



# The Gradient Subspace Approximation and Its Application to Bi-objective Optimization Problems

Oliver Schütze<sup>1</sup>(✉), Lourdes Uribe<sup>2</sup>, and Adriana Lara<sup>2</sup>

<sup>1</sup> Computer Science Department, Cinvestav-IPN, Mexico City, Mexico  
schuetze@cs.cinvestav.mx

<sup>2</sup> ESFM del Instituto Politécnico Nacional, Mexico City, Mexico  
{lourdesur,adriana}@esfm.ipn.mx

**Abstract.** Evolutionary algorithms are very popular and are frequently applied to many different optimization problems. Reasons for this success include that methods of this kind are of global nature, very robust, and only require minimal assumptions on the optimization problem. It is also known that such methods need quite a few resources to generate accurate approximations of the solution sets. As a remedy, researchers have used hybrid (or memetic) algorithms, i.e., evolutionary algorithms coupled with local search for which mainly techniques from mathematical programming are utilized. Such hybrids typically yield satisfying results, the problem, however, remains that the algorithms are relatively expensive since the gradients have to be computed or approximated at each given candidate solution that is designated for local search.

In this chapter, we review the Gradient Subspace Approximation (GSA) which allows to compute a descent direction in a best fit manner from given neighborhood information that is e.g. already given in evolutionary algorithms. The computation of such directions comes hence for free in terms of additional function evaluations of the given problem which opens the door for the realization of low-cost local search engines within evolutionary algorithms. In a next step, we show how GSA can be applied to the context of bi-objective optimization. Finally, to demonstrate the benefit of the method we present some results on a hybrid that is based on the evolutionary algorithm NSGA-II.

**Keywords:** Gradient Subspace Approximation · Gradient free optimization · Bi-objective optimization · Descent directions

## 1 Introduction

In many problems in engineering and finance the problem arises that several objectives have to be optimized concurrently [3, 7, 8, 11, 12, 14, 17, 22, 27, 28, 30, 37]. One main challenge of such *multi-objective optimization problems* (MOPs)

is that their solution set—the so-called Pareto set respectively its image, the Pareto front—typically forms a  $(k - 1)$ -dimensional object, where  $k$  is the number of objectives involved in the problem. This is in contrast to classical scalar optimization problems (SOPs), where one expects that the optimum is taken at one single solution. Modern heuristics such as Multiobjective Evolutionary Algorithms (MOEAs, e.g., [2, 9, 10, 23]) are able to provide an approximation of the entire Pareto set/front of a given MOP in one single run of the algorithm due to their set oriented approach. These methods are very robust e.g. to initial conditions and only require minimal assumptions from the model (e.g., no derivative information). Evolutionary algorithms as well as other related heuristics are hence very popular for the numerical treatment of MOPs as well as other optimization problems. One drawback, however, that most of them suffer is that they need quite a few function evaluations in order to obtain accurate approximations of the Pareto sets/fronts. As a remedy, many researchers have in the past hybridized the (global) evolutionary algorithms with local search techniques mainly coming from Mathematical Programming that utilize derivative information from the objectives and the constraint functions. Such methods are termed hybrid evolutionary algorithms or memetic algorithms. While such methods yield in almost all cases satisfying results, they are still relatively expensive since the derivative information is required for every point that is designated for local search.

In this chapter, we review several recently developed tools that allow to realize a local search within a population based optimization algorithm with low computational cost. The basic idea of the Gradient Subspace Approximation (GSA, [33]) is to utilize existing neighborhood information to estimate the most greedy direction within the search space that is spanned by the samples. One advantage of this approach is that it can easily be extended to the context of constrained problems. The focus of this chapter is on bi-objective optimization problems (BOPs, i.e., MOPs with  $k = 2$  objective functions). For this, we will consider the descent direction for BOPs proposed in [25] and present recent adaptations for constrained problems ([38]). Next, we will show how GSA can be used to build a low-cost local search engine that can be used within a population based algorithm such as a MOEA. In order to show the efficiency of the method, we will show some numerical results from a hybrid evolutionary algorithm that coupled the GSA based local search engine with the famous evolutionary algorithm NSGSA-II ([10]) which is state-of-the-art for MOPs with two and three objectives.

The remainder of this chapter is organized as follows: in Sect. 2, we review the Gradient Subspace Approximation for the approximation of the most greedy search direction out of given neighborhood information. In Sect. 3, we review a particular descent direction for bi-objective optimization problems for unconstrained, equality and inequality constrained problems. We further combine these two concepts in order to build a low-cost local searcher within a chosen set-oriented optimization method (such as an evolutionary algorithm). In Sect. 4

we present some numerical examples on a hybrid evolutionary algorithm that is based on the famous NSGA-II. Finally, we draw our conclusions Sect. 5.

## 2 Gradient Subspace Approximation

In this section, we will review the Gradient Subspace Approximation that aims to compute a descent direction at a given point  $x_0$  and a given scalar optimization problem using existing neighborhood information. For more details the reader is referred to [33].

### 2.1 Background and Related Work

In this section, we will consider continuous scalar optimization problems (SOP) of the following form

$$\begin{aligned} \min_x & f(x) \\ \text{s.t. } & g_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_j(x) = 0, \quad j = 1, \dots, m. \end{aligned} \tag{1}$$

Hereby,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is called the objective function, and the functions  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$  are called the inequality and equality constraints, respectively. We assume that all functions  $f$ ,  $g_i$  and  $h_j$  are differentiable.

A point  $x$  is called feasible if it satisfies all constraints. A point  $x^*$  is called a solution to (1) if it is feasible and if there exists no other feasible point  $y$  that has a lower objective value.

The object of interest in this section is the gradient of a function at a given point  $x_0$ . Formally, the gradient of the objective at  $x_0$  is defined by

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T \in \mathbb{R}^n. \tag{2}$$

A vector  $\nu \in \mathbb{R}^n$  is called a *descent direction* for  $f$  at  $x_0$  if

$$\langle \nabla f(x_0), \nu \rangle < 0;$$

in that case, it holds for all sufficiently small step sizes  $t > 0$  that  $f(x_0 + t\nu) < f(x_0)$ .

When the gradient of a function is not available in analytic form, there are several ways to obtain either  $\nabla f(x_0)$  at a given point  $x_0$  or approximations of this vector. The most prominent and widely used technique is to use finite differences (e.g., [29]). The method presented in this section, GSA is also using a finite difference approach. The variance, however, is that GSA can gather the sampling points from all directions whereas the classical finite difference method utilizes samples in coordinate directions. GSA is hence more suited to the use within population based algorithms since in this case the neighboring

samples are already given (and typically not aligned in coordinate directions). In [6], a very similar method is proposed to approximate the Jacobians of the objective map of a given unconstrained multi-objective optimization problem. This work, however, does not discuss how to address constrained problems. The Hill Climber with Sidestep [26] and the Directed Search Method [35, 36] both use neighborhood samples in order to determine promising search directions for the local search within hybrid evolutionary algorithms. The difference to the GSA is that these works do not directly aim to approximate the gradient. Automatic Differentiation (AD, [18]) can be used to evaluate the exact gradient at a given point  $x_0$  if the function is specified by a computer program. One drawback of AD is that it can not be applied if this computer program is provided in form of binary code.

Further, there are several methods that replace the original objective by easier models. One example is response surface methodology (RSM), where the objective function  $f$  is replaced by low-order polynomials  $\tilde{f}$  (mainly of degree one and two) those gradients are approximated using least squares techniques [24]. If a first-order model is chosen, the match of the gradients  $\nabla\tilde{f}(x_0)$  and  $\nabla f(x_0)$  is typically quite good for a nonlinear function  $f$  if the chosen point  $x_0$  is sufficiently far away from the optimum. For second-order models, the match is in general much better, however, this accuracy comes with an additional cost since  $n^2$  parameters have to be fitted at every point  $x_0$ . Further works that can utilize scattered samples can be found in [13, 20]. In [20], a least squares regression is performed while in [13] statistical expectation is used. In both works, the authors restrict themselves to unconstrained problems.

## 2.2 The Basic Idea

The task is to compute a cost-free good approximation of the normalized gradient

$$n(x_0) := -\frac{\nabla f(x_0)}{\|\nabla f(x_0)\|_2}, \quad (3)$$

evaluated at a given point  $x_0$  within a particular subspace of the  $\mathbb{R}^n$ . For this, we make use of the fact that the gradient is the direction of the steepest ascent, and hence  $n(x_0)$  can be seen as the solution of the following optimization problem:

$$\begin{aligned} \min_{\nu \in \mathbb{R}^n} & \langle \nabla f(x_0), \nu \rangle \\ \text{s.t.} & \|\nu\|_2^2 = 1. \end{aligned} \quad (4)$$

To avoid to directly compute the gradient we can make use of neighboring information as follows: assume that we are given the points  $x_1, \dots, x_r$  in the vicinity of  $x_0$  (e.g., these samples may be contained within the population or archive of a set based optimization method such as an evolutionary algorithm) as well as their function values  $f(x_i)$ . In that case, we can use this information to approximate the directional derivatives in the directions

$$\nu_i := \frac{x_i - x_0}{\|x_i - x_0\|_2}, \quad i = 1, \dots, r. \quad (5)$$

Since it holds

$$f'_{\nu_i}(x_0) = \langle \nabla f(x_0), \nu_i \rangle = \frac{f(x_i) - f(x_0)}{\|x_i - x_0\|_2} + O(\|x_i - x_0\|_2), \tag{6}$$

where  $O$  denotes the Landau symbol, and since we are given the points  $x_0$  and  $x_i$  with their respective objective values  $f(x_0)$  and  $f(x_i)$ , we can hence use the approximations

$$f'_{\nu_i}(x_0) \approx \frac{f(x_i) - f(x_0)}{\|x_i - x_0\|_2}, \quad i = 1, \dots, r. \tag{7}$$

Given the samples  $x_1, \dots, x_r$  and  $\nu_i$  as above we define the subspace  $S$  as

$$S = \text{span}\{\nu_1, \dots, \nu_r\} \tag{8}$$

and are interested in a best approximation of  $n(x_0)$  within  $S$ . Since every  $\nu \in S$  can be written as

$$v = \sum_{i=1}^r \lambda_i \nu_i \tag{9}$$

for some  $\lambda = (\lambda_1, \dots, \lambda_r) \in \mathbb{R}^r$ , and

$$\langle \nabla f(x_0), \nu \rangle = \sum_{i=1}^r \lambda_i \langle \nabla f(x_0), \nu_i \rangle \tag{10}$$

we can state problem (4), where we restrict the search to  $S$ , as follows:

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^r} \quad & \sum_{i=1}^r \lambda_i \langle \nabla f(x_0), \nu_i \rangle \\ \text{s.t.} \quad & \left\| \sum_{i=1}^r \lambda_i \nu_i \right\|_2^2 = 1. \end{aligned} \tag{11}$$

Hence, when using an approximation of the directional derivatives as in (7) via using neighboring samples, we can avoid to directly compute the gradient. One advantage of using problem (11) is that constraint information can directly be incorporated.

In the following, we will analyze the best fit approximations of  $n(x_0)$  within the subspace  $S$  both for unconstrained and constrained SOPs. For this, we will first consider the ideal scenario where we assume that we are given all directional derivatives, and later on we will discuss the gradient-free realizations.

### 2.3 Gradient Subspace Approximation

We will in the following discuss how to approximate the most greedy search direction out of the given data, separately for unconstrained, equality constrained, and inequality constrained problems.

### 2.3.1 Unconstrained Problems

We are given the problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{12}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . Further, we are given a point  $x_0 \in \mathbb{R}^n$  and the samples  $x_1, \dots, x_r \in \mathbb{R}^n$  in the vicinity of  $x_0$ , together with their objective values  $f(x_i)$ . Further, for purpose of a better analysis of the problem we also assume that we are given the directional derivatives

$$\langle \nabla f(x_0), \nu_i \rangle, \quad i = 1, \dots, r. \tag{13}$$

Define the matrix  $V$  by

$$V = (\nu_1, \dots, \nu_r) \in \mathbb{R}^{n \times r}. \tag{14}$$

Then, the most greedy search direction within the subset

$$S = \text{span}\{\nu_1, \dots, \nu_r\} \tag{15}$$

is given by the solution of the following problem

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^r} \quad & \sum_{i=1}^r \lambda_i \langle \nabla f(x_0), \nu_i \rangle \\ \text{s.t.} \quad & \lambda^T V^T V \lambda - 1 = 0. \end{aligned} \tag{16}$$

The following result shows that the solution of (16) can be computed in closed form.

**Proposition 1.** *Let  $\nu_1, \dots, \nu_r \in \mathbb{R}^n$ ,  $r \leq n$ , be linearly independent and*

$$\tilde{\lambda}^* := -(V^T V)^{-1} V^T \nabla f(x_0). \tag{17}$$

Then

$$\lambda^* := \frac{\tilde{\lambda}^*}{\|V \lambda^*\|_2} \tag{18}$$

is the unique solution of (16) and

$$\nu^* = \frac{-1}{\|V \lambda^*\|_2} V (V^T V)^{-1} V^T \nabla f(x_0) \tag{19}$$

is the most greedy search direction in  $S$ .

*Proof.* The Karush-Kuhn Tucker (KKT) system of (16) reads as

$$\nabla_{\lambda} L(\lambda, \mu) = V^T \nabla f(x_0) + 2\mu V^T V \lambda = 0 \tag{20}$$

$$h(\lambda) = \lambda^T V^T V \lambda - 1 = 0. \tag{21}$$

Apparently, Equation (21) is only used for normalization. If we omit this equation and the factor  $2\mu$  in (20) we can rewrite (20) as the following normal equation system

$$V^T V \lambda = -V^T \nabla f(x_0). \tag{22}$$

To solve the entire KKT system we have to choose  $2\mu = \|V\lambda^*\|_2^2$ . Finally, the claim follows since the Hessian of the Lagrangian

$$\nabla_{\lambda\lambda}^2 L(\lambda, \mu) = V^T V \tag{23}$$

is positive definite since the directions  $\nu_i$  are linearly independent. □

Next, we discuss how to approximate the most greedy solution  $\nu^*$  without explicitly computing or approximating the gradients. Since

$$V^T \nabla(x_0) = \begin{pmatrix} \langle \nabla f(x_0), \nu_1 \rangle \\ \vdots \\ \langle \nabla f(x_0), \nu_r \rangle \end{pmatrix} \tag{24}$$

we can do the approximation as follows: let  $d = (d_1, \dots, d_r) \in \mathbb{R}^r$ , where

$$d_i := \frac{f(x_i) - f(x_0)}{\|x_i - x_0\|_2}, \quad i = 1, \dots, r, \tag{25}$$

and  $\tilde{\lambda}$  be the vector that solves the system of linear equations

$$V^T V \tilde{\lambda} = -d, \tag{26}$$

then the most greedy search direction can be approximated as

$$\tilde{\nu}^* = \frac{-1}{\|V\tilde{\lambda}\|_2^2} V(V^T V)^{-1} d. \tag{27}$$

*Remark 1.* (a) To compute  $\nu^*$  one has to solve system (22). It is hence advisable to avoid to choose directions  $\nu_i$  that nearly point into the same directions. The linear equation system yields the best condition number if the directions are chosen orthogonal to each other. In this case, we obtain

$$\nu^* = \frac{-1}{\|\lambda^*\|_2^2} V V^T \nabla f(x_0), \tag{28}$$

i.e., the orthogonal projection of  $\nabla f(x_0)$  onto  $S$ . That is,  $\nu^*$  is the best approximation of  $n(x_0)$  in  $S$ .

(b) In the special case that the coordinate directions are chosen, i.e., if we choose

$$x_i = x_0 + t_i e_{j_i}, \quad i = 1, \dots, r, \tag{29}$$

for the samples, where  $e_j$  denotes the  $j$ -th unit vector, we obtain for the  $j_i$ -th entry of  $\tilde{\nu}^*$  (without normalization)

$$\tilde{\nu}_{j_i}^* = \frac{f(x_0 + t_i e_{j_i}) - f(x_0)}{|t_i|}. \tag{30}$$

That is, if we for instance choose  $x_i = x_0 + t_i e_i$ ,  $i = 1, \dots, n$  (i.e., all coordinate directions), the search direction  $\nu^*$  coincides with the forward difference quotient.

- (c) The idea of GSA is to utilize existing data whenever possible. However, it may be the case that for a given point  $x_0$  that the existing data is not sufficient (e.g., not enough individuals of the current population are close enough to  $x_0$ ). A possible remedy may be to sample further points in order to compute a search direction. In that case it makes sense to choose the points so that the resulting directions  $\nu_i$  are orthogonal to each other as well as to all existing directions. See [1] for a possible realization.

*Example 1.* We consider the objective  $f : \mathbb{R}^6 \rightarrow \mathbb{R}$ , where

$$f(x) = \sum_{i=1}^6 x_i^2. \tag{31}$$

Let  $x_0 = [1, 1, 1, 1, 1, 1]^T$ , then  $\nabla f(x_0) = [2, 2, 2, 2, 2, 2]^T$  and

$$g = \frac{-1}{\sqrt{24}} [2, 2, 2, 2, 2, 2]^T \tag{32}$$

for which  $\langle \nabla f(x_0), g \rangle = -4.8990$ .

First, we choose three orthogonal search directions that form the matrix  $V^T$  as follows:

$$V^T = \begin{pmatrix} 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 1 \end{pmatrix}. \tag{33}$$

Doing so, we obtain  $\langle \nabla f(x_0), \frac{v_1}{\|v_1\|} \rangle = 2.6833$ ,  $\langle \nabla f(x_0), \frac{v_2}{\|v_2\|} \rangle = 2.5298$  and  $\langle \nabla f(x_0), \frac{v_3}{\|v_3\|} \rangle = 2.5298$ . If we solve problem (16) we get

$$v^* = [-0.5367, -0.1789, -0.2683, -0.5367, -0.5367, -0.1789]^T, \tag{34}$$

for which  $\langle \nabla f(x_0), v^* \rangle = -4.4721$ .

Next, we use the samples

$$x_i = x_0 + 0.1v_i, \text{ for } i = 1, 2, 3, \tag{35}$$

and via formula (27) we obtain

$$\tilde{v}^* = [-0.5237, -0.1813, -0.2618, -0.5438, -0.5438, -0.1813]^T$$

which leads to  $\langle \nabla f(x_0), \tilde{v}^* \rangle = -4.4714$ .

If choosing the non-orthogonal search directions

$$V^T = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 & 2 \end{pmatrix}, \tag{36}$$



we obtain  $\langle \nabla f(x_0), \frac{v_1}{\|v_1\|} \rangle = 2.8284$ ,  $\langle \nabla f(x_0), \frac{v_2}{\|v_2\|} \rangle = 3.3333$  and  $\langle \nabla f(x_0), \frac{v_3}{\|v_3\|} \rangle = 3.7947$ .

Solving (16) leads to

$$v^* = [-0.3769, -0.4744, -0.5686, -0.3054, -0.2080, -0.4159]^T \tag{37}$$

with  $\langle \nabla f(x_0), v^* \rangle = -4.6985$ . For the discretized problem we obtain

$$\tilde{v}^* = [-0.3595, -0.4674, -0.5770, -0.3171, -0.2092, -0.4184]^T \tag{38}$$

with  $\langle \nabla f(x_0), \tilde{v}^* \rangle = -4.6972$ .

### 2.3.2 Equality Constrained Problems

Next, we assume that the SOP contains some equality constraints, i.e., that we are given the following problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } h_i(x) = 0, \quad i = 1, \dots, p, \end{aligned} \tag{39}$$

where we assume that each  $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is differentiable.

Analogously to the unconstrained case discussed above the most greedy search direction at  $x_0$  in the entire space  $\mathbb{R}^n$  is given by

$$\begin{aligned} \min_{\nu \in \mathbb{R}^n} \langle \nabla f(x_0), \nu \rangle \\ \text{s.t. } \|\nu\|_2^2 = 1 \\ \langle \nabla h_i(x), \nu \rangle = 0, \quad i = 1, \dots, p, \end{aligned} \tag{40}$$

and the most greedy direction at  $x_0$  within the subspace  $S$  is given by

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^r} \sum_{i=1}^r \lambda_i \langle \nabla f(x_0), \nu_i \rangle \\ \text{s.t. } \lambda^T V^T V \lambda - 1 = 0 \\ \sum_{i=1}^r \lambda_i \langle \nabla h_j(x_0), \nu_i \rangle = 0, \quad j = 1, \dots, p. \end{aligned} \tag{41}$$

Denote the matrix  $H$  by

$$H = \begin{pmatrix} \nabla h_1(x_0)^T \\ \vdots \\ \nabla h_p(x_0)^T \end{pmatrix} \in \mathbb{R}^{p \times n}. \tag{42}$$

As for the unconstrained case, we can also express the most greedy solution for an equality constrained problem analytically.

**Proposition 2.** Let  $\nu_1, \dots, \nu_r \in \mathbb{R}^n$  be linearly independent where  $p \leq r \leq n$ , let  $\text{rank}(H) = p$ , and

$$\begin{pmatrix} \tilde{\lambda}^* \\ \tilde{\mu}^* \end{pmatrix} = \begin{pmatrix} V^T V & (HV)^T \\ HV & 0 \end{pmatrix}^{-1} \begin{pmatrix} -V^T \nabla f(x_0) \\ 0 \end{pmatrix}, \tag{43}$$

then

$$\lambda^* := \frac{\tilde{\lambda}^*}{\|V\lambda^*\|_2^2} \tag{44}$$

is the unique solution of (41) and thus

$$\nu^* = \frac{-1}{\|V\lambda^*\|_2^2} V(V^T V)^{-1} V^T \nabla f(x_0) \tag{45}$$

is the most greedy search direction in  $\text{span}\{\nu_i, \dots, \nu_r\}$ .

*Proof.* The KKT system of (41) is given by

$$V^T \nabla f(x_0) + 2\mu_0 V^T V \lambda + (HV)^T \mu = 0 \tag{46}$$

$$HV \lambda = 0 \tag{47}$$

$$\lambda^T V^T V \lambda - 1 = 0, \tag{48}$$

and via applying the same “normalization trick” as above we can transform the KKT equations into

$$\begin{pmatrix} V^T V & (HV)^T \\ HV & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} -V^T \nabla f(x_0) \\ 0 \end{pmatrix}. \tag{49}$$

To show that the matrix is regular, let  $y \in \mathbb{R}^r$  and  $z \in \mathbb{R}^p$  such that

$$\begin{pmatrix} V^T V & (HV)^T \\ HV & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = 0. \tag{50}$$

It follows that  $HV y = 0$  and hence that

$$0 = \begin{pmatrix} y \\ z \end{pmatrix}^T \begin{pmatrix} V^T V & (HV)^T \\ HV & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = y V^T V y. \tag{51}$$

Thus, it is  $y = 0$  since  $V^T V$  is positive definite. Further, by (50) it follows that  $V^T H^T z = 0$ . Since  $V^T \in \mathbb{R}^{n \times r}$  has rank  $r \geq p$ , it follows that  $V^T H^T$  has rank  $p$ . This implies that also  $z = 0$ , and thus, that the matrix in (43) is regular.

The rest follows by the discussion above setting  $2\mu_0 = \|\sum_{i=1}^r \tilde{\lambda}_i^* \nu_i\|_2^2$  and since the Hessian of the Lagrangian  $\nabla_{\lambda\lambda}^2 L(\lambda, \mu) = V^T V$  is positive definite.  $\square$

The key for a gradient-free approximation of the search direction is the matrix  $HV$ . Since

$$(HV)_{ij} = \nabla h_i(x_0)^T \nu_j,$$

we compute an approximation  $M = (m_{ij})$  of  $HV$  via

$$m_{ij} := \frac{h_i(x_j) - h_i(x_0)}{\|x_j - x_0\|_2}, \quad i = 1, \dots, p, \quad j = 1, \dots, r. \tag{52}$$

Doing so, we can now solve the system

$$\begin{pmatrix} V^T V & M^T \\ M & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} -d \\ 0 \end{pmatrix} \tag{53}$$

which leads to  $\lambda^*$ . To obtain  $\nu^*$  we proceed as for the unconstrained case using the approximation  $V^T \nabla f(x_0) \approx d$ .

*Example 2.* We consider the SOP from Example 1 and impose the constraint

$$h(x) = x_1 + x_2 + x_3 = 0. \tag{54}$$

For  $x_0 = [-1, 0, 1, 1, 1, 1]^T$  we have  $\nabla f(x_0) = [2, 2, 2, 2, 2, 2]^T$ ,  $g = \frac{-1}{\sqrt{24}} [2, 2, 2, 2, 2, 2]^T$ ,  $\nabla_h(x_0) = [1, 1, 1, 0, 0, 0]^T$  and  $\langle \nabla f(x_0), g \rangle = -4.8990$ .

First, we choose again the three orthogonal search directions

$$V^T = \begin{pmatrix} 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 1 \end{pmatrix}; \tag{55}$$

and, we obtain  $\langle \nabla f(x_0), \frac{v_1}{\|v_1\|} \rangle = 2.6833$ ,  $\langle \nabla f(x_0), \frac{v_2}{\|v_2\|} \rangle = 2.5298$ ,  $\langle \nabla f(x_0), \frac{v_3}{\|v_3\|} \rangle = 2.5298$ , and

$$v^* = [0.1210, -0.1815, 0.0605, -0.5444, -0.7662, -0.2554]^T \tag{56}$$

with  $\langle \nabla f(x_0), v^* \rangle = -3.1322$ .

Using the above setting and the samples

$$x_i = x_0 + 0.1v_i, \quad \text{for } i = 1, 2, 3 \tag{57}$$

we obtain the search direction

$$\tilde{v}^* = [0.1293, -0.1939, 0.0646, -0.5817, -0.7368, -0.2456]^T$$

which leads to  $\langle \nabla f(x_0), \tilde{v}^* \rangle = -3.1281$ .

In a next step, we consider the non-orthogonal search directions

$$V^T = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 & 2 \end{pmatrix} \tag{58}$$

leading to  $\langle \nabla f(x_0), \frac{v_1}{\|v_1\|} \rangle = 2.8284$ ,  $\langle \nabla f(x_0), \frac{v_2}{\|v_2\|} \rangle = 3.3333$  and  $\langle \nabla f(x_0), \frac{v_3}{\|v_3\|} \rangle = 3.7947$ .

When solving (41) we obtain

$$v^* = [0.4050, 0.2244, -0.6294, -0.3963, -0.2156, -0.4312]^T$$

with  $\langle \nabla f(x_0), v^* \rangle = -2.0863$  for the idealized problem and for the discretized problem we obtain

$$\tilde{v}^* = [0.4490, 0.1617, -0.6107, -0.4742, -0.1868, -0.3736]^T$$

with  $\langle \nabla f(x_0), \tilde{v}^* \rangle = -2.0691$ .

### 2.3.3 Inequality Constrained Problems

Finally, we assume we are given an inequality constrained SOP of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } g_i(x) \leq 0, \quad i = 1, \dots, m, \end{aligned} \tag{59}$$

where we for simplicity assume that all inequalities are active at a given point  $x_0$ . The most greedy search direction at  $x_0$  is given by

$$\begin{aligned} \min_{\nu \in \mathbb{R}^n} \langle \nabla f(x_0), \nu \rangle \\ \text{s.t. } \|\nu\|_2^2 = 1 \\ \langle \nabla g_i(x), \nu \rangle \leq 0, \quad i = 1, \dots, m, \end{aligned} \tag{60}$$

and the related subspace optimization problem reads as

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^r} \sum_{i=1}^r \lambda_i \langle \nabla f(x_0), \nu_i \rangle \\ \text{s.t. } \lambda^T V^T V \lambda - 1 = 0 \\ \sum_{i=1}^r \lambda_i \langle \nabla g_j(x_0), \nu_i \rangle \leq 0, \quad j = 1, \dots, m. \end{aligned} \tag{61}$$

One way to find a solution to (61) is to use gradient projection which is advantageous in particular if  $m$  is small and  $r \gg m$ . In the following, we first discuss the special case  $m = 1$  (i.e., one active inequality constraint) and will later on discuss the general case.

The classical gradient projection approach is to take the solution  $\nu^*$  of the underlying unconstrained problem (16) and to project it to the space  $\nabla g(x_0)^\perp$  that is orthogonal to  $\nabla g(x_0)$  (see Fig. 1): given a  $QR$  decomposition of  $\nabla g(x_0)$ , i.e.,

$$\nabla g(x_0) = QR = (q_1, \dots, q_n)R, \tag{62}$$

then the vectors  $q_2, \dots, q_n$  build an orthonormal basis (ONB) of  $\nabla g(x_0)^\perp$ . Using  $Q_g = (q_2, \dots, q_n)$ , the projection is hence given by

$$\nu_{new} = Q_g Q_g^T \nu^*. \tag{63}$$

It is of course not advisable to follow this approach directly since  $\nabla g(x_0)$  is neither given, nor do we want to approximate it. Alternatively, we propose to proceed as follows: let

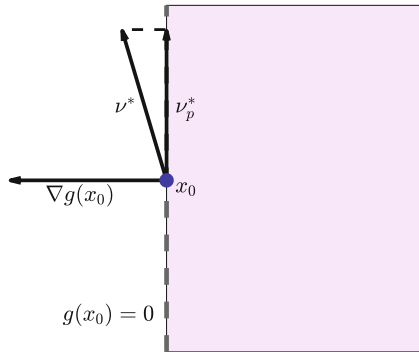
$$M := \nabla g(x_0)^T V = (\langle \nabla g(x_0), \nu_1 \rangle, \dots, \langle \nabla g(x_0), \nu_r \rangle) \in \mathbb{R}^{1 \times r}. \tag{64}$$

Note that if  $w$  is a kernel vector of  $M$  then  $Vw$  is perpendicular to  $\nabla g(x_0)$  and vice versa. Thus, we can compute the matrix

$$K = (k_1, \dots, k_{r-1}) \in \mathbb{R}^{r \times (r-1)} \tag{65}$$

those column vectors build an ONB of the kernel of  $M$ . If the search directions  $\nu_i$  are orthogonal, then also the vectors  $Vk_1, \dots, Vk_{r-1}$  are orthogonal to each other. The latter are the column vectors of  $VK \in \mathbb{R}^{n \times (r-1)}$  (if the  $\nu_i$ 's are not orthogonal to each other,  $VK$  has to be orthogonalized via another  $QR$  decomposition). Doing so, the projected vector to the kernel of  $M$  is given by

$$\tilde{\nu}_{new} = VK(VK)^T \nu^* = VKK^T V^T \nu^*. \tag{66}$$



**Fig. 1.** Handling inequality constraints using gradient projection.

For a general number  $m$  of inequality constraints we can extend the method as follows: let

$$M = GV = (\langle \nabla g_i(x_0), \nu_j \rangle)_{\substack{i=1, \dots, m \\ j=1, \dots, r}}, \tag{67}$$

and perform the following steps

- (1) compute an orthonormal basis  $K \in \mathbb{R}^{r \times (r-m)}$  of the kernel of  $M$
- (2) compute  $VK = QR = (q_1, \dots, q_{r-m}, \dots, q_n)R$  and set  $O := (q_1, \dots, q_{r-m}) \in \mathbb{R}^{n \times (r-m)}$
- (3)  $\tilde{v}_{new} = \tilde{Q}\tilde{Q}^T v^*$

*Example 3.* We consider again the SOP from Example 1 but this time we impose the inequality

$$g(x) = 1 - x_1 \leq 0. \tag{68}$$

For  $x_0 = [1, 1, 1, 1, 1, 1]^T$  we have  $\nabla f(x_0) = [2, 2, 2, 2, 2, 2]^T$ ,  $g = \frac{-1}{\sqrt{24}} [2, 2, 2, 2, 2, 2]^T$  and  $\nabla_c(x_0) = [-1, 0, 0, 0, 0, 0]^T$ . Thus  $\langle \nabla f(x_0), g \rangle = -4.8990$ .

First, we again choose the three orthogonal search directions

$$V^T = \begin{pmatrix} 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 1 \end{pmatrix}, \tag{69}$$

and obtain  $\langle \nabla f(x_0), \frac{v_1}{\|v_1\|} \rangle = 2.6833$ ,  $\langle \nabla f(x_0), \frac{v_2}{\|v_2\|} \rangle = 2.5298$  and  $\langle \nabla f(x_0), \frac{v_3}{\|v_3\|} \rangle = 2.5298$ .

For the search direction we obtain

$$v^* = [-0.5367, -0.1789, -0.2683, -0.5367, -0.5367, -0.1789]^T$$

with  $\langle \nabla f(x_0), v^* \rangle = -4.4721$ . Then by the gradient projection approach and  $v^*$ , we obtain the projected vector

$$v_{new} = [0, -0.1789, -0.2683, -0.5367, -0.5367, -0.1789]^T$$

with  $\langle \nabla f(x_0), v_{new} \rangle = -3.3988$ . Next, we obtain

$$\tilde{v}_{new} = [0, -0.1789, 0, -0.5367, -0.5367, -0.1789]^T$$

via Eq. (54) with  $\langle \nabla f(x_0), \tilde{v}_{new} \rangle = -2.8622$ .

Next we use the sampling

$$x_i = x_0 + 0.1v_i, \text{ for } i = 1, 2, 3 \tag{70}$$

which leads to the search direction

$$\tilde{v}^* = [-0.5237, -0.1813, -0.2618, -0.5438, -0.5438, -0.1813]^T$$

with  $\langle \nabla f(x_0), \tilde{v}^* \rangle = -4.4714$ . Then by the gradient projection approach and  $\tilde{v}^*$ , we obtain  $\tilde{v}_{new} = [0, -0.1813, 0, -0.5438, -0.5438, -0.1813]^T$  and  $\langle \nabla f(x_0), \tilde{v}_{new} \rangle = -2.9004$ .

### 3 Bi-objective Optimization

In this section, we will review a descent direction for bi-objective optimization problems, and will show how GSA can be used approximate these directions gradient-free. For details the reader is referred to [38].

#### 3.1 Background and Related Work

In many applications one is faced with the problem that several objectives have to be optimized concurrently leading to a multi-objective optimization problem (MOP, e.g., [3, 8, 12, 14, 17, 27, 28, 30, 37]).

A continuous MOP can be expressed mathematically as

$$\begin{aligned} \min_x & (f_1(x), \dots, f_k(x))^T \\ \text{s.t. } & g_i(x) \leq 0, \quad i = 1, \dots, p \\ & h_j(x) = 0, \quad j = 1, \dots, m, \end{aligned} \tag{71}$$

where the  $f_i, i = 1, \dots, k$  are the objectives to be minimized, and the  $g_i$ 's and  $h_j$ 's are the inequalities and equalities, respectively. Denote by  $Q$  the feasible set. We assume that all objectives and all constraint functions are differentiable. To define optimality of a MOP the concept of Pareto dominance is used: let  $v, w \in \mathbb{R}^k$ , then we say that the vector  $v$  is *less than* the vector  $w$  ( $v <_p w$ ), if  $v_i < w_i$  for all  $i \in \{1, \dots, k\}$ ; the relation  $\leq_p$  is defined analogously. A vector  $y \in Q$  is *dominated* by a vector  $x \in Q$  ( $x \prec y$ ) with respect to (71) if  $F(x) \leq_p F(y)$  and  $F(x) \neq F(y)$ , else  $y$  is called non-dominated by  $x$ . A point  $x^* \in \mathbb{R}^n$  is Pareto optimal to (71) if there is no  $y \in Q$  which dominates  $x$ . The set of all the Pareto optimal points is called the Pareto set and its image is the Pareto front. Both Pareto set and front form under certain (mild) smoothness assumptions a  $(k - 1)$ -dimensional object ([21]). We will in this section focus on bi-objective problems (BOPs), i.e., on MOPs where  $k = 2$ .

A Multi-objective Descent Direction (MODD)  $\nu$  at a point  $x_0$  and a given MOP is a direction in which a sufficiently small movement yields dominating solutions, i.e.,

$$x_0 + t\nu \prec x_0 \quad \forall t \in (0, \bar{t}) \tag{72}$$

for a certain  $\bar{t} > 0$ . In [25], a descent direction has been proposed for unconstrained BOPs.

**Proposition 3** ([25]). *Let  $x \in \mathbb{R}^n$  and  $f_1, f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$  define an unconstrained BOP. If  $\nabla f_i(x) \neq 0$  for  $i \in \{1, 2\}$  then*

$$\nu_L := - \left[ \frac{\nabla f_1(x)}{\|\nabla f_1(x)\|} + \frac{\nabla f_2(x)}{\|\nabla f_2(x)\|} \right], \tag{73}$$

*is a descent direction of  $x$  at  $F$ .*

In the sequel we will show how to adapt this to the context of constrained BOPs, and how to make a gradient-free realization via utilizing GSA.

It is worth mentioning that there exist some proposals to compute MODDs in general; they make use of first or second-order information related to the objective functions. Among these proposals is what is called the Steepest Descent Direction [15], which requires the solution to a quadratic programming problem involving the Jacobian of  $F$ . This method is valid for both convex or non-convex Pareto fronts. In [32], the authors introduced a mathematical formula that uses the solution of a stochastic differential equation related to the Karush-Kuhn-Tucker conditions in order to generate the solution set. This technique requires all the involved functions to be continuously differentiable, and it only works for unconstrained MOPs. One particular work to mention is [5] where appears the idea of the descent cone. The descent cone gets determined by the intersection of the negative half-spaces generated by the objective function gradients. In [4] the authors notice that computing a MODD yields again a multi-objective optimization problem since the negative gradient of every objective function is in conflict with the remaining objectives' gradients. In this same work, the authors propose a way to calculate the entire set of MODDs to take one of them at random finally. Another proposal [19], called the Pareto Descent Method, computes a set of possible Pareto descent directions by solving a linear programming problem with the information from the descent cones. This method applies just for unconstrained and inequality CMOP. There exist also approaches based on the Newton method as [14] where at each iteration of a line search, one minimization subproblem has to be solved to obtain the direction to follow. In this method, the Hessians of all functions need to be available. In this direction, also, [12, 16, 34] proposed an extension of the multi-objective Newton method for equality constrained problems. In this chapter, we decided to work with expression (73) since it is the easiest and no-cost proposal since no linear or quadratic programming solvers are necessary for its computation. This feature also makes it suitable for the application presented in Sect. 4 when this direction will be introduced into a population-based heuristic.

## 3.2 A Descent Direction for Constrained BOPs

### 3.3 Equality Constrained BOPs

Here we consider equality constrained BOPs of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} F(x) &:= [f_1(x), f_2(x)]^T, \\ \text{s.t.} \quad &h_j(x) = 0, \quad j = 1, \dots, m. \end{aligned} \quad (74)$$

We will in the following discuss separately the cases where  $x_0$  is feasible and infeasible.



**Feasible Case**

We first assume that  $x_0$  is feasible, i.e., that  $h_j(x_0) = 0$  for all  $j = 1, \dots, m$ . One way to obtain a MODD for the given BOP is to project the MODD of the related unconstrained BOP to the tangent space  $T_{h^{-1}(0)}(x_0)$  of the feasible set  $h^{-1}(0)$  at  $x_0$ : let  $\nu$  be the MODD for the unconstrained BOP, i.e.,

$$\nu := - \left[ \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} + \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right], \tag{75}$$

and let

$$H^T = (\nabla h_1(x_0) \dots, \nabla h_m(x_0)) = QR = (q_1, \dots, q_m, q_{m+1}, \dots, q_n) R \tag{76}$$

be a  $QR$ -factorization of  $H^T$ . Then, the vectors  $q_1, \dots, q_m$  build an orthonormal basis of the image of  $H^T$ . Further, since the image of  $H^T$  is the orthogonal complement of the kernel of  $H$ , we have for  $i = m + 1, \dots, n$ :

$$Hq_i = 0 \quad \Leftrightarrow \quad (\nabla h_j(x_0))^T q_i = 0, \quad j = 1, \dots, m). \tag{77}$$

That is, the column vectors of

$$\tilde{Q} := (q_{m+1}, \dots, q_n) \tag{78}$$

form an orthonormal basis of the tangent space  $T_{h^{-1}(0)}(x_0)$  of  $h^{-1}(0)$  at  $x_0$ . The orthogonal projection of  $\nu$  onto  $T_{h^{-1}(0)}(x_0)$  is hence given by (see also Algorithm 1)

$$\nu_p := \tilde{Q}\tilde{Q}^T\nu. \tag{79}$$

The following result establishes criteria under which  $\nu_p$  is a MODD.

---

**Algorithm 1.** Computation of the search direction  $\nu_p$  for equality constrained BOPs, feasible case

---

**Require:** BOP of form (),  $x_0$  with  $\nabla f_i(x_0) \neq 0, i = 1, 2$

**Ensure:** search direction  $\nu_p$

1:  $\nu := - \left[ \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} + \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right]$

2:  $H := \begin{pmatrix} \nabla h_1(x_0)^T \\ \vdots \\ \nabla h_m(x_0)^T \end{pmatrix}$

3: compute  $Q$  and  $R$  s.t.  $H^T = QR = (q_1, \dots, q_m, q_{m+1}, \dots, q_n)R$

4:  $\tilde{Q} := (q_{m+1}, \dots, q_n)$

5:  $\nu_p := \tilde{Q}\tilde{Q}^T\nu$

6: **return**  $\nu_p$

---

**Proposition 4.** *Let a BOP of the form (74) be given and  $x_0$  with  $\nabla f_i(x_0) \neq 0$  for  $i \in \{1, 2\}$  and  $h_j(x_0) = 0$ ,  $j = 1, \dots, m$ . Further, let  $\nu_p$  be given as in Eq. (79) such that  $\langle \nu_p, \nabla h_j(x_0) \rangle = 0$  for  $j \in \{1, \dots, m\}$ . Then the following holds:*

- (a) *If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) > 0$ , then  $\nu_p$  is a MODD of  $F$  at  $x_0$ .*
- (b) *If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) = 0$  and  $\tilde{Q}^T \nabla f_i(x_0) \neq 0$  for an index  $i \in \{1, 2\}$ , then  $\nu_p$  is a MODD of  $F$  at  $x_0$ .*
- (c) *If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) < 0$ , then  $\nu_p$  is not a descent direction of  $F$  at  $x_0$ .*

*Proof.* Note that for the first objective we obtained

$$\begin{aligned} \nabla f_1(x_0)^T \nu_p &= \nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nu_L \\ &= - \left[ \frac{\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} + \frac{\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right], \end{aligned} \tag{80}$$

and for the second objective

$$\begin{aligned} \nabla f_2(x_0)^T \nu_p &= \nabla f_2(x_0)^T \tilde{Q} \tilde{Q}^T \nu_L \\ &= - \left[ \frac{\nabla f_2(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} + \frac{\nabla f_2(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right]. \end{aligned} \tag{81}$$

Further, since  $0 < \|\nabla f_1(x_0)\|$  and  $0 < \|\nabla f_2(x_0)\|$ , and  $\tilde{Q} \tilde{Q}^T$  is symmetric we obtain

$$\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) = \nabla f_2(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_1(x_0) \tag{82}$$

and

$$\begin{aligned} \nabla f_i(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_i(x_0) &= \left\langle \tilde{Q}^T \nabla f_i(x_0), \tilde{Q}^T \nabla f_i(x_0) \right\rangle \\ &= \|\tilde{Q}^T \nabla f_i(x_0)\|^2 \geq 0 \text{ for } i \in \{1, 2\}. \end{aligned} \tag{83}$$

Then, three cases arise:

**Case 1.** If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) > 0$  holds, then by (80), (81) and (83) we obtain

$$\nabla f_1(x_0)^T \nu_p < 0 \text{ and } \nabla f_2(x_0)^T \nu_p < 0;$$

which means that  $\nu_p$  is a descent direction of  $F$  at  $x_0$ .

**Case 2.** If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) = 0$ , then

- (i) If  $\|\tilde{Q}^T \nabla f_i(x_0)\| > 0$  for  $i \in \{1, 2\}$ , then by (80), (81) and (83)  $\nabla f_i(x_0)^T \nu_p < 0$  for  $i \in \{1, 2\}$ , i.e.,  $\nu_p$  is a descent direction for  $F$  at  $x_0$ .
- (ii) If  $\|\tilde{Q}^T \nabla f_1(x_0)\| > 0$  and  $\|\tilde{Q}^T \nabla f_2(x_0)\| = 0$ , then  $\nabla f_1(x_0)^T \nu_p < 0$  and  $\nabla f_2(x_0)^T \nu_p = 0$ , i.e.,  $\nu_p$  is a descent direction for  $F$  at  $x_0$ .

(iii) If  $\|\tilde{Q}^T \nabla f_1(x_0)\| = 0$  and  $\|\tilde{Q} \nabla f_2(x_0)\| > 0$ , then  $\nabla f_1(x_0)^T \nu_p = 0$  and  $\nabla f_2(x_0)^T \nu_p < 0$ , i.e.,  $\nu_p$  is a descent direction for  $F$  at  $x_0$ .

Therefore, if  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) = 0$  and  $\tilde{Q}^T \nabla f_i(x_0) \neq 0$  for an index  $i \in \{1, 2\}$ , then  $\nu_p$  is a descent direction of  $F$  at  $x_0$ .

**Case 3.** If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) < 0$  assume, for the sake of contradiction, that  $\nu_p$  is a descent direction for  $F$  at  $x_0$ .

Then, if  $\nabla f_1(x_0)^T \nu_p < 0$  we have by Eq. (80) the following:

$$\begin{aligned}
 & \frac{\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} < \frac{\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \\
 \Leftrightarrow & \frac{\langle \tilde{Q}^T \nabla f_1(x_0), \tilde{Q}^T \nabla f_2(x_0) \rangle}{\|\nabla f_2(x_0)\|} < \frac{\langle \tilde{Q}^T \nabla f_1(x_0), \tilde{Q}^T \nabla f_1(x_0) \rangle}{\|\nabla f_1(x_0)\|} \\
 \Leftrightarrow & \frac{\|\tilde{Q}^T \nabla f_1(x_0)\| \|\tilde{Q}^T \nabla f_2(x_0)\| \cos \theta}{\|\nabla f_2(x_0)\|} < \frac{\|\tilde{Q}^T \nabla f_1(x_0)\|^2}{\|\nabla f_1(x_0)\|} \\
 \Leftrightarrow & \frac{\|\tilde{Q}^T \nabla f_2(x_0)\| \cos \theta}{\|\nabla f_2(x_0)\|} < \frac{\|\tilde{Q}^T \nabla f_1(x_0)\|}{\|\nabla f_1(x_0)\|} \\
 \Leftrightarrow & \left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\| \cos \theta < \left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\| \\
 \Leftrightarrow & \cos \theta < \frac{\left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\|}{\left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\|}. \tag{84}
 \end{aligned}$$

Analogously, if  $\nabla f_2(x_0)^T \nu_p < 0$  then by Eq. (81) we have

$$\cos \theta < \frac{\left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\|}{\left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\|}.$$

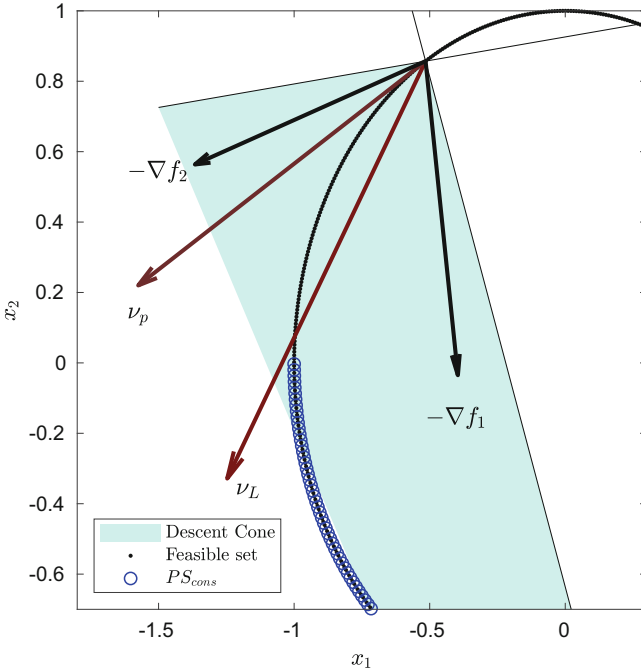
When considering  $-1 < \cos \theta < 1$ ,

$$\frac{\left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\|}{\left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\|} < 1 \quad \text{and} \quad \frac{\left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\|}{\left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\|} < 1$$

leads to

$$\left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\| < \left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\| \quad \text{and} \quad \left\| \tilde{Q}^T \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right\| < \left\| \tilde{Q}^T \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} \right\|,$$

which is not possible. Thus, we conclude that if  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) < 0$ , then  $\nu_p$  is not a descent direction of  $F$  at  $x_0$ , and we are done.  $\square$



**Fig. 2.** Considering a feasible starting point when computing  $\nu_p$  using Eq. (73). Here  $\nu_p$  is a MODD of BOP (85). This figure illustrates the case presented in Example 4.

*Example 4.* Consider the following BOP:

Minimize

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 + (x_2 + 3)^2 \\ f_2(x_1, x_2) &= (x_1 + 3)^2 + x_2^2 \end{aligned} \tag{85}$$

subject to

$$h(x_1, x_2) = x_1^2 + x_2^2 - 1 = 0.$$

For this example, consider the computation of direction  $\nu_p$  starting from a feasible initial point. Figure 2 illustrates the case when  $\nu_p$  lays over the descent cone, hence  $\nu_p$  is a descent direction.

**Infeasible Case**

Next, we consider that the initial point  $x_0$  is infeasible, i.e., that for at least one  $j \in \{1, \dots, m\}$  it holds  $h_j(x) \neq 0$ . Further, we assume that all equalities are linear, i.e., we are given the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} F(x) &:= [f_1(x), f_2(x)]^T, \\ \text{s.t. } h(x) &= Ax - b = 0, \end{aligned} \tag{86}$$

where  $A \in \mathbb{R}^{m \times n}$ . In this case we can consider the Newton method applied to the residual  $r : \mathbb{R}^{n+m+k} \rightarrow \mathbb{R}^{n+m+k}$  which is given by

$$r(x, \alpha, \nu) = \begin{pmatrix} J(x)^T \alpha + A^T \nu \\ Ax - b \\ \bar{e}^T \alpha - 1 \end{pmatrix}, \tag{87}$$

where  $J$  denotes the Jacobian of  $F$  and  $\bar{e} = [1, \dots, 1]^T \in \mathbb{R}^k$ . The first order Taylor approximation of  $r$  near an estimate  $y = (x, \alpha, \nu) \in \mathbb{R}^{n+k+m}$  is given by

$$r(y + z) \approx r(y) + Dr(y)z, \tag{88}$$

where  $Dr(y)$  is the Jacobian of  $r$  at  $y$ . The Newton step  $\Delta y$  for the Newton method applied to  $r$  solves the following linear system of equations:

$$Dr(y)\Delta y = -r(y). \tag{89}$$

Denote

$$W_\alpha := \sum_{j=1}^2 \alpha_j \nabla^2 f_j(x), \tag{90}$$

then

$$Dr(x, \alpha, \nu) = \begin{pmatrix} W_\alpha & J(x)^T & A^T \\ A & 0 & 0 \\ 0 & \bar{e}^T & 0 \end{pmatrix}, \tag{91}$$

and the Newton step is given by the vector

$$\Delta y = (\Delta x, \Delta \alpha, \Delta \nu)$$

that solves

$$\begin{pmatrix} W_\alpha & J(x)^T & A^T \\ A & 0 & 0 \\ 0 & \bar{e}^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \alpha \\ \Delta \nu \end{pmatrix} = - \begin{pmatrix} J(x)^T \alpha + A^T \nu \\ Ax - b \\ \bar{e}^T \alpha - 1 \end{pmatrix}. \tag{92}$$

If we set  $\nu^+ := \nu + \Delta \nu$ , we can rewrite the above system as

$$\begin{pmatrix} W_\alpha & J(x)^T & A^T \\ A & 0 & 0 \\ 0 & \bar{e}^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \alpha \\ \nu^+ \end{pmatrix} = - \begin{pmatrix} J(x)^T \alpha \\ Ax - b \\ \bar{e}^T \alpha - 1 \end{pmatrix}. \tag{93}$$

The following discussion shows that the norm of  $r$  decreases for sufficiently small step sizes in direction  $\Delta y$ : it is

$$\begin{aligned} \frac{d}{dt} \|r(y + t\Delta y)\|^2 \Big|_{t=0} &= -2r(y)^T Dr(y)\Delta y \\ &= -2r(y)^T r(y). \end{aligned} \tag{94}$$

Taking out the square leads to

$$\frac{d}{dt} \|r(y + t\Delta y)\| \Big|_{t=0} = -r(y)^T r(y) = -\|r(y)\| \tag{95}$$

which is negative at  $y$  with  $r(y) \neq 0$ .

In the following we summarize this result.

**Proposition 5.** *Let a BOP be of the form (86) and suppose  $x_0$  is given such that  $h_j(x_0) \neq 0$  for at least one  $j \in \{1, \dots, m\}$ . The Newton step on the residual  $r$  as defined in (87) is given by the vector that solves equation system (93), and  $\|r\|$  decreases for sufficiently small steps in direction of the Newton step.*

*Proof.* It follows by the above discussion. □

*Example 5.* Consider

$$F : \mathbb{R}^5 \rightarrow \mathbb{R}^2$$

subject to one linear equality constraint as

$$\begin{aligned} f_j(x) &= \sum_{i=1, i \neq j}^5 (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \quad j = 1, 2 \\ \text{s.t. } \frac{1}{2}x_1 &= x_2. \end{aligned} \tag{96}$$

Here,  $a^1 = [1, \dots, 1]^T \in \mathbb{R}^5$  and  $a^2 = -a^1$ . We apply Newton’s method for the initial infeasible point  $p_0 = a^1$  with  $h(p_0) = -0.5$ . Figure 3 shows the obtained solutions in each Newton step in the (a) variable and (b) objective space. In the fourth step we obtain the final solution

$$p_4 = [0.2668, 0.1334, 0.3750, 0.3750, 0.3750]^T$$

with  $h(p_4) = -2.7756e-17$ , which can be considered to be feasible.

### 3.4 Inequality Constrained BOPs

Next we consider inequality constrained BOPs of the form

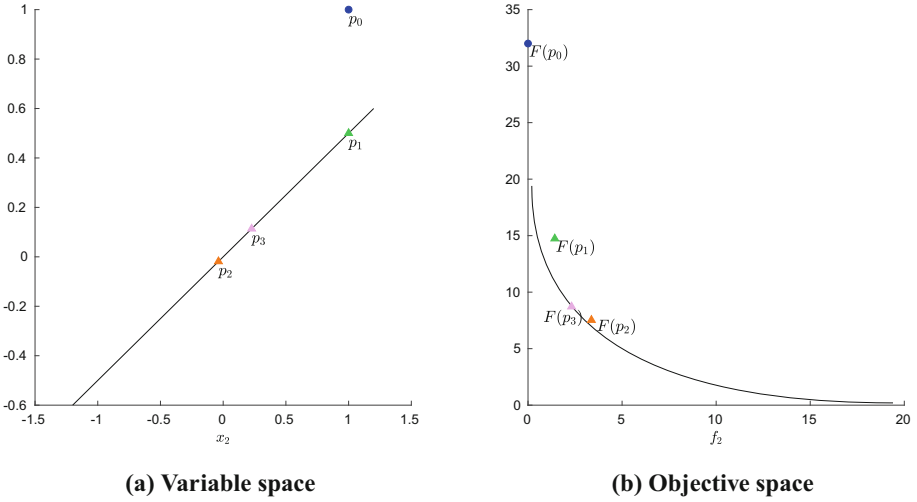
$$\begin{aligned} \min_{x \in \mathbb{R}^n} F(x) &:= [f_1(x), f_2(x)]^T, \\ \text{s.t. } g_j(x) &\leq 0, \quad j = 1, \dots, p. \end{aligned} \tag{97}$$

Let  $x_0$  be given and denote by

$$I(x_0) := \{g_{i1}(x_0), g_{i2}(x_0), \dots, g_{is}(x_0)\} \tag{98}$$

the set of active inequalities at  $x_0$ . Assume that  $I(x_0)$  is not empty, i.e., that  $s \geq 1$ . Denote by

$$G := \begin{pmatrix} \nabla g_{i1}(x_0)^T \\ \vdots \\ \nabla g_{is}(x_0)^T \end{pmatrix} \in \mathbb{R}^{s \times n} \tag{99}$$



**Fig. 3.** Newton steps starting from  $p_0$  which is an infeasible point for Problem (96). This figure illustrates Example 5.

the matrix formed by the gradients of the active inequality constraints.

Similarly to the feasible case for equality constrained BOPs, we can also in this case generate descent directions via projection as follows: suppose that  $rank(G) = s$  (i.e., maximal), then we can compute the factorization

$$G^T = QR = (q_1, \dots, q_s, q_{s+1}, \dots, q_n) R, \tag{100}$$

where  $Q \in \mathbb{R}^{n \times n}$  is orthogonal and  $R \in \mathbb{R}^{n \times s}$  right upper triangular. Doing so, the last  $n - s$  column vectors of  $Q$  form an orthonormal basis of the tangent space of the feasible set  $g_i^{-1}(0)$  at  $x_0$  with  $\nabla f_i(x_0) \neq 0$  and  $g_i(x_0) = 0$ . The orthogonal projection  $\nu^p$  onto  $T_{g_i^{-1}(0)}(x)$  is hence given by

$$\nu_p := \tilde{Q}\tilde{Q}^T \nu_L, \tag{101}$$

where  $\nu_L$  is as in (73) and

$$\tilde{Q} := (q_{s+1}, \dots, q_n). \tag{102}$$

Similarly to the equality constrained case,  $\nu_p$  is a descent direction under certain conditions (see also Algorithm 2).

**Proposition 6.** For a BOP of the form (97), suppose  $\nabla f_i(x_0) \neq 0$  for  $i \in \{1, 2\}$ . Let  $\nu_p$  be as in Eq. (101) such that  $\langle \nu_p, \nabla g_i(x_0) \rangle = 0$ . Let  $x \in \mathbb{R}^n$  such that  $g_i(x_0) = 0$  for every  $i \in I(x_0)$ . Then

(a) If  $\nabla f_1(x_0)^T \tilde{Q}\tilde{Q}^T \nabla f_2(x_0) > 0$ , then  $\nu_p$  is a MOPP of  $F$  at  $x_0$ .

- (b) If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) = 0$  and  $\tilde{Q}^T \nabla f_i(x_0) \neq 0$  for an index  $i \in \{1, 2\}$ , then  $\nu_p$  is a MODD of  $F$  at  $x_0$ .
- (c) If  $\nabla f_1(x_0)^T \tilde{Q} \tilde{Q}^T \nabla f_2(x_0) < 0$ , then  $\nu_p$  is not a descent direction of  $F$  at  $x_0$ .

*Proof.* Note that by construction  $\nabla g_i(x_0)^T \nu_p = 0$  for all active  $i \in I(x_0)$ ; thus, the proof is analog to the one from Proposition 4. □

---

**Algorithm 2.** Computation of the search direction  $\nu_p$  for inequality constrained BOPs

---

**Require:** BOP of form (97),  $x_0$  with  $\nabla f_i(x_0) \neq 0, i = 1, 2$

**Ensure:** search direction  $\nu_p$

- 1:  $\nu := - \left[ \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} + \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right]$
  - 2:  $G := \begin{pmatrix} \nabla g_{i1}(x_0)^T \\ \vdots \\ \nabla g_{is}(x_0)^T \end{pmatrix} \in \mathbb{R}^{s \times n}$
  - 3: compute  $Q$  and  $R$  s.t.  $G^T = QR = (q_1, \dots, q_s, q_{s+1}, \dots, q_n)R$
  - 4:  $\tilde{Q} := (q_{s+1}, \dots, q_n)$
  - 5:  $\nu_p := \tilde{Q} \tilde{Q}^T \nu$
  - 6: **return**  $\nu_p$
- 

*Example 6.* Consider the following BOP proposed in [8]:

Minimize

$$\begin{aligned}
 f_1(x) &= x_1 & (103) \\
 f_2(x) &= g(x) * \left( 1 - \sqrt{\frac{x_1}{g(x)}} \right), \\
 \text{with } g(x) &= 1 + x_2.
 \end{aligned}$$

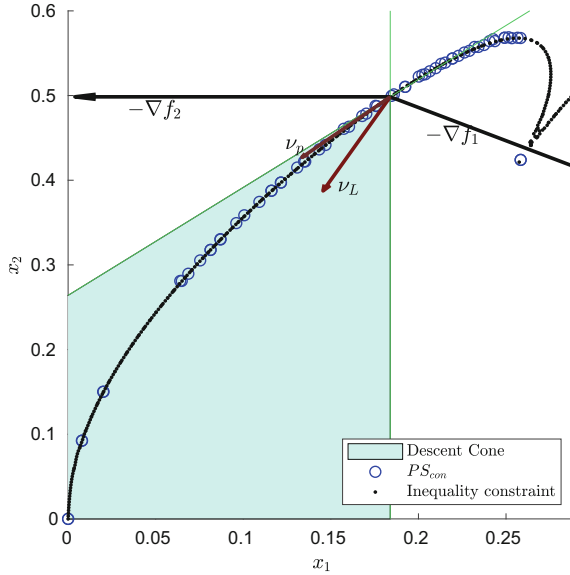
Subject to

$$\cos \theta (f_2(x_1, x_2) - e) - \sin \theta f_1(x_1, x_2) \geq \tag{104}$$

$$a |\sin (b\pi (\sin \theta (f_2(x_1, x_2) - e) + \cos \theta f_1(x_1, x_2))^c)|^d, \tag{105}$$

where  $0 \leq x_i \leq 1$  for  $i \in \{1, 2\}$ . And  $a = 0.1, b = 10, c = 2, d = 0.5, e = 1$  and  $\theta = -0.2 * \pi$ . Fig. 4 shows an example of the proposed MODD. Note that  $\nu_p$  lays over the descent cone and fulfills the above criterion; thus  $\nu_p$  is a descent direction.





**Fig. 4.** Considering a starting point for an active constraint when computing  $\nu_p$  using Eq. (73), it can be a descent direction or not. This figure illustrates the behavior in decision variable space for Example 6.

### 3.5 A Gradient Free Approximation of a MODD for CBOPs

In the following we use the GSA method to compute the above discussed descent directions for (unconstrained or constrained) bi-objective optimization problems.

We assume that we are given a candidate solution  $x_0$  that is designated for local search, and that we are given sample points  $x_1, x_2, \dots, x_r$  in the vicinity of  $x_0$ . We further assume that their objective functions values  $f(x_i)$  are already known, which is indeed the case if the  $x_i$ 's are chosen from a given population within a MOEA. Recall from GSA that we can use

$$\nu_i := \frac{x_i - x_0}{\|x_i - x_0\|_2}, \quad d_i := \frac{f(x_i) - f(x_0)}{\|x_i - x_0\|_2}, \quad i \in \{1, \dots, r\}, \quad (106)$$

Thus, (16) turns into

$$\begin{aligned} \min \quad & \lambda^T d \\ \text{s.t.} \quad & \lambda^T V^T V \lambda - 1 = 0, \end{aligned} \quad (107)$$

which is the problem we have to solve. Then, consider the matrix

$$\tilde{V} := (\nu_1, \dots, \nu_r), \quad (108)$$

to finally obtain

$$\tilde{\nu}^* = - \frac{1}{\|\tilde{V} \tilde{\lambda}^*\|} \tilde{V} (\tilde{V}^T \tilde{V})^{-1} d. \quad (109)$$

Then,  $\tilde{\nu}^*$  is the most greedy direction for a single-objective. It is worth to notice that this derivation can be done for all objective functions in a MOP.

**Unconstrained Case**

For the unconstrained case, we can apply the previous idea to approximate Eq. (73), hence obtaining a gradient-free descent direction. Recall that

$$\nu_L := - \left[ \frac{\nabla f_1(x)}{\|\nabla f_1(x)\|} + \frac{\nabla f_2(x)}{\|\nabla f_2(x)\|} \right], \tag{110}$$

then we can approximate  $\nabla f_1(x)$  and  $\nabla f_2(x)$  as follows:

$$\tilde{\nu}_j^* = - \frac{1}{\|\tilde{V}\tilde{\lambda}^*\|} \tilde{V}(\tilde{V}^T\tilde{V})^{-1}d^j, \tag{111}$$

for  $j = \{1, 2\}$ . Then, we can approximate  $\nu_L$  as

$$\tilde{\nu}_L := - \left[ \frac{\tilde{\nu}_1^*}{\|\tilde{\nu}_1^*\|} + \frac{\tilde{\nu}_2^*}{\|\tilde{\nu}_2^*\|} \right]. \tag{112}$$

**Equality Constrained Case**

For this scenario assume that  $x_0$  is a feasible solution, and that we are given  $x_1, x_2, \dots, x_r$  which are sample points in the neighborhood of  $x_0$ ; also, that their objective functions values  $f(x_i)$  are already known. From the constrained case of GSA recall that the Matrix  $M := (m_{ji}) \in \mathbb{R}^{m \times r}$  is given by

$$m_{j,i} := \frac{h_j(x_i) - h_j(x_0)}{\|x_i - x_0\|_2}, \quad i \in \{1, \dots, r\}, \quad j \in \{1, \dots, m\}. \tag{113}$$

Via Eqs. (108) and (113) we can compute an approximation  $\tilde{H}^T$  of the Jacobian matrix  $H^T$  of the constraint functions given by

$$\tilde{H}^T = \tilde{V}(\tilde{V}^T\tilde{V})^{-1}M^T \tag{114}$$

that will be used to compute the projection of  $\nu_p$ . Then, considering a feasible solution  $x_0$ , we proceed analogously to the unconstrained case, and first compute  $\tilde{\nu}_L$  as in Eq. (112). Next, in order to compute  $\tilde{\nu}_p$ , we compute a QR decomposition of  $\tilde{H}^T$ , and define

$$\tilde{Q} := (\tilde{q}_{m+1}, \dots, \tilde{q}_n),$$

where  $\tilde{q}_i, i \in \{m+1, \dots, n\}$ , are the last  $n - m$  column vectors of the orthogonal matrix  $Q$  obtained by the QR-decomposition of  $\tilde{H}^T$ . Then

$$\tilde{\nu}_p := \tilde{Q}\tilde{Q}^T\tilde{\nu}_L, \tag{115}$$

is the orthogonal projection of  $\tilde{\nu}_p$  onto the set of feasible directions. Assuming the notation for  $\tilde{V}$  and  $d_i^j$  as in Eq. (108) and (106), the following criteria manages the application of our gradient-free proposal:

For a BOP of the form (74), with  $\nabla f_i(x) \neq 0$  for  $i \in \{1, 2\}$  and  $h_j(x) = 0$ . Compute  $\tilde{\nu}_p$ , as in Eq. (115) and

$$C_g := d^{1T}(\tilde{V}^T \tilde{V})^{-1} \tilde{V}^T \tilde{Q} \tilde{Q}^T \tilde{V} (\tilde{V}^T \tilde{V})^{-1} d^2. \quad (116)$$

Then we proceed as follows:

1. If  $C_g > 0$ , then perform a line search over  $\tilde{\nu}_p$ .
2. If  $C_g = 0$  and  $d^{iT}(\tilde{V}^T \tilde{V})^{-1} \tilde{V}^T \tilde{Q} \tilde{Q}^T \tilde{V} (\tilde{V}^T \tilde{V})^{-1} d^i \neq 0$  for an index  $i \in \{1, 2\}$ , then perform a line search over  $\tilde{\nu}_p$ .
3. If  $C_g < 0$ , then the line search is not applied.

Note that the above criteria allow us to decide, during the algorithm's running time, when the information available is likely enough to have an approximation of a MODD. After deciding to approximate such direction, we compute the new iterative point  $x_i$  as follows:

$$x_i := x_0 + t\tilde{\nu}_p, \quad (117)$$

where  $t$  is a suitable step length. In this work, we computed  $t$  by a backtracking procedure based on the Armijo's condition [29]. The description in Algorithm 3 corresponds to the standalone gradient-free algorithm for equality constrained MOPs.

### Inequality Constrained Case

In the case that inequality constraints are present, the consideration is made over  $I(x)$ , that is the set of active inequality constraints at  $x$ . Thus we obtain the new approximation of  $\tilde{\nu}_L^p$  as follows:

$$\tilde{\nu}_L^p := \tilde{Q} \tilde{Q}^T \tilde{\nu}_L. \quad (118)$$

Assuming the notation for  $\tilde{V}$  and  $d_i^j$  as in Eq. (108) and (113), the following result states the criteria for the application of the gradient-free proposal:

For a BOP of the form (97), with  $\nabla f_i(x) \neq 0$  for  $i \in \{1, 2\}$  and  $g_i(x) = 0$  for every  $i \in I(x)$ . Compute  $\tilde{\nu}_L^p$  as in Eq. (118) and

$$C_g := d^{1T}(\tilde{V}^T \tilde{V})^{-1} \tilde{V}^T \tilde{Q} \tilde{Q}^T \tilde{V} (\tilde{V}^T \tilde{V})^{-1} d^2. \quad (119)$$

Then we proceed as follows:

1. If  $C_g > 0$ , then perform a line search over  $\tilde{\nu}_p$ .
2. If  $C_g = 0$  and  $d^{iT}(\tilde{V}^T \tilde{V})^{-1} \tilde{V}^T \tilde{Q} \tilde{Q}^T \tilde{V} (\tilde{V}^T \tilde{V})^{-1} d^i \neq 0$  for an index  $i \in \{1, 2\}$ , then perform a line search over  $\tilde{\nu}_p$ .
3. If  $C_g < 0$ , then the line search is not applied.

**Algorithm 3.** Standalone gradient-free MODD for equality constrained BOPs.**Require:**  $x_0$  :initial solution,  $r$  :number of neighbors,  $\varepsilon$  : threshold for  $C_g$ **Ensure:**  $x_f$  :final solution.

---

```

1: while Stopping Criterion does not fulfill do
2:    $i \leftarrow 1$ ;
3:    $x_i \leftarrow x_0$ ;
4:   Compute  $x_1, \dots, x_r$  neighbor points for  $x_0$ ;
5:   Compute  $\nu_i$  and  $d$  as in Eq. (113);
6:   Compute  $\tilde{V}$  as in Eq. (108);
7:   if  $x_i$  is a feasible solution then
8:     Compute  $C_g$  as in Eq. (116).
9:      $c_j \leftarrow d^{jT} (\tilde{V}^T \tilde{V})^{-1} \tilde{V}^T \tilde{Q} \tilde{Q}^T \tilde{V} (\tilde{V}^T \tilde{V})^{-1} d^j \quad j = 1, 2$ ;
10:    if  $C_g > 0$  then
11:      Compute  $\tilde{\nu}_p$  as in Eq. (115);
12:      Compute  $t \in \mathbb{R}^+$ ;
13:       $x_{i+1} \leftarrow x_i + t\tilde{\nu}_L$ ;
14:       $i \leftarrow i + 1$ ;
15:    else if  $|C_g| < \varepsilon$  then
16:      if  $c_1 > 0$  or  $c_2 > 0$  then
17:        Compute  $\tilde{\nu}_p$  as in Eq. (115);
18:        Compute  $t \in \mathbb{R}^+$ , a suitable step size
19:         $x_{i+1} \leftarrow x_i + t\tilde{\nu}_L$ ;
20:         $i \leftarrow i + 1$ ;
21:      else
22:        It is not a descent direction.
23:      end if
24:    else
25:      It is not a descent direction.
26:    end if
27:  end if
28: end while
29:
30: return  $x_f \leftarrow x_i$ ;

```

---

After deciding to approximate such direction, we compute the new iterative point  $x_i$  as follows:

$$x_i := x_0 + t\tilde{\nu}_L^p, \quad (120)$$

where  $t$  is a suitable step length. In this work, we compute  $t$  via a backtracking procedure based again on Armijo's condition. The standalone algorithm for inequality CMOPs is described in Algorithm 4.

We have presented in this section some criteria to decide when a solution is a MODD. The potential of these criteria will be displayed in the next section when used in combination with population-based heuristics.

---

**Algorithm 4.** Standalone gradient-free MODD for inequality CBOPs.

---

**Require:**  $x_0$  :initial solution,  $r$  :number of neighbors,  $\varepsilon$  : threshold for  $C_g$ ,  $I(x)$  :active set.

**Ensure:**  $x_f$  :final solution.

```

1: while Stopping Criterion does not fulfill do
2:    $i \leftarrow 1$ ;
3:    $x_i \leftarrow x_0$ ; Compute  $x_1, \dots, x_r$  neighbor points for  $x_0$ ;
4:   Compute  $\nu_i$  and  $d$  as in Eq. (113);
5:   Compute  $\tilde{V}$  as in Eq. (108);
6:   Compute  $C_g$  as in Eq. (119);
7:    $c_j \leftarrow d^{jT}(\tilde{V}^T\tilde{V})^{-1}\tilde{V}^T\tilde{Q}\tilde{Q}^T\tilde{V}(\tilde{V}^T\tilde{V})^{-1}d^j \quad j = 1, 2$ ;
8:   if  $C_g > 0$  then
9:     Compute  $\tilde{\nu}_p$  as in Eq. (115);
10:    Compute  $t \in \mathbb{R}^+$ , a suitable step size
11:     $x_{i+i} \leftarrow x_i + t\tilde{\nu}_L$ ;
12:     $i \leftarrow i + 1$ ;
13:   else if  $|C_g| < \varepsilon$  then
14:     if  $c_1 > 0$  or  $c_2 > 0$  then
15:       Compute  $\tilde{\nu}_p$  as in Eq. (115);
16:       Compute  $t \in \mathbb{R}^+$ ;
17:        $x_{i+1} \leftarrow x_i + t\tilde{\nu}_L$ ;
18:        $i \leftarrow i + 1$ ;
19:     else
20:       It is not a descent direction;
21:     end if
22:   else
23:     It is not a descent direction;
24:   end if
25: end while
26:
27: return  $x_f \leftarrow x_i$ ;

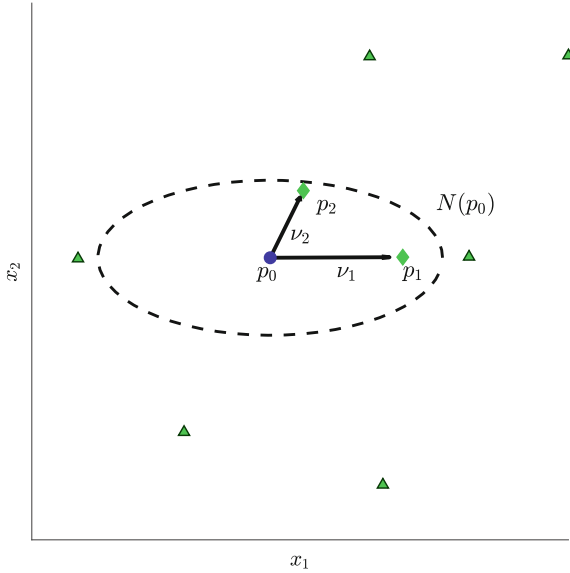
```

---

## 4 Application: Use of GFDD Within NSGA-II

In order to apply the developed ideas, we will in the following show some examples of the integration the above ideas to perform a multi-objective local search within the execution of the well-known algorithm NSGA-II [10] as demonstrator. This is a state-of-the-art algorithm for bi- and three-objective optimization problems that makes use of an archiving strategy based on the *crowding distance*. This strategy will play an important role in the hybridization as it will help us to decide which individual is a suitable starting point to perform the local search.

Algorithm 5 shows a pseudo code of a hybrid of GFDD and NSGA-II which also shows that such a coupling can be done relatively easily with in principle



**Fig. 5.** Points  $p_1$  and  $p_2$  are considered to be neighbors of  $p_0$ . The green triangles outside the circle are population elements which are not used for the GSA computation.

---

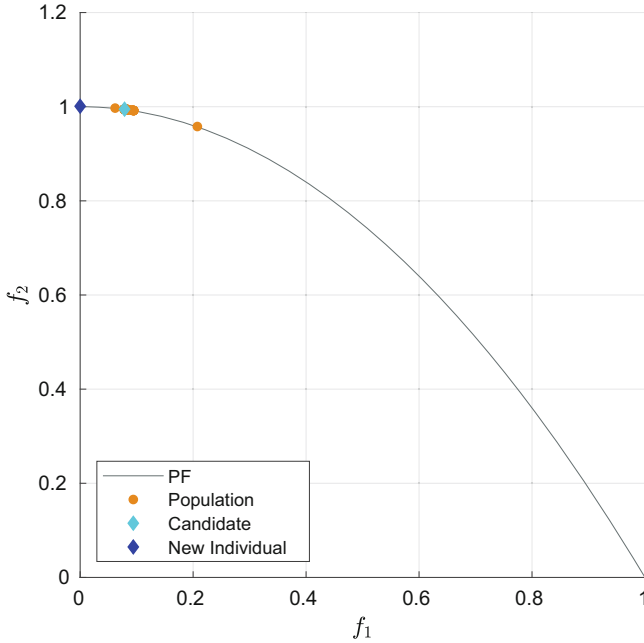
**Algorithm 5.** Pseudocode of *NSGA – II + Localsearch*

---

**Require:**  $P_s$  := Population size,  $size$  := Problem Size,  $P_c$  := crossover probability,  $P_m$  := mutation probability

**Ensure:** Set of approximated solutions

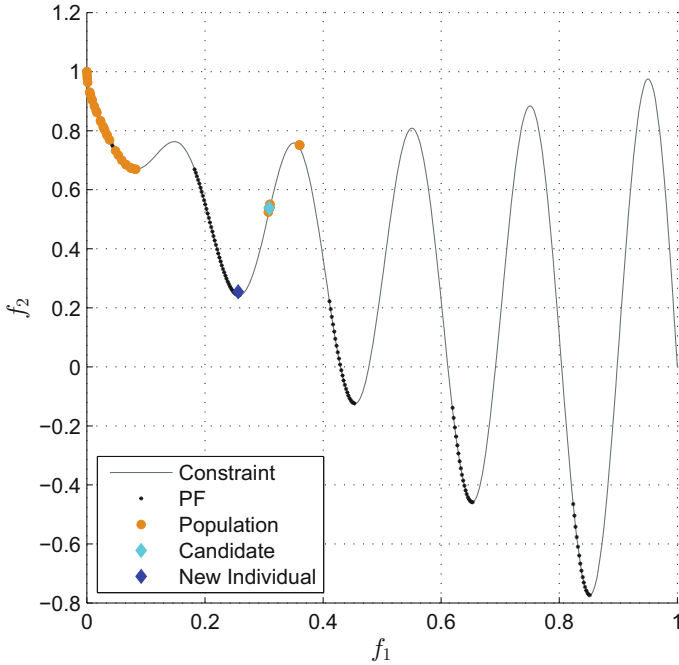
- 1: Population  $\leftarrow$  InitializePopulation( $P_s$ ,  $size$ )
  - 2: FitnessEvaluation(Population)
  - 3: FastNondominatedSort(Population)
  - 4: Selected  $\leftarrow$  SelectParentsByRank(Population,  $P_s$ )
  - 5: Offspring  $\leftarrow$  CrossoverAndMutation(Selected,  $P_c$ ,  $P_m$ )
  - 6: **while**  $\neg$ StopCondition() **do**
  - 7:   FitnessEvaluation(Offspring)
  - 8:   Union  $\leftarrow$  Merge(Population, Offspring)
  - 9:   Fronts  $\leftarrow$  FastNondominatedSort(Union)
  - 10:   CrowdingDistanceAssignment(Fronts)
  - 11:   Selected  $\leftarrow$  SelectParentsByRankAndDistance(Population,  $P_s$ )
  - 12:   Population  $\leftarrow$  Offspring
  - 13:   Selected individual  $x_s$
  - 14:   ApplyLocalSearch( $x_s$ ) if suitable
  - 15:   Offspring  $\leftarrow$  CrossoverAndMutation(Selected,  $P_c$ ,  $P_m$ )
  - 16: **end while**
  - 17:
  - 18: **return** Offspring
-



**Fig. 6.** This figure illustrates a particular instant of a certain iteration in the population-based algorithm. The population is represented by circles, while the initial and final solutions involved in the local search procedure are marked by light and dark diamonds correspondingly. The tested function corresponds to the CZDT2 from [31].

any other MOEA. The first part of the algorithm (lines 1 to 12) coincides with the evolutionary process of the NSGA-II. Then, the interleaving of our proposal starts; in lines 13 and 14 we select an individual  $x_s$  related to the biggest crowding value. If  $x_s$  is feasible we decide based on the propositions presented above if the local search is suitable. For the case of inequality constraints we just consider the set of active constraints. Once the proposed low-cost MODD is successfully computed, we apply a regular line search through it with a suitable step size control provided with a traditional backtracking tuning.

At each generation, we applied the local search only to one selected individual mainly because we do not want to make big changes through the entire population of the evolutionary algorithm. If we apply the local search from many individual the possibility of diversity losses or premature convergence increases. Also, the computational cost (in terms of function evaluations) will increase due to step size computation. When selecting the starting point for the local search, the straight decision is to choose the individual with the most significant crowding distance value in order to assure the existence of close neighbors. These

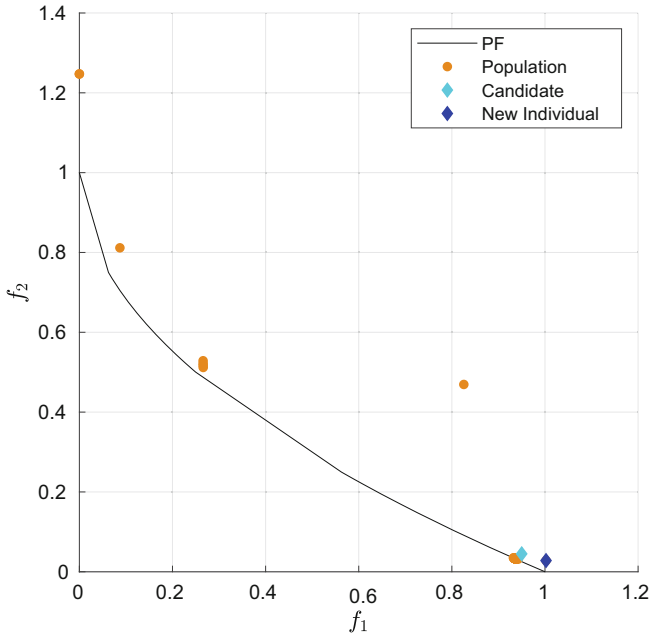


**Fig. 7.** This figure illustrates a particular instant of a certain iteration in the population-based algorithm. The population is represented by circles, while the initial and final solutions involved in the local search procedure are marked by light and dark diamonds correspondingly. The image of the constraint function is also marked in the figure. The tested function corresponds to the CZDT3 from [31].

neighbors are used to approximate the gradient information required by the proposed operator (see Fig. 5). By doing this, there is a chance to generate a new individual such that: (i) it is not that far from  $x_s$ ; but (ii) it can be deleted by the crowding process itself. Therefore, there is a compromise between choosing a candidate that has enough neighbors to approximate the gradient-free MODD and the chances of losing the new candidate due to crowding. A better idea could be to choose an individual  $x_s$  with an average crowding value and at least  $r$  close neighbors.

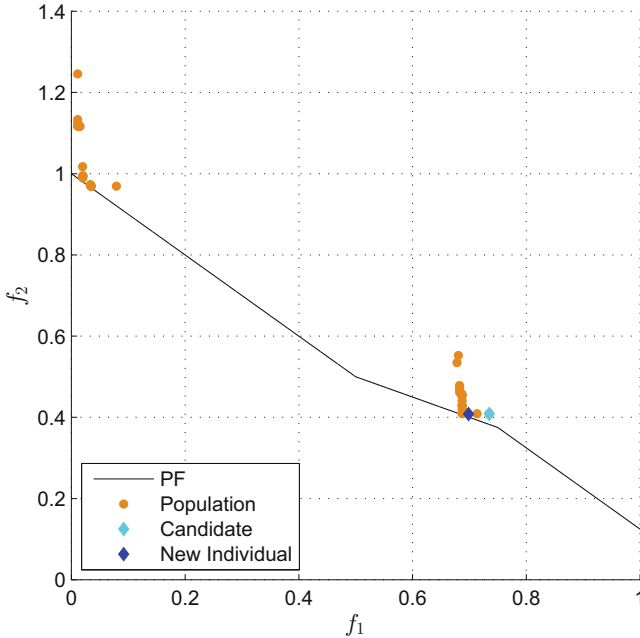
Numerical results that support the advantages of the use of this proposal can be found in [38]. Next, we present some examples of the performance of a population-based algorithm when applying this proposed low-cost local search. Figures 6 and 7 illustrates the application of NSGA-II applied to the CZDT2 and CZDT3 benchmark functions given in [31]. These functions are bi-objective optimization problems with equality constraints. The figures show a certain





**Fig. 8.** This figure illustrates a particular instant of a certain iteration in the population-based algorithm. The population is represented by circles, while the initial and final solutions involved in the local search procedure are marked by light and dark diamonds correspondingly. The tested function corresponds to the CF2 from [39].

generation of the Multi-objective Evolutionary Algorithm (MOEA) when the local search is applied to generate a new individual. Figures 8 and 9 illustrates the application of NSGA-II applied to the CF2 and CF4 benchmark functions [39]. These functions are bi-objective optimization problems with inequality constraints. The figures show a certain generation of the MOEA when the local search is applied to generate a new individual.



**Fig. 9.** This figure illustrates a particular instant of a certain iteration in the population-based algorithm. The population is represented by circles, while the initial and final solutions involved in the local search procedure are marked by light and dark diamonds correspondingly. The tested function corresponds to the CF4 from [39].

## 5 Conclusions

In this chapter we have reviewed some tools that allow to realize a local search engine for constrained bi-objective optimization problems with a low cost when using within set based optimization strategies such as evolutionary algorithms. The basic idea of the Gradient Subspace Approximation is to compute the most greedy search direction at a given point out of given neighborhood information. Next, we have presented a way of how to compute descent directions for bi-objective optimization problems. The method can be applied both to unconstrained as well as to constrained problems. Next, we have shown how to utilize GSA in order to obtain a “gradient free” approximation of these search directions. Finally, we have demonstrated the possible benefit of such a resulting low cost searcher on a hybrid evolutionary algorithm, which couples the proposed search technique with the widely used evolutionary algorithm NSGA-II. We stress, however, that the gradient free local search engine can in principle be integrated into any other evolutionary algorithm or any other set-oriented search heuristic.

**Acknowledgements.** The authors acknowledge support from Conacyt Basic Science project no. 285599, SEP-Cinvestav project no. 231, and IPN project no. SIP20201381.

## References

1. Alvarado, S., Segura, C., Schütze, O., Zapotecas, S.: The gradient subspace approximation as local search engine within evolutionary multi-objective optimization algorithms. *Computación y Sistemas* **22**(2) (2018)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**(3), 1653–1669 (2007)
3. Bogoya, J., Vargas, A., Cuate, O., Schütze, O.: A  $(p, q)$ -averaged Hausdorff distance for arbitrary measurable sets. *Math. Comput. Appl.* **23**(3), 51 (2018)
4. Bosman, P.A.: On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. *IEEE Trans. Evol. Comput.* **16**(1), 51–69 (2011)
5. Brown, M., Smith, R.E.: Effective use of directional information in multi-objective evolutionary computation. In: *Genetic and Evolutionary Computation Conference*, pp. 778–789. Springer (2003)
6. Brown, M., Smith, R.E.: Directed multi-objective optimization. *Int. J. Comput. Syst. Signals* **6**(1), 3–17 (2005)
7. Coello, C.C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edn. Springer, New York (2007). ISBN 978-0-387-33254-3
8. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, New York (2001)
9. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2014)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
11. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. *J. Optim. Theory Appl.* **124**(1), 113–155 (2005)
12. Dilettoso, E., Rizzo, S.A., Salerno, N.: A weakly Pareto compliant quality indicator. *Math. Comput. Appl.* **22**(1), 25 (2017)
13. Domínguez, I.S., Aguirre, A.H., Valdez, S.I.: A new EDA by a gradient-driven density. In: *International Conference on Parallel Problem Solving from Nature*, pp. 352–361. Springer (2014)
14. Fliege, J., Drummond, L.G., Svaiter, B.F.: Newton’s method for multiobjective optimization. *SIAM J. Optim.* **20**(2), 602–626 (2009)
15. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.* **51**(3), 479–494 (2000)
16. Gebken, B., Peitz, S., Dellnitz, M.: A descent method for equality and inequality constrained multiobjective optimization problems. In: *Numerical and Evolutionary Optimization*, pp. 29–61. Springer (2017)
17. Gebken, B., Peitz, S., Dellnitz, M.: On the hierarchical structure of Pareto critical sets. In: *AIP Conference Proceedings*, vol. 2070, p. 020041. AIP Publishing (2019)
18. Griewank, A., Walther, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, vol. 105. Siam, Philadelphia (2008)
19. Harada, K., Sakuma, J., Kobayashi, S.: Local search for multiobjective function optimization: Pareto descent method. In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 659–666. ACM (2006)
20. Hazen, M., Gupta, M.R.: A multiresolutional estimated gradient architecture for global optimization. In: *2006 IEEE International Conference on Evolutionary Computation*, pp. 3013–3020. IEEE (2006)

21. Hillermeier, C.: *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, vol. 135. Springer, Heidelberg (2001)
22. Jahn, J.: Multiobjective search algorithm with subdivision technique. *Comput. Optim. Appl.* **35**(2), 161–175 (2006)
23. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **18**(4), 602–622 (2014)
24. Kleijnen, J.P.: Response surface methodology. In: *Handbook of simulation optimization*, pp. 81–104. Springer (2015)
25. Lara, A.: Using gradient based information to build hybrid multi-objective evolutionary algorithms. Ph.D. thesis, Computer Science Department, CINVESTAV-IPN (2012)
26. Lara, A., Sanchez, G., Coello, C.A.C., Schütze, O.: HCS: a new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **14**(1), 112–132 (2009)
27. Martín, A., Schütze, O.: Pareto tracer: a predictor-corrector method for multi-objective optimization problems. *Eng. Optim.* **50**(3), 516–536 (2018)
28. Miettinen, K.: *Nonlinear Multiobjective Optimization*, vol. 12. Springer, Heidelberg (2012)
29. Nocedal, J., Wright, S.: *Numerical Optimization*. Springer, Heidelberg (2006)
30. Peitz, S., Dellnitz, M.: A survey of recent trends in multiobjective optimal control-surrogate models, feedback control and objective reduction. *Math. Comput. Appl.* **23**(2), 30 (2018)
31. Saha, A., Ray, T.: Equality constrained multi-objective optimization. In: *2012 IEEE Congress on Evolutionary Computation*, pp. 1–7 (2012)
32. Schäffler, S., Schultz, R., Weinzierl, K.: Stochastic method for the solution of unconstrained vector optimization problems. *J. Optim. Theory Appl.* **114**(1), 209–222 (2002)
33. Schütze, O., Alvarado, S., Segura, C., Landa, R.: Gradient subspace approximation: a direct search method for memetic computing. *Soft Comput.* **21**, 6331–6350 (2016)
34. Schütze, O., Esquivel, X., Lara, A., Coello, C.C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **16**(4), 504–522 (2012)
35. Schütze, O., Lara, A., Coello, C.C.: The directed search method for unconstrained multi-objective optimization problems. In: *Proceedings of the EVOLVE–A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation*, pp. 1–4 (2011)
36. Schütze, O., Martín, A., Lara, A., Alvarado, S., Salinas, E., Coello, C.A.C.: The directed search method for multi-objective memetic algorithms. *Comput. Optim. Appl.* **63**(2), 305–332 (2016)
37. Sun, J.Q., Xiong, F.R., Schütze, O., Hernández, C.: *Cell mapping methods - algorithmic approaches and applications*. Springer (2018)
38. Uribe, L., Lara, A., Schütze, O.: On the efficient computation and use of multi-objective descent directions within constrained MOEAs. *Swarm Evol. Comput.* **52**, 100617 (2020)
39. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. Technical report special session on performance assessment of multi-objective optimization algorithms, University of Essex, Colchester, UK and Nanyang technological University, Singapore **264** (2008)