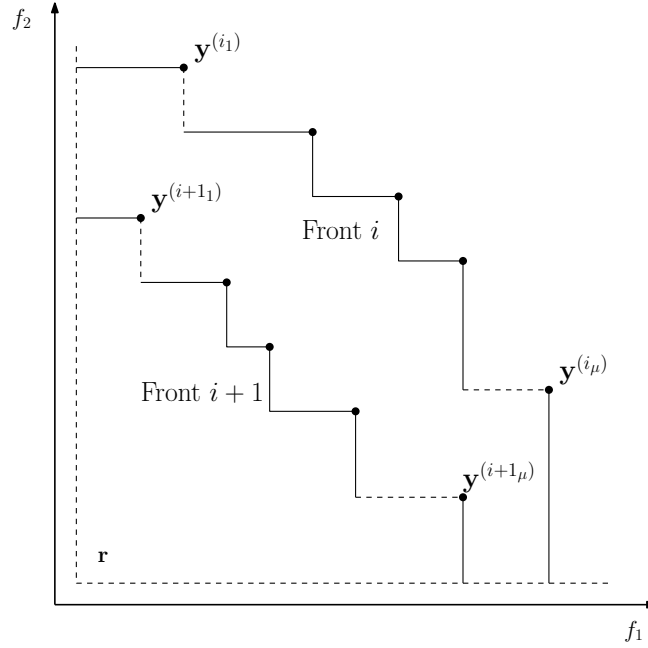
In this work, we propose to use the *non-dominated sorting* technique that is developed in the NSGA-II algorithm [15], in order to compute the hypervolume indicator gradients of multiple layers of non-dominated sets. In detail, the decision and objective vectors are partitioned into  $q$  subsets, or *layers* according to their dominance rank in the objective space:

$$\mathbf{X} \rightarrow \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^q\},$$

$$\mathbf{X}^i = \left[ \mathbf{x}^{(i_1)\top}, \mathbf{x}^{(i_2)\top}, \dots, \mathbf{x}^{(i_\mu)\top} \right]^\top,$$

where  $X^i$  indicates a layer of order  $i$  and  $i_\mu$  denotes the number of decision vectors in the  $i$ th rank layer. The layers can be recursively defined as (given  $ND$  as the operator that selects the non-dominated subset from approximation set):

$$X^1 = ND(X), \quad X^{i+1} = ND\left(X - \bigcup_{j=1}^i X^j\right),$$



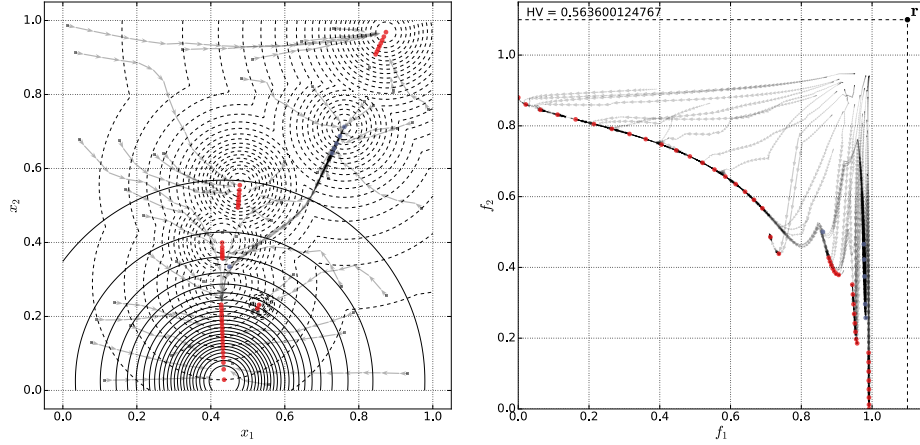
**Fig. 1.** Schematic graph showing the partition of the objective vectors using non-dominated sorting. For each partition (layer), a hypervolume indicator is defined and thus its gradient can be computed.

where  $q$  is the highest index  $i$  such that  $X^i \neq \emptyset$ . Note that the  $\mu \cdot m$ -vector is also partitioned as above. In principle, it is possible to compute the hypervolume indicator gradient for any layer by ignoring all the layers that dominate it (have a lower rank) temporarily. This partition is illustrated in Fig. 1. In this manner, the hypervolume volume indicator gradient on the whole approximation set  $\mathbf{X}$  can be (re-)defined as the concatenation of the hypervolume indicator gradient on each layer:

$$\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{X}} := \left[ \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X}^1)}{\partial \mathbf{X}^1}^\top, \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X}^2)}{\partial \mathbf{X}^2}^\top, \dots, \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X}^q)}{\partial \mathbf{X}^q}^\top \right]^\top \quad (4)$$

Note that again  $q$  is the number of layers obtained from non-dominated sorting techniques. The gradient computation given in Eq. 3 can be used to compute each gradient term above. Thus, each decision vector is associated with a steepest ascent direction that maximizes its hypervolume contribution on each layer.

There are two advantages of using the non-dominated sorting procedure. Firstly, maximizing the hypervolume will not only steer the points towards the Pareto front, but also spread out the points across the intermediate Pareto front approximation. By applying the hypervolume indicator gradient direction on each layer, the decision vectors on each layer will be well distributed before a dominated layer merges into the global Pareto front and thus the additional cost to spread out points after the merging is small. Moreover, when the Pareto efficient set is disconnected in the decision space, the proposed approach will increase the convergence speed due to the fact that each connected efficient set



**Fig. 2.** Trajectories of 50 points under hypervolume indicator gradient direction to approximate the Pareto front using  $10^3$  function evaluations. The experiment is conducted on a bi-objective problem MPM2 (from R *smoof* package) in the 2-D decision space. All five disconnected components of the Pareto front are obtained with well distributed points. **Left:** the decision space. **Right:** the objective space.

is treated as one layer and the decision vectors on it are spread quickly over the efficient sets. This effect can be shown by visualizing the trajectories of the approximation set on a simple objective landscape. In Fig. 2, trajectories of the approximation set are illustrated in both decision and objective space, on MPM2 functions (from the R *smoof* package<sup>1</sup>). In the decision space, it is clear that our layering approach (Fig. 2) manages to approximate five disconnected efficient sets with a good distribution of points.

Secondly, on the real landscape, it is possible that local Pareto fronts exist (e.g. consider the well-known ZDT4 problem [18]). Using the non-dominated sorting, it is more likely to identify those local Pareto fronts, which could be helpful to balance global and local search. This advantage of the proposed approach is exploited by the authors in multi-objective multi-modal landscape analysis [8].

## 5 Step-size adaptation

The constant step-size setting that is common in gradient descent (ascent) for the single objective optimization task, is no longer appropriate. Usually, the length of the gradient vector (in the gradient field) gradually goes to zero when approaching the local optimum. In this case, a properly set constant step-size will lead to the local optimum in a stable manner. However, in our case, due to the normalization, the length of the search steps is always 1 when decision vectors are approaching the Pareto efficient set. If a constant step-size is applied here, the decision vector will *overshoot* its optimal position and begin to oscillate (even diverge). In order to tackle this issue, the step-size of the decision vectors needs to 1) gradually decrease when approaching the Pareto efficient set and 2) increase quickly when the decision vectors are far away from the efficient set. In addition, it is reasonable to use individual step-sizes that are controlled independently for each decision vector because their optimal step-size differs largely.

A cumulative step-size adaptation mechanism is proposed to approximate the optimal step-size in the optimization process. It is inspired by the following observation: in single objective gradient optimization, if the step-size is set optimally, then consecutive search direction are perpendicular to each other. In order to approximate the optimal step-size setting, the inner product of consecutive normalized hypervolume indicator gradients is calculated. If such an inner product is positive, it indicates the current step-size is smaller than the optimal one and vice versa:

$$I_t^{(i)} = \left\langle \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})^{(t-1)}}{\partial \mathbf{x}^{(i)}}, \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})^{(t)}}{\partial \mathbf{x}^{(i)}} \right\rangle, \quad i = 1, \dots, \mu, \quad t = 1, 2, \dots$$

Note that superscripts  $(t)$ ,  $(t-1)$  are iteration indices. In addition, such an inner product computed in each iteration fluctuates hugely and direct use of it leads to unstable adaptation behavior. Therefore, the inner product is cumulated using

<sup>1</sup> <https://github.com/jakobbossek/smoof>



exponentially decreasing weights through the iterations to get a more stable indicator for the step-size adaptation. The cumulative rule for the inner product is written as follows:

$$p_t^{(i)} \leftarrow (1 - c) \cdot p_{t-1}^{(i)} + c \cdot I_t^{(i)}, \quad i = 1, \dots, \mu, \quad t = 1, 2, \dots \quad (5)$$

Note that  $p_t^{(i)}$  denotes the cumulated inner product for search point  $i$  at iteration  $t$  and  $c$  ( $0 < c < 1$ ) is the accumulation coefficient. Such an inner product accumulation rule is similar to the cumulative step-size adaptation mechanism in the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6], where consecutive mutation steps are accumulated for step-size adaptation. Based on the cumulated inner product, a simple control rule is designed to adapt the step-size online:

$$\sigma_{t+1}^{(i)} = \begin{cases} \sigma_t^{(i)} \cdot \alpha & \text{if } p_t^{(i)} < 0, \\ \sigma_t^{(i)} & \text{if } p_t^{(i)} = 0, \\ \sigma_t^{(i)} / \alpha & \text{if } p_t^{(i)} > 0. \end{cases} \quad 0 < \alpha < 1 \quad (6)$$

where  $\sigma_t^{(i)}$  is the individual step-size for search point  $i$  at iteration  $t$ . This rule dictates that 1) if the inner product accumulation is positive, then the step-size is increased by a factor of  $\alpha$ , 2) if the inner product accumulation is negative, the step-size is decrease by a factor of  $\alpha$ , and 3) otherwise, the step-size remains unchanged. In this work, the settings of  $c = 0.7, \alpha = 0.8$  are suggested by tuning the algorithmic performance on MPM2 functions.

The backtracking line search [10], which is a common technique to approximate the optimal step-size in single objective gradient ascent, is not suitable for the proposed algorithm. It requires additional function evaluations for each search point to estimate the optimal step-size setting. Such additional costs are no longer acceptable for the set-based algorithm. In contrast, the proposed cumulative step-size adaptation mechanism does not bring any additional overheads.

## 6 Hypervolume Indicator Gradient Ascent Algorithm

In this section, the algorithmic components developed in the previous sections are combined into the hypervolume indicator gradient ascent algorithm.

### 6.1 Handling non-differentiable points

In practice, the continuous objective function can be non-differentiable at some points, even the function is almost everywhere differentiable (e.g. on the constraint boundary of the ZDT1 problem). To overcome this issue, it is suggested to *mutate* those points in the decision space. Given a point  $\mathbf{x} \in \mathbb{R}^d$ , it is mutated in the decision space  $S$  when the gradient of objective functions at  $\mathbf{x}$  contains invalid values (e.g. infinite). The mutation of  $\mathbf{x}$  should be local but large enough to escape from the non-differentiable regions. For this purpose, then mutation

operator in Differential Evolution [16] is adopted here because it is adaptive and only contains a single parameter. Suppose  $\mathbf{x}$  is in the  $i$ th ranked layer ( $\mathbf{x} \in \mathbf{X}^i$ ), then it is mutated as follows:

$$\mathbf{x} \leftarrow \mathbf{x} + F \cdot (\mathbf{x}^{(a)} - \mathbf{x}^{(b)}), \quad (7)$$

where  $\mathbf{x}^{(a)}, \mathbf{x}^{(b)}$  are randomly picked from  $\mathbf{X}^i$ .  $F \in [0, 2]$  is the differential weight that is set according to the literature. It is necessary to compute the differential vector within the same layer of  $\mathbf{x}$  because the Pareto efficient set is possibly disconnected in the decision space and differential vectors computed across layers possibly create non-local mutations.

## 6.2 Pseudo-code

---

### Algorithm 1 Hypervolume Indicator Gradient Ascent Multi-Objective Optimization

---

```

1 Initialize the search points  $\mathbf{X}$  uniformly in the search space
2 while the termination criteria are not satisfied do
3   Evaluation:  $\mathbf{Y} \leftarrow \mathbf{X}$ .
4    $\{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^q\} \leftarrow \text{non-dominated-sorting}(\mathbf{X})$ 
5   for  $k = 1$  to  $q$  do
6     for every  $\mathbf{x}^{(i)}$  in  $\mathbf{X}^k$  do
7       Compute the sub-gradient  $\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}}$  according to Eq. 2
8       if  $\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}_j}$  is undefined then
9          $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} + F \cdot (\mathbf{x}^{(a)} - \mathbf{x}^{(b)})$ , where  $\mathbf{x}^{(a)}, \mathbf{x}^{(b)}$  are randomly picked from  $\mathbf{X}^k$ 
10      else
11        Sub-gradient normalization:  $\mathbf{g}^{(i)} \leftarrow \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}} / \|\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}}\|$ 
12        Gradient ascent:  $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} + \sigma^{(i)} \mathbf{g}^{(i)}$ 
13        Apply the step-size control according to Eq. 5 and 6.
14      end if
15    end for
16  end for
17 end while
18 return  $\mathbf{X}, \mathbf{Y}$ 
```

---

The resulting algorithm is presented in Alg. 1. In line 4, the non-dominated sorting procedure is called to partition the approximation set. In line 7 the hypervolume indicator gradient is computed for every decision vector on each layer. If a decision vector has either zero gradient or is not differentiable, it is mutated in line 9 according to Eq. 7. In line 11, the hypervolume indicator sub-gradient is normalized before decision vectors are moved in the steepest ascent manner (line 12). The cumulative step-size adaptation (Eq. 5 and 6) is then applied in line 13. In addition to the common usage of the function evaluation budget for the termination criterion, it is suggested here to check stationarity of search points: a decision vector is considered stationary if the norm of its sub-gradient multiplied by the step-size is close to zero.

## 7 Experiments

*Experiment settings* To test the performance of HIGA-MO, the well-known ZDT problems [2] are selected as benchmark problem set. The proposed algorithm is compared to three well-established evolutionary multi-objective optimization algorithm: NSGA-II, SPEA2 and SMS-EMOA. The parameters in those two algorithms are set according to the literature [2, 1, 20]. In addition, other methods for steering the dominated point (Section 4), Lara’s direction and Gap-filling, are tested against HIGA-MO. For these two methods, the non-dominated points are moved using the hypervolume indicator gradient.

The hypervolume indicator and convergence measure used in [1], are adopted here as the performance metrics. The convergence measure is calculated numerically by discretizing the Pareto front into 1000 points. For the hypervolume indicator computation, the reference point  $[11, 11]^\top$  is used for the test problems ZDT1-4 and ZDT6. Two experiments are conducted: one with a relatively small population setting  $\mu = 40$  while the other uses a large population,  $\mu = 100$ . A relatively small function evaluation budget,  $100\mu$  is chosen here due the reason that in long runs, all deterministic methods stagnate to local optima. All the algorithms terminate if the maximal function evaluation budget is reached. For each algorithm, 15 independent runs are conducted to obtain average performance measures. The initial step-size of the proposed HIGA-MO algorithm is set to 0.05 multiplied by the maximum range of the decision space. The internal reference point to compute the hypervolume indicator gradient is set to  $[11, 11]^\top$  to ensure every objective vector is within the reference space.

*Results* The test results are shown in Table 1 for  $\mu = 40$  and Table 2 for  $\mu = 100$ . The hypervolume of the non-dominated set after termination is used to compute the performance measures. For the small population setting, HIGA-MO outperforms the evolutionary algorithms (NSGA-II, SPEA2 and SMS-EMOA) on ZDT1-3 and ZDT6 problems, both in terms of hypervolume indicator and convergence measure. By checking the standard deviation, it is obvious that HIGA-MO generates more stable results compared to evolutionary algorithms and such deviations are only affected by the initialization of the approximation set and the technique to handle the non-differentiable points (Eq. 7). Comparing it to the other two methods, namely, Lara’s direction and Gap-filling, that steer the dominated points independently, HIGA-MO gives a higher hypervolume indicator value on ZDT1-3 while Lara’s method performs better on ZDT6. In terms of the convergence measure, Lara’s direction always outperforms HIGA-MO on ZDT1-3 and 6. Lara’s direction moves the dominated points toward the Pareto front without considering the distribution of them while HIGA-MO is designed to achieve both. Thus, HIGA-MO requires more efforts to approach the Pareto front than Lara’s direction, in terms of the convergence measure. On ZDT4, which has a highly multi-modal landscape, none of the gradient-based methods (HIGA-MO, Lara’s direction and Gap-filling) achieves comparable results to evolutionary algorithms. **The gradient-based methods easily stagnate in the local Pareto-front and fail to move towards the global one.** For such a

**Table 1.**  $\mu = 40$ : performance measures on ZDT1-4 and ZDT6 problems.

Test-function	Algorithm	Convergence measure			Hypervolume indicator		
		Average	Std. dev.	Rank	Average	Std. dev.	Rank
ZDT1	HIGA-MO	<b>0.00500490</b>	1.3075e-02	1	<b>120.62948062</b>	4.0750e-03	1
	Lara’s direction	0.07747718	6.4031e-02	3	120.33761711	1.2309e-01	2
	Gap-filling	0.06061863	1.2352e-01	2	120.22307239	4.6840e-01	3
	NSGA-II	0.10960371	3.2542e-02	5	119.33541376	3.7345e-01	4
	SMS-EMOA	0.09376444	3.5934e-02	4	119.20965862	4.8101e-01	5
	SPEA2	0.32006024	5.9788e-02	6	116.27370195	1.6826e+00	6
ZDT2	HIGA-MO	0.00036082	3.6233e-05	3	<b>120.31634691</b>	9.8307e-04	1
	Lara’s direction	<b>0.00011253</b>	5.0289e-05	1	118.92812930	3.5019e+00	3
	Gap-filling	0.00015973	2.0645e-04	2	119.45871166	2.5324e+00	2
	NSGA-II	0.16511979	7.7092e-02	4	114.03423180	3.7806e+00	4
	SMS-EMOA	0.24929199	8.4178e-02	5	109.17629732	3.2584e+00	5
	SPEA2	0.67688451	1.5708e-01	6	104.54506810	3.3537e+00	6
ZDT3	HIGA-MO	0.00031903	5.0492e-05	2	128.55259300	7.9970e-01	2
	Lara’s direction	<b>0.00028076</b>	5.0842e-05	1	125.78304061	3.5114e+00	6
	Gap-filling	0.00034568	5.4557e-05	3	<b>128.75911576</b>	9.2658e-03	1
	NSGA-II	0.00228282	5.9689e-03	4	126.56081625	2.8857e+00	3
	SMS-EMOA	0.00405046	5.7238e-03	5	125.88966563	2.9289e+00	5
	SPEA2	0.00635668	1.0852e-02	6	126.55026001	2.5895e+00	4
ZDT4	HIGA-MO	38.13060527	7.6780e+00	4	0.00000000	0.0000e+00	6
	Lara’s direction	43.19742796	1.1544e+01	5	0.00000000	0.0000e+00	5
	Gap-filling	52.35972878	1.2465e+01	6	1.16325406	4.3525e+00	4
	NSGA-II	4.07411956	1.6869e+00	2	75.28344930	1.8038e+01	2
	SMS-EMOA	<b>3.52099683</b>	1.7386e+00	1	<b>78.04608227</b>	1.8555e+01	1
	SPEA2	11.17677922	4.9514e+00	3	19.34577362	2.2000e+01	3
ZDT6	HIGA-MO	3.83694298	1.3668e+00	6	113.28359226	1.3577e+00	2
	Lara’s direction	<b>0.00010409</b>	4.3909e-05	1	<b>116.86127498</b>	1.6820e+00	1
	Gap-filling	3.02249489	2.7090e+00	5	106.81768735	2.0573e+01	3
	NSGA-II	1.28139859	3.0071e-01	2	97.53535725	3.8143e+00	4
	SMS-EMOA	1.36426329	3.1163e-01	3	96.84386232	4.2309e+00	5
	SPEA2	2.22799304	7.2398e-01	4	86.25780584	7.9570e+00	6

highly multi-modal optimization problem, a restart heuristic could improve the performance of gradient-based algorithms. For the large population setting, Table 2 shows roughly the same results for algorithm comparisons as for the small population setting.

## 8 Conclusions

In this paper, a full gradient-based multi-objective optimization algorithm is proposed. The gradient direction is derived by differentiating the hypervolume indicator with respect to the concatenation of decision vectors. Moreover, several techniques are devised to solve difficulties in applying the hypervolume indicator gradient to the approximation set: 1) the non-dominated sorting procedure is used to steer the dominated points using the hypervolume indicator gradient.

**Table 2.**  $\mu = 100$ : performance measures on ZDT1-4 and ZDT6 problems.

Test-function	Algorithm	Convergence measure			Hypervolume indicator		
		Average	Std. dev.	Rank	Average	Std. dev.	Rank
ZDT1	HIGA-MO	<b>0.00031201</b>	4.1269e-05	1	<b>120.64580412</b>	1.7718e-03	1
	Lara's direction	0.02103585	4.7314e-02	5	120.48926778	5.2474e-02	2
	Gap-filling	0.02091304	6.1387e-02	4	120.42616648	2.7937e-01	5
	NSGA-II	0.01769266	4.6048e-03	3	120.45030137	4.5135e-02	4
	SMS-EMOA	0.01234011	2.6377e-03	2	120.48071780	3.6130e-02	3
	SPEA2	0.06017346	1.7966e-02	6	119.86686583	2.1615e-01	6
ZDT2	HIGA-MO	0.00028335	3.3303e-05	3	<b>120.31710222</b>	2.3560e-03	1
	Lara's direction	<b>0.00005498</b>	1.2085e-05	1	120.30338190	2.9998e-03	2
	Gap-filling	0.00007857	8.7094e-05	2	120.14758158	1.5778e-01	3
	NSGA-II	0.02834448	4.4153e-03	5	119.16220851	1.0985e+00	4
	SMS-EMOA	0.02338094	7.0938e-03	4	118.40070248	2.7352e+00	5
	SPEA2	0.08566545	4.8472e-02	6	114.48551919	4.4285e+00	6
ZDT3	HIGA-MO	0.00047505	7.5997e-05	3	128.77154126	8.5828e-03	3
	Lara's direction	0.00046485	5.9553e-05	2	128.77257561	5.2596e-03	2
	Gap-filling	<b>0.00039660</b>	4.9392e-05	1	128.77099724	3.3611e-03	4
	NSGA-II	0.00063823	5.1880e-05	5	<b>128.77436195</b>	1.1318e-03	1
	SMS-EMOA	0.00055256	3.5594e-05	4	128.34841609	1.0889e+00	6
	SPEA2	0.00243258	6.6391e-03	6	128.55447469	7.9741e-01	5
ZDT4	HIGA-MO	31.34155544	3.9090e+00	4	0.00000000	0.0000e+00	6
	Lara's direction	40.35930710	1.1041e+01	5	0.00000000	0.0000e+00	5
	Gap-filling	43.47103886	1.5933e+01	6	5.23444012	1.5425e+01	4
	NSGA-II	<b>0.80498648</b>	5.0038e-01	1	<b>109.60569075</b>	5.4368e+00	1
	SMS-EMOA	1.01209147	6.3095e-01	2	107.14186469	7.1460e+00	2
	SPEA2	2.80155378	1.3959e+00	3	83.82023960	1.5461e+01	3
ZDT6	HIGA-MO	3.54689504	1.2985e+00	5	113.79978098	8.8488e-01	2
	Lara's direction	<b>0.00004369</b>	1.2553e-05	1	<b>116.49314419</b>	1.4990e+00	1
	Gap-filling	4.12388484	2.9230e+00	6	86.58598768	3.4123e+01	6
	NSGA-II	0.43202530	7.1773e-02	3	109.28079070	1.2513e+00	4
	SMS-EMOA	0.40028650	1.1394e-01	2	109.87049482	1.8951e+00	3
	SPEA2	0.49692387	1.2882e-01	4	108.17997611	1.9177e+00	5

2) a cumulative step-size adaptation mechanism is developed to approximate the optimal step-size in gradient ascent search. The algorithm is tested on 5 ZDT problems, and its performance is compared to evolutionary algorithms and some other gradient-based approaches. The proposed algorithm shows a fast convergence speed in terms of the hypervolume indicator.

As shown in the experimental results on ZDT4, the proposed algorithm fails to approach the global Pareto front and gets stuck in local ones instead. In practice, such an issue can be tackled by using restart heuristics to re-sample the stagnated points. In addition, it is possible to hybridize HIGA-MO with an evolutionary multi-objective (EMO) algorithm, where the global search ability of an EMO helps the algorithm to escape from a deceptive, local Pareto front and HIGA-MO could achieve fast convergence speed when approaching the global

Pareto front. Such an approach has been proposed in [9] and the optimal way to combine HIGA-MO with EMOs should be investigated.

The experiments conducted in this paper are on a small number of problems. In future research, the proposed algorithm should be investigated on more multi-objective problems. When using a large number of search points, the objective vectors on the Pareto front are close to each other, which might result in relatively slow movement. In this case, its performance needs to be further tested. In addition, it is of interest to compare HIGA-MO empirically to other set-based scalarization method [12].

For the proposed method for steering the dominated points, it should be also be empirically compared to alternative methods that are proposed in [17]. Those methods should be thoroughly compared to characterize their performance in terms of convergence measure and the hypervolume indicator value. In addition, as described in Section 5, the parameter tuning for the step-size adaptation is merely tested on a simple test problem (MPM2 function). A most rigorous parameter tuning procedure should be performed to get a reliable and robust parameter setting.

*Acknowledgments* This work presented in this paper is financially supported by the Dutch Research Project (NWO) PROMIMOOC (project number: 650.002.001).

## References

1. N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
2. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature, 2000 Conference on*, pages 849–858. Springer, Berlin, Heidelberg, 2000.
3. M. Emmerich and A. Deutz. Time complexity and zeros of the hypervolume indicator gradient field. In O. Schütze, C. A. Coello Coello, A.-A. Tantar, E. Tantar, P. Bouvry, P. D. Moral, and P. Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III*, volume 500 of *Studies in Computational Intelligence*, pages 169–193. Springer International Publishing, 2014.
4. M. Emmerich, A. Deutz, and N. Beume. Gradient-based/evolutionary relay hybrid for computing pareto front approximations maximizing the S-metric. In *Hybrid Metaheuristics, 2007 Conference on*, pages 140–156, Berlin, Heidelberg, 2007. Springer-Verlag.
5. J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
6. N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary computation*, 9(2):159–195, 2001.
7. C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.

8. P. Kerschke, H. Wang, M. Preuss, C. Grimme, A. Deutz, H. Trautmann, and M. Emmerich. *Towards Analyzing Multimodality of Continuous Multiobjective Landscapes*, pages 962–972. Springer International Publishing, Cham, 2016.
9. A. L. López, C. A. C. Coello, and O. Schütze. *Using Gradient Based Information to Build Hybrid Multi-objective Evolutionary Algorithms*. PhD thesis, CINVESTAV-IPN, Mexico City, May 2012.
10. J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2000.
11. Y. Ren, A. Deutz, and M. Emmerich. On steering dominated points in hypervolume gradient ascent for bicriteria continuous optimization (extended abstract). In *Numerical and Evolutionary Optimization, NEO 2015, Tijuana, Mexico (Book of abstracts)*, 2015.
12. O. Schütze, C. Domínguez-Medina, N. Cruz-Cortés, L. Gerardo de la Fraga, J.-Q. Sun, G. Toscano, and R. Landa. A scalar optimization approach for averaged hausdorff approximations of the pareto front. *Engineering Optimization*, 48(9):1593–1617, 2016.
13. O. Schütze, A. Lara, and C. Coello Coello. The directed search method for unconstrained multi-objective optimization problems. *Proceedings of the EVOLVE-A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation*, 2011.
14. V. A. Sosa Hernández, O. Schütze, and M. Emmerich. *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, chapter Hypervolume Maximization via Set Based Newton’s Method, pages 15–28. Springer International Publishing, Cham, Switzerland, 2014.
15. N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.
16. R. Storn and K. Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
17. H. Wang, Y. Ren, A. Deutz, and M. Emmerich. On steering dominated points in hypervolume indicator gradient ascent for bi-objective optimization. In O. Schütze, L. Trujillo, P. Legrand, and Y. Maldonado, editors, *NEO 2015: Results of the Numerical and Evolutionary Optimization Workshop NEO 2015 held at September 23-25 2015 in Tijuana, Mexico*, pages 175–203. Springer International Publishing, Cham, 2017.
18. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.
19. E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature, 2004 Conference on*, pages 832–842. Springer, 2004.
20. E. Zitzler, M. Laumanns, L. Thiele, et al. SPEA2: improving the strength pareto evolutionary algorithm. In *Eurogen*, volume 3242, pages 95–100, 2001.
21. E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms: a comparative case study. In *Parallel Problem Solving from Nature, 1998 Conference on*, pages 292–301. Springer, 1998.
22. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.