Book

Bagirov A., Gaudioso M., Karmitsa N., Mäkelä M., Taheri S. (eds) Numerical Nonsmooth Optimization. Springer, Cham.
ISBN 978-3-030-34909-7 (PRINT)
ISBN 978-3-030-34910-3 (ONLINE)

DOI https://doi.org/10.1007/978-3-030-34910-3_14

CITATION Montonen O., Joki K. (2020) Multiobjective Double Bundle Method for DC Optimization. In: Bagirov A., Gaudioso M., Karmitsa N., Mäkelä M., Taheri S. (eds) Numerical Nonsmooth Optimization. Springer, Cham. URL: https://doi.org/10.1007/978-3-030-34910-3_14

Multiobjective double bundle method for DC optimization

Outi Montonen and Kaisa Joki

Abstract We discuss about the multiobjective double bundle method for nonsmooth multiobjective optimization where objective and constraint functions are presented as a difference of two convex (DC) functions. By utilizing a special technique called the improvement function, we are able to handle several objectives and constraints simultaneously. The method improves every objective at each iteration and the improvement function preserves DC property of the objectives and constraints. Once the improvement function is formed, we can approximate it by using a special cutting plane model capturing the convex and concave behaviour of a DC function. We solve the problem with a modified version of the single-objective double bundle method using the cutting plane model as a objective. The multiobjective double bundle method is proved to be finitely convergent to a weakly Pareto stationary solution under mild assumptions. Moreover, the applicability of the method is considered.

1 Introduction to multiobjective DC optimization

The vast range of practical optimization problems involve several goals. Usually, these goals are conflicting and compromises have to be made. Thus, it is sensible to study multiobjective optimization [8, 28]. The real-life application areas for multiobjective optimization are for example chemical engineering [36], cancer treatment planning [6], and humanitarian aid [12]. Compared to

Outi Montonen
University of Turku, Department of Mathematics and Statistics, FI-20014 Turku, Finland, e-mail: outi.montonen@utu.fi

Kaisa Joki
University of Turku, Department of Mathematics and Statistics, FI-20014 Turku, Finland e-mail: kaisa.joki@utu.fi

single-objective optimization, in multiobjective optimization various objectives need to be handled simultaneously. One fundamental idea to solve a multiobjective problem is the scalarization [8, 28], where several objectives are transformed into one objective, and then some efficient single-objective method can be applied. A typical feature of scalarization is that we have to take a stand on the relative importance of objectives.

As many of the practical applications have nonsmooth nature as well, we focus here on nonsmooth multiobjective optimization. Instead of the scalarization, we want to treat the objectives as they are, and in particular, to investigate descent methods. We classify a method to be of a descent one if at each iteration it improves every objective, and in that sense, every objective is considered to be equally important. Some examples of descent methods can be found in the literature for convex (see e.g. [3, 4, 21, 31, 33]) and for nonconvex (see e.g. [26, 29, 35, 41]) problems.

Here we limit our study to DC functions (i.e. functions which can be represented as a difference of convex functions, so-called DC components) forming a wide subclass of nonconvex functions. In addition to the wideness, the class of DC functions has another unquestionable advantage compared to general nonconvex functions. Due to the convex DC components, we are able to utilize the convex analysis. Unfortunately, it may be hard to define DC components for a function even if the function is known to be a DC function. Moreover, due to the fact that a DC decomposition is not unique, the found DC decompostion may not be the most suitable one. However, there exist many practical applications where the objectives are in the explicit DC form, like in clustering [2], spherical separability problems [11], production-transportation planning [15], wireless sensor network planning [1], and data visualization [5]. It is worth noticing that these applications usually either model the problem directly as a single-objective problem or scalarize a bi-objective problem, even if they have multiobjective nature. Additionally, in [18] a probabilistic lot sizing model is solved as a multiobjective DC problem.

In this chapter, we are aiming to solve a DC problem of the form

$$\min_{\mathbf{x}\in X} \quad f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}), \tag{1}$$

where $X = \{\mathbf{x} \in \mathbb{R}^n \,|\, g_l(\mathbf{x}) \leq 0,\, l \in \mathcal{L}\}$ and $\mathcal{L} = \{1, \ldots, m\}$. The objectives $f_i = p_i - q_i : \mathbb{R}^n \to \mathbb{R}$ for all $i \in \mathcal{I}$, such that the set $\mathcal{I} = \{1, \ldots, k\}$ denotes the indices of the objectives, and the constraints $g_l = r_l - s_l : \mathbb{R}^n \to \mathbb{R}$ for all $l \in \mathcal{L}$ are assumed to be at least partially conflicting and DC functions. Thus, the DC components p_i, q_i, r_l, s_l for all $i \in \mathcal{I}, l \in \mathcal{L}$ are convex.

The theory of DC functions has been widely studied in the past few decades (see e.g. [13, 14, 40]). Additionally, various methods have been developed for single-objective DC problems like DCA [23, 24, 32], proximal point based methods [38], special cutting plane based bundle methods [10, 19, 20], and branch-and-bound and outer approximation algorithms [16]. However, a little research devoted to multiobjective DC optimization focuses mainly on opti-

mality conditions like in [9, 34, 39]. Moreover, the exact and inexact proximal point methods in [17, 18] have lately introduced.

Our contribution to the field of multiobjective DC optimization is the multiobjective double bundle method for DC optimization (MDBDC) originally presented in [30]. MDBDC generalizes the single-objective double bundle method (DBDC) [20] to the multiobjective optimization by utilizing the improvement function presented in [21, 26, 41]. MDBDC has some distinctive features. First, MDBDC can handle several DC objectives together with inequality DC constraints. Second, under mild assumptions MDBDC converges finitely to a weakly Pareto stationary solution. Last, MDBDC is a descent type method. In addition, MDBDC has proved to be usable also in practice, and when compared to a general nonconvex solver, MDBDC can obtain better solutions (i.e. every objective has a better value) by taking into account the DC structure of the problem [30].

We say some words about the comparison between MDBDC and the exact proximal point algorithm developed in [17]. Both of these methods base their ability to handle multiple objectives to the fact that the optimum of some single-objective problem is known to be a weak Pareto optimum, and they both utilize convex single-objective subproblems. However, in MDBDC the subgradients for both DC components needs to be evaluated while in the proximal point algorithm only the subgradient of the second DC component needs to be known. The first clear difference between the methods is that MDBDC can handle DC constraints while the proximal point algorithm is designed only for convex constraints. In addition, MDBDC is a descent type method improving every objective at each iteration. In the proximal point algorithm, the descent direction is provided only for a function $\sum_{i\in\mathcal{I}} \lambda_i f_i(\mathbf{x})$ such that $\sum_{i\in\mathcal{I}} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i \in \mathcal{I}$. Thus, every objective is not necessarily improved. Finally, MDBDC produces weakly Pareto stationary solutions while the proximal point algorithm yields only Pareto critical solutions. This theoretical difference is discussed with details in Section 2.

The rest of this chapter is organized as follows. First, in Section 2, we consider the optimality in multiobjective optimization and exemplify the difference between Pareto critical and weakly Pareto stationary solutions. Section 3 compresses all you need to know about MDBDC into a sketch of the method and proves its finite convergence. In Section 4, the numerical behaviour of MDBDC is discussed and argued in favour of to utilize a method specially designed for DC functions instead of a general nonconvex method. Finally, some concluding remarks are given in Section 5.

2 Critical versus stationary solution in multiobjective DC optimization

We begin by saying some words about optimality in multiobjective optimization. A solution $\mathbf{x}^* \in X$ is a *global Pareto optimum* for the problem (1) if there does not exist another solution $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i \in \mathcal{I}$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one $j \in \mathcal{I}$. This means that we cannot improve any objective without deteriorating some other objective simultaneously. If we have a solution such that there does not exist any other solution yielding better values for every objective, we call the solution a *global weak Pareto optimum*. In other words, a solution $\mathbf{x}^* \in X$ is a global weak Pareto optimum for the problem (1) if there does not exist another solution $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ for all $i \in \mathcal{I}$. Moreover, a solution $\mathbf{x}^*$ is a *local (weak) Pareto optimum* for the problem (1) if there exists $\varepsilon > 0$ such that $\mathbf{x}^*$ is a global (weak) Pareto optimum on $X \cap B(\mathbf{x}^*; \varepsilon)$. To conclude, every Pareto optimum is also a weak Pareto optimum and instead of only one optimum we usually have several (weak) Pareto optima.

We start with defining weak Pareto stationary and critical solutions and after that the difference between these two concepts is exemplified. To simplify notations, we denote by

$$F(\mathbf{x}) = \bigcup_{i \in \mathcal{I}} \partial f_i(\mathbf{x}) \text{ and } G(\mathbf{x}) = \bigcup_{l \in L(\mathbf{x})} \partial g_l(\mathbf{x}),$$

where $\partial f_i(\mathbf{x})$ and $\partial g_l(\mathbf{x})$ are subdifferentials of f_i and g_l at $\mathbf{x} \in \mathbb{R}^n$, respectively and $L(\mathbf{x}) = \{l \in \mathcal{L} \,|\, g_l(\mathbf{x}) = 0\}$. Next, we give a necessary condition for a solution to be a local weak Pareto optimum of the problem (1) when objectives and constraints are general nonconvex functions. Recall that for a set $S \subseteq \mathbb{R}^n$ we denote by $K_S(\mathbf{x})$ and $S^{\leq}$ a contingent cone at $\mathbf{x} \in \mathbb{R}^n$ and a polar cone, respectively.

Theorem 1. [25] *If $\mathbf{x}^* \in X$ is a local weak Pareto optimum for the problem (1) with LLC objective and constraint functions, and the constraint qualification $G^{\leq}(\mathbf{x}^*) \subseteq K_X(\mathbf{x}^*)$ holds, then*

$$\mathbf{0} \in \operatorname{conv} F(\mathbf{x}^*) + \operatorname{cl} \operatorname{cone} G(\mathbf{x}^*). \tag{2}$$

Proof. See Theorem 15 in [25]

We say that the point $\mathbf{x}^*$ satisfying the condition (2) is *weakly Pareto stationary*. In general, nonconvex multiobjective methods find a solution $\mathbf{x}^* \in X$ being weakly Pareto stationary.

A suitable necessary condition for the problem (1) can be derived also by assuming that the objectives and constraints are DC functions. The condition like this is given in Theorem 3.1 in [34]. For our purposes to illustrate the properties of different conditions, it is enough to consider the unconstrained

case of this condition given in [17]: If $\mathbf{x}' \in \mathbb{R}^n$ is a local weak Pareto optimum for the problem (1), where $X = \mathbb{R}^n$ and the objectives are DC functions, then

$$\operatorname{conv}\{\partial q_i(\mathbf{x}') \,|\, i \in \mathcal{I}\} \subseteq \operatorname{conv}\{\partial p_i(\mathbf{x}') \,|\, i \in \mathcal{I}\}.$$

However, this condition is hard to verify in practice, and thus in [17], the condition

$$\mathbf{0} \in \operatorname{conv}\{\partial p_i(\mathbf{x}') - \partial q_i(\mathbf{x}') \,|\, i \in \mathcal{I}\} \tag{3}$$

is validated and the solution $\mathbf{x}'$ satisfying this condition is called *Pareto critical*. Clearly, weakly Pareto stationary solution $\mathbf{x}^*$ is also Pareto critical, since

$$\mathbf{0} \in \operatorname{conv}\{\partial f_i(\mathbf{x}^*) \,|\, i \in \mathcal{I}\} \subseteq \operatorname{conv}\{\partial p_i(\mathbf{x}^*) - \partial q_i(\mathbf{x}^*) \,|\, i \in \mathcal{I}\}.$$

To argue why the inverse does not necessarily hold, we give an example.

Example 1. Consider the unconstrained bi-objective case of the problem (1) and let the DC components be $p_1(x) = \max\{-x, 2x\}$, $q_1(x) = \max\{-2x, x\}$, $p_2(x) = \max\{x^2, x\}$, and $q_2(x) = \max\{0.5x^2, -x\}$, where $x \in \mathbb{R}$. We consider the point $x' = 0$ and to verify its Pareto criticality we investigate the intersection

$$\lambda\partial p_1(x') + (1-\lambda)\partial p_2(x') \cap \lambda\partial q_1(x') + (1-\lambda)\partial q_2(x').$$

At the point x', neither of the objectives is differentiable, and the intersection is of the form $\lambda[-1,2] + (1-\lambda)[0,1] \cap \lambda[-2,1] + (1-\lambda)[-1,0]$. For instance, with $\lambda = 1$ this intersection equals $[-1,1]$ being a nonempty set. Thus, the condition (3) is valid and x' is Pareto critical. However, x' is not weakly Pareto stationary, since $0 \notin \operatorname{conv}\{\partial f_1(x'), \partial f_2(x')\} = \{1\}$. The similar observation can be made in the single-objective case as well, as was exemplified in [20].

A natural approach towards solving the multiobjective DC problem would be to verify that the condition (3) is satisfied. However, in order to reduce the number of possible non-optimal solutions in our set of feasible solutions, we want to ensure that the solution produced is weakly Pareto stationary. To obtain a stationary solution is not a trivial task even in the single-objective case and to obtain weak Pareto stationarity in the multiobjective case, we introduce the *improvement function* $H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ [21, 41] defined by

$$H(\mathbf{x}, \mathbf{y}) = \max\{f_i(\mathbf{x}) - f_i(\mathbf{y}),\, g_l(\mathbf{x}) \,|\, i \in \mathcal{I}, l \in \mathcal{L}\}. \tag{4}$$

One reason for the utility of the improvement function in the case of multiobjective DC optimization raises from the fact that it is a DC function. Indeed, since the objectives f_i for all $i \in \mathcal{I}$ and the constraints g_l for all $l \in \mathcal{L}$ are assumed to be DC functions, then $H(\,\cdot\,\mathbf{y})$ is a DC function as a maximum of DC functions [14]. Moreover, the improvement function has three useful

elementary properties stated in the following theorem legitimating the use of it.

Theorem 2. [41, 26] *The improvement function* $H(\cdot, \mathbf{y})$ (4) *has the following properties:*

(i) *If* $H(\mathbf{x}, \mathbf{y}) < H(\mathbf{y}, \mathbf{y})$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in X$ *then* $f_i(\mathbf{x}) < f_i(\mathbf{y})$ *for all* $i \in \mathcal{I}$ *and* $g_l(\mathbf{x}) < 0$ *for all* $l \in \mathcal{L}$.

(ii) *If the solution* $\mathbf{x}^* \in X$ *is a global weak Pareto optimum of the problem* (1)*, then*

$$\mathbf{x}^* = argmin_{\mathbf{x} \in \mathbb{R}^n} H(\mathbf{x}, \mathbf{x}^*).$$

(iii) *If* $\mathbf{0} \in \partial H(\mathbf{x}^*, \mathbf{x}^*)$*, then the solution* $\mathbf{x}^* \in X$ *of the problem* (1) *is weakly Pareto stationary.*

Proof. See the proof of Theorem 2 in [30].

Based on the above theorem, we obtain a weakly Pareto stationary solution $\mathbf{x}^*$ for the problem (1) if we find a *Clarke stationary solution* for $H(\cdot, \mathbf{x}^*)$ (i.e. $\mathbf{0} \in \partial H(\mathbf{x}^*, \mathbf{x}^*)$). Therefore, we are able to utilize some single-objective DC method to find a Clarke stationary solution. However, many of these methods produce only critical solutions. Since we are able to use the DC structure of the improvement function as an advantage, we will apply the escape procedure presented in [20] being able to test whether an approximate Clarke stationary condition is valid. Additionally, if this condition is not satisfied, the procedure generates a new search direction.

3 Multiobjective double bundle method for DC optimization

The multiobjective double bundle method for DC optimization (MDBDC) collects ideas from three different bundle type methods and combines them into one package such that a method for multiobjective DC optimization with inequality DC constraints is created. These three methods are the multiobjective proximal bundle method (MPB) [26, 29], the proximal bundle method for DC optimization (PBDC) [19], and the double bundle method for DC optimization (DBDC) [20].

The idea how to apply the improvement function with a single-objective bundle-based method is absorbed from MPB being a bundle-based method to solve nonconvex constrained multiobjective problems. In MPB, the technique utilizing improvement function [21, 26, 41] was combined with the single-objective proximal bundle method [22]. Besides the ability to handle several objectives simultaneously, the improvement function gives a way to handle constraints as well. Additionally, the descent property of MDBDC is the direct consequence of the properties of the improvement function.

While MPB gives ideas to treat multiple objectives, PBDC gives core ingredients for the algorithm of MDBDC related to the single-objective part. The cutting plane model used in MDBDC for the improvement function bases on the one used in PBDC. The peculiarity of this model is that it captures both the convex and concave behaviour of a DC function by utilizing explicitly the DC decomposition.

DBDC is an unconstrained single-objective sibling of MDBDC. In order to avoid Pareto critical solutions in MDBDC, we take an advantage of the escape procedure being part of DBDC. DBDC is developed from PBDC by adding the escape procedure giving an ability to avoid critical solutions. If we end up to a critical point, the procedure either gives a new descent search direction or ensures that the approximate Clarke stationary condition is valid.

MDBDC is designed to solve the problem (1) with DC objectives and constraints. The method improves all the objectives simultaneously meaning that MDBDC is a method of descent type. The idea of MDBDC in its simplicity is to apply the improvement function such that instead of a constrained multiobjective problem we can solve an unconstrained single-objective one. This new problem is then solved by using a special cutting plane model and the modified version of DBDC. As a result of this process, we end up to a weakly Pareto stationary solution. Here, we describe the general idea of MDBDC and the more detailed description can be found in [30].

3.1 Improvement function and cutting plane model

Since the cutting plane model used in MDBDC utilizes the DC decomposition of the objective, we discuss first about the DC decomposition of the improvement function. As it was previously mentioned, the improvement function formed by DC functions is a DC function, and thus, the DC decomposition exists. This decomposition can be obtained as in [14]. For example, we can rewrite the objectives $f_i = p_i - q_i$ for all $i \in \mathcal{I}$ and the constraints $g_l = r_l - s_l$ for all $l \in \mathcal{L}$ as

$$f_i(\mathbf{x}) = p_i(\mathbf{x}) + \sum_{\substack{j \in \mathcal{I} \\ j \neq i}} q_j(\mathbf{x}) + \sum_{t \in \mathcal{L}} s_t(\mathbf{x}) - \sum_{j \in \mathcal{I}} q_j(\mathbf{x}) - \sum_{t \in \mathcal{L}} s_t(\mathbf{x})$$

$$g_l(\mathbf{x}) = r_l(\mathbf{x}) + \sum_{\substack{t \in \mathcal{L} \\ t \neq l}} s_t(\mathbf{x}) + \sum_{j \in \mathcal{I}} q_j(\mathbf{x}) - \sum_{j \in \mathcal{I}} q_j(\mathbf{x}) - \sum_{t \in \mathcal{L}} s_t(\mathbf{x}).$$

In order to simplify the presentation, we denote

$$A_i(\mathbf{x},\mathbf{y}) = p_i(\mathbf{x}) + \sum_{\substack{j\in\mathcal{I}\\ j\neq i}} q_j(\mathbf{x}) + \sum_{t\in\mathcal{L}} s_t(\mathbf{x}) - f_i(\mathbf{y}) \quad \text{and}$$
$$B_l(\mathbf{x}) = r_l(\mathbf{x}) + \sum_{\substack{t\in\mathcal{L}\\ t\neq l}} s_t(\mathbf{x}) + \sum_{j\in\mathcal{I}} q_j(\mathbf{x}).$$

Now the DC decomposition of the improvement function is of the form

$$H(\mathbf{x},\mathbf{y}) = H_1(\mathbf{x},\mathbf{y}) - H_2(\mathbf{x}),$$

where

$$H_1(\mathbf{x},\mathbf{y}) = \max\{A_i(\mathbf{x},\mathbf{y}), B_l(\mathbf{x}) \,|\, i\in\mathcal{I}, l\in\mathcal{L}\} \quad \text{and} \tag{5}$$
$$H_2(\mathbf{x}) = \sum_{j\in\mathcal{I}} q_j(\mathbf{x}) + \sum_{t\in\mathcal{L}} s_t(\mathbf{x}).$$

Both DC components H_1 and H_2 are convex with respect to $\mathbf{x}$ and the vector $\mathbf{y}$ is treated as a constant.

As the name of MDBDC suggests, we collect information from the previous iterations into bundles. In the following, the index h denotes the h-th iteration and $\mathbf{x}_h$ is the current iteration point. We assume that at each auxiliary point the function value and one arbitrary subgradient of p_i, q_i, r_l and s_l can be evaluated. From these, we can compose values for functions $A_i(\,\cdot\,,\mathbf{x}_h)$, B_l, $H_1(\,\cdot\,,\mathbf{x}_h)$, and H_2 and their subgradients $\mathbf{a}_i$, $\mathbf{b}_l$, $\mathbf{h}_1$, and $\mathbf{h}_2$, respectively.

In MDBDC, we collect information into two separate bundles $\mathcal{B}_1^h$ and $\mathcal{B}_2^h$ for $H_1(\,\cdot\,,\mathbf{y})$ and H_2, respectively. The bundles consist of triplets formed by an auxiliary point $\mathbf{y}_j$, a corresponding function value, and a subgradient. Here the index j is an element of the index set $\mathcal{J}_1^h$ or $\mathcal{J}_2^h$ depending on whether the bundle is for $H_1(\,\cdot\,,\mathbf{x}_h)$ or H_2. In practice, the bundle $\mathcal{B}_1^h$ is formed by having separate bundles for each $A_i(\,\cdot\,,\mathbf{x}_h)$, $i\in\mathcal{I}$ and B_l, $l\in\mathcal{L}$ and taking the union of them.

In order to find a search direction, we approximate the improvement function by utilizing the special cutting plane model which is based on the one proposed in [19]. Therefore, we linearize the convex DC components separately by utilizing the classical cutting plane model [22, 27, 37]. This way, we can capture both the convex and concave behaviour of the improvement function. To form an approximation for $H_1(\,\cdot\,,\mathbf{x}_h)$ and H_2, we linearize all the components $A_i(\,\cdot\,,\mathbf{x}_h)$ and B_l of $H_1(\,\cdot\,,\mathbf{x}_h)$ and H_2. We begin by giving the linearization for $A_i(\,\cdot\,,\mathbf{x}_h)$:

$$\hat{A}_i^h(\mathbf{x}) = \max_{j\in\mathcal{J}_1^h} \left\{ A_i(\mathbf{x}_h,\mathbf{x}_h) + \mathbf{a}_{i,j}^\top(\mathbf{x}-\mathbf{x}_h) - \alpha_{i,j}^A \right\},$$

where $\mathbf{a}_{i,j} \in \partial A_i(\mathbf{y}_j,\mathbf{x}_h)$ for $j\in\mathcal{J}_1^h$. The linearization errors evaluated at $\mathbf{x}_h$ for all $j\in\mathcal{J}_1^h$ are

$$\alpha_{i,j}^A = A_i(\mathbf{x}_h, \mathbf{x}_h) - A_i(\mathbf{y}_j, \mathbf{x}_h) - \mathbf{a}_{i,j}^\top(\mathbf{x}_h - \mathbf{y}_j) \text{ for all } i \in \mathcal{I}$$

Note that due to the convexity, all the linearization errors are nonnegative.

Similarly, we can linearize functions B_l for all $l \in \mathcal{L}$ and H_2 and denote these approximations by $\hat{B}_l^h(\mathbf{x})$ and $\hat{H}_2^h(\mathbf{x})$, respectively. Thus, we obtain the cutting plane model for $H_1(\,\cdot\,, \mathbf{x}_h)$ of the form

$$\hat{H}_1^h(\mathbf{x}) = \max\{\hat{A}_i^h(\mathbf{x}),\ \hat{B}_l^h(\mathbf{x}) \,|\, i \in \mathcal{I}, l \in \mathcal{L}\}. \tag{6}$$

Finally, by utilizing the above approximations, we obtain the following piecewise linear, nonconvex DC cutting plane model of $H(\,\cdot\,, \mathbf{x}_h)$:

$$\hat{H}^h(\mathbf{x}) = \hat{H}_1^h(\mathbf{x}) - \hat{H}_2^h(\mathbf{x}).$$

From the definition of the cutting plane model, it follows that $\hat{H}_1^h(\mathbf{x}_h + \mathbf{d}) \leq H_1(\mathbf{x}_h + \mathbf{d}, \mathbf{x}_h)$ and $\hat{H}_2^h(\mathbf{x}_h + \mathbf{d}) \leq H_2(\mathbf{x}_h + \mathbf{d})$ [30].

3.2 Direction finding

By bearing in mind Theorem 2, we are motivated to find a solution $\mathbf{x}^* \in X$ such that $\mathbf{0} \in \partial H(\mathbf{x}^*, \mathbf{x}^*)$. Therefore, we define a search direction as a solution of the problem

$$\min_{\mathbf{d} \in \mathbb{R}^n} \quad H(\mathbf{x}_h + \mathbf{d}, \mathbf{x}_h). \tag{7}$$

By utilizing the model of $H(\,\cdot\,, \mathbf{x}_h)$, we can estimate the problem (7) with a nonsmooth nonconvex DC problem

$$\min_{\mathbf{d} \in \mathbb{R}^n} \quad P^h(\mathbf{d}) = \hat{H}_1^h(\mathbf{x}_h + \mathbf{d}) - \hat{H}_2^h(\mathbf{x}_h + \mathbf{d}) + \frac{1}{2t}\|\mathbf{d}\|^2, \tag{8}$$

where $t > 0$ is a proximity parameter. The solution of this problem is denoted by $\mathbf{d}_t^h$.

We use the solution approach described in [23, 24, 32] to find a global solution of the problem (8). This approach can be applied, since the second DC component of P^h is $\hat{H}_2^h(\mathbf{x}_h + \mathbf{d})$ and it is polyhedral convex. The objective of the problem (8) can now be rewritten as

$$P^h(\mathbf{d}) = \min_{j \in \mathcal{J}_2^h} \left\{ P_j^h(\mathbf{d}) = \hat{H}_1^h(\mathbf{x}_h + \mathbf{d}) - H_2(\mathbf{x}_h) - \mathbf{h}_{2,j}^\top \mathbf{d} + \alpha_{2,j}^H + \frac{1}{2t}\|\mathbf{d}\|^2 \right\}.$$

This enables us to change the order of the minimization in the problem (8). Thus, we end up to solve $|\mathcal{J}_2^h|$ convex subproblems

$$\min_{\mathbf{d} \in \mathbb{R}^n} \{ P_j^h(\mathbf{d}) = \hat{H}_1^h(\mathbf{x}_h + \mathbf{d}) - H_2(\mathbf{x}_h) - \mathbf{h}_{2,j}^\top \mathbf{d} + \alpha_{2,j}^H + \frac{1}{2t}\|\mathbf{d}\|^2 \}, \tag{9}$$

where $j \in \mathcal{J}_2^h$. The solution of the individual subproblem $j \in \mathcal{J}_2^h$ is denoted by $\mathbf{d}_t^h(j)$, and the global solution $\mathbf{d}_t^h$ of the problem (8) is $\mathbf{d}_t^h = \mathbf{d}_t^h(j^*)$, where the index $j^* = \operatorname{argmin}\left\{P_j^h\left(\mathbf{d}_t^h(j)\right) \mid j \in \mathcal{J}_2^h\right\}$. In practice, the amount of computation can be controlled, since the size of the bundle $\mathcal{B}_2^h$ can be freely chosen such that $|\mathcal{J}_2^h| \geq 1$.

If $\|\mathbf{d}_t^h\| < \delta$, where $\delta > 0$ is a fixed stopping tolerance, we either generate a new descent direction or Clarke stationarity is achieved. In order to test Clarke stationarity, we need some information about the subdifferential $\partial H(\mathbf{x}_h, \mathbf{x}_h)$. Unfortunately, by calculating arbitrary subgradients $\mathbf{h}_1 \in \partial H_1(\mathbf{x}_h, \mathbf{x}_h)$ and $\mathbf{h}_2 \in \partial H_2(\mathbf{x}_h)$, we cannot guarantee that $\mathbf{h}_1 - \mathbf{h}_2 \in \partial H(\mathbf{x}_h, \mathbf{x}_h)$. Thus, we are justified to use the escape procedure (see Algorithm 1 in [20]) having the ability to select $\mathbf{h}_1^* \in \partial H_1(\mathbf{x}, \mathbf{x})$, $\mathbf{h}_2^* \in \partial H_2(\mathbf{x})$ for the objective function $H = H_1 - H_2$ at any $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{h}_1^* - \mathbf{h}_2^* \in \partial H(\mathbf{x}, \mathbf{x})$ is ensured [20].

At the j-th iteration of the escape procedure, we approximate the Goldstein ε-subdifferential $\partial_\varepsilon^G H(\mathbf{x}_h, \mathbf{x}_h)$ with a set U_j consisting of subgradients calculated as the difference of subgradients of the DC components. Thus, the new search direction can be found by calculating $\mathbf{d}_{j+1} = -\bar{\mathbf{u}}_j / \|\bar{\mathbf{u}}_j\|$, where $\bar{\mathbf{u}}_j$ is a solution of the problem

$$\min_{\mathbf{u} \in U_j} \frac{1}{2}\|\mathbf{u}\|^2.$$

If this direction is not descent or Clarke stationarity is not achieved, then the approximation of $\partial_\varepsilon^G H(\mathbf{x}_h, \mathbf{x}_h)$ is improved with a new subgradient. To conclude, in order to exit from the escape procedure, we either find a new descent search direction or $\|\bar{\mathbf{u}}_j\| \leq \delta$ meaning that the approximate Clarke stationary condition is satisfied and the algorithm of MDBDC is terminated.

3.3 Algorithm

In this section, we give a general idea of the method with the simplified version of the MDBDC algorithm. More detailed description of the MDBDC algorithm is given in [30].

We make some remarks about the algorithm. First, in Step 4 we execute the escape procedure given in Algorithm 1 in [20]. Note that in this case, the procedure is executed for the current iteration point $\mathbf{x}_h$ with the improvement function $H(\,\cdot\,, \mathbf{x}_h)$ as its objective. Second, we utilize the proximity parameter $t \in [t_{\min}, t_{\max}]$ in Algorithm 1, which can be either decreased in Step 6 and 7 or updated in Step 9 by utilizing the updating procedure inspired by the weighting update method in [22]. During the latter update, the proximity parameter may either increase or decrease. The update of the proximity parameter yields an improvement for the model in the both cases.

Algorithm 1: Multiobjective double bundle method for DC optimization (MDBDC)

Data: The stopping tolerance $\delta \in (0,1)$, the enlargement parameter $\theta > 0$, the decrease parameters $r, c \in (0,1)$, the increase parameter $R > 1$, and the descent parameter $m \in (0,1)$.

Step 0. (*Initialization*) Select $\mathbf{x}_0 \in X$ and calculate $\mathbf{h}_1(\mathbf{x}_0) \in \partial H_1(\mathbf{x}_0, \mathbf{x}_0)$ and $\mathbf{h}_2(\mathbf{x}_0) \in \partial H_2(\mathbf{x}_0)$. Initialize $\mathcal{B}_1^0$ and $\mathcal{B}_2^0$, $t = 0$, $\mathbf{h}_{2,\max} = \mathbf{0}$, and $h = 0$.

Step 1. (*Criticality*) If $\|\mathbf{h}_1(\mathbf{x}_h) - \mathbf{h}_2(\mathbf{x}_h)\| < \delta$, then $\mathbf{d}_t = \mathbf{0}$ and go to Step 4.

Step 2. (*Proximity parameter*) If $\|\mathbf{h}_2(\mathbf{x}_h)\| > \|\mathbf{h}_{2,\max}\|$, then $\mathbf{h}_{2,\max} = \mathbf{h}_2(\mathbf{x}_h)$. Set

$$t_{\min} = \frac{r\theta}{2(\|\mathbf{h}_1(\mathbf{x}_h)\| + \|\mathbf{h}_{2,\max}\|)} \tag{10}$$

and $t_{\max} = Rt_{\min}$. If $t \notin [t_{\min}, t_{\max}]$, then select $t \in [t_{\min}, t_{\max}]$.

Step 3. (*Search direction*) Calculate the search direction $\mathbf{d}_t$ as a solution of (8).

Step 4. (*Escape procedure*) If $\|\mathbf{d}_t\| < \delta$, then execute Algorithm 1 presented in [20] for the point $\mathbf{x}_h$ to obtain $\mathbf{x}^+$. Set $\mathbf{x}_{h+1} = \mathbf{x}^+$ and go to Step 8.

Step 5. (*Descent test*) Set $\mathbf{y} = \mathbf{x}_h + \mathbf{d}_t$. If

$$H(\mathbf{y}, \mathbf{x}_h) - H(\mathbf{x}_h, \mathbf{x}_h) \leq m\big(\hat{H}^h(\mathbf{y}) - H(\mathbf{x}_h, \mathbf{x}_h)\big),$$

then set $\mathbf{x}_{h+1} = \mathbf{y}$ and go to Step 8.

Step 6. (*Parameter update*) If $f_i(\mathbf{y}) > f_i(\mathbf{x}_0)$ for any $i \in \mathcal{I}$ and $\|\mathbf{d}_t\| > \theta$, then set $t = t - c(t - t_{\min})$. Go to Step 3.

Step 7. (*Bundle update*) Decrease t if necessary, and update $\mathcal{B}_1^h$ and $\mathcal{B}_2^h$. If a new subgradient $\mathbf{h}_2 \in \partial H_2(\mathbf{y})$ satisfies $\|\mathbf{h}_2\| > \|\mathbf{h}_{2,\max}\|$, then set $\mathbf{h}_{2,\max} = \mathbf{h}_2$ and update $t_{\min}$ using (10). Go to Step 3.

Step 8. (*Clarke stationarity*) If $\mathbf{x}_{h+1} = \mathbf{x}_h$, then Clarke stationarity is achieved and STOP with $\mathbf{x}^* = \mathbf{x}_h$ as the final solution.

Step 9. (*Model update*) Update t and the bundles $\mathcal{B}_1^{h+1} \subseteq \mathcal{B}_1^h$ and $\mathcal{B}_2^{h+1} \subseteq \mathcal{B}_2^h$ selected. Calculate $\mathbf{h}_1(\mathbf{x}_{h+1}) \in \partial H_1(\mathbf{x}_{h+1}, \mathbf{x}_{h+1})$ and $\mathbf{h}_2(\mathbf{x}_{h+1}) \in \partial H_2(\mathbf{x}_{h+1})$. Set $h = h + 1$ and go to Step 1.

As was mention, we give here a more general outline of the algorithm working well in theory. In practice, we can improve the numerical behaviour of MDBDC significantly. For example, in the initialization phase of Algortihm 1, the scaling procedure [30] may be executed. The positive affect of scaling has its roots in the DC decomposition of the improvement function. If the objective functions have different magnitudes, one of the DC components may dominate the others and hide their effects. Even if the scaling is executed, it does not affect the optimality of the solution, since the modified objectives have the same optima than the unmodified original objectives. Other possible numerical improvements are, for instance, to execute the escape procedure when the decrease in the model is nearly non-existing or to utilize more sophisticated update procedure for the proximity parameter in Step 7.

Lastly, some words about the bundles. Obviously, in practice the size of the bundles must be limited. The size of the bundle $\mathcal{B}_1^h$ has to be selected

such that information regarding both $A_i(\,\cdot\,,\mathbf{x}_h)$ and B_l for all $i \in \mathcal{I}$ and $l \in \mathcal{L}$ is included. Thus, $|\mathcal{J}_1^h| \geq k+m$. For the bundle $\mathcal{B}_2^h$, the size of the bundle $|\mathcal{J}_2^h| \geq 1$, since the bundle element associated to the current iteration point must be included. As was mention previously, via restriction of $|\mathcal{J}_2^h|$ the number of subproblems solved can be controlled.

3.4 Convergence analysis

The following convergence analysis is divided as follows: In Theorem 3, we state that MDBDC stops after a finite number of iterations at the point which satisfies the approximate Clarke stationary condition. In order to prove that, we need Lemma 1 to guarantee that the loop between Steps from 3 to 7 is finite and Theorem 4.11 in [20] to give the finite maximum number of iterations needed in the execution of the escape procedure in Step 4. Finally, Theorem 4 considers weak Pareto stationarity of the solution.

Throughout the convergence analysis, we assume that the following assumptions are valid:

Assumption 0.1 *The subdifferentials $\partial H_1(\mathbf{x},\mathbf{y})$ and $\partial H_2(\mathbf{x})$ are polytopes for each $\mathbf{x} \in \mathbb{R}^n$.*

Assumption 0.2 *The level set $\mathcal{F}_0 = \{\mathbf{x} \in X \mid f_i(\mathbf{x}) \leq f_i(\mathbf{x}_0),\ \text{for all } i \in \mathcal{I}\}$ is compact.*

We begin our convergence analysis by investigating the loop between Steps from 3 to 7. If this loop is infinite, it would lead to a contradiction as is seen in the proof of Lemma 1.

Lemma 1. [30] *Let Assumption 0.2 be valid. For any $\delta \in (0,1)$, Algorithm 1 cannot pass infinitely through the sequence of Steps from 3 to 7.*

Proof. The proof is similar to the one given in Theorem 5.8 in [20].

Now we show the finite convergence of Algorithm 1 to a solution satisfying the approximate Clarke stationary condition for the improvement function.

Theorem 3. [30] *Let Assumptions 0.1 and 0.2 be valid. For any $\delta \in (0,1)$ and $\varepsilon > 0$, the execution of Algorithm 1 stops after a finite number of iterations at the point $\mathbf{x}^*$ satisfying the approximate Clarke stationary condition $\|\boldsymbol{\xi}^*\| \leq \delta$, where $\boldsymbol{\xi}^* \in \partial_\varepsilon^G H(\mathbf{x}^*,\mathbf{x}^*)$.*

Proof. The execution of Algorithm 1 stops only if the Clarke stationary point $\mathbf{x}^*$ is found in Step 8 meaning that the approximate Clarke stationary condition is satisfied in the escape procedure. Assume, that Algorithm 1 is executed infinitely, and thus, the stopping condition in Step 8 is never satisfied.

Similarly to the proof of Theorem 5.9 in [20], we can deduce that after each iteration we have $H(\mathbf{x}_{h+1},\mathbf{x}_h) - H(\mathbf{x}_h,\mathbf{x}_h) \leq -\sigma < 0$, where

$$\sigma = \min\left\{\hat{m}\varepsilon\delta,\ \frac{m\delta^3}{Rr\theta}\right\} > 0$$

and $m, r \in (0,1)$, $R > 1$ and $\theta > 0$ are the parameters of the Algorithm 1. Additionally, $\varepsilon > 0$ and $\hat{m} \in (0,1)$ are the parameters of the escape procedure in Algorithm 1 in [20]. Due to the definition of $H(\,\cdot\,, \mathbf{x}_h)$ in (4), $H(\mathbf{x}_h, \mathbf{x}_h) = 0$ yielding that $H(\mathbf{x}_{h+1}, \mathbf{x}_h) \leq -\sigma$. Especially,

$$f_i(\mathbf{x}_{h+1}) - f_i(\mathbf{x}_h) < -\sigma < 0 \text{ for all } i \in \mathcal{I},$$

and after the h-th iteration,

$$f_i(\mathbf{x}_h) - f_i(\mathbf{x}_0) \leq -h\sigma \text{ for all } i \in \mathcal{I}.$$

By passing to the limit $h \to \infty$, we obtain

$$\lim_{h\to\infty} f_i(\mathbf{x}_h) - f_i(\mathbf{x}_0) \leq -\infty \text{ for all } i \in \mathcal{I}$$

yielding a contradiction, since based on Assumption 0.2 and the fact that DC functions are LLC, every f_i, $i \in \mathcal{I}$ must be bounded from below. □

Finally, we are interested in to argue that a Clarke stationary solution for the improvement function yields a weakly Pareto stationary solution for the original multiobjective problem. In order to prove this, the properties of the improvement function described in Theorem 2 are applied.

Theorem 4. [30] *Let f_i and g_l be DC functions for all $i \in \mathcal{I}$ and $l \in \mathcal{L}$. Suppose that Assumptions 0.1 and 0.2 are valid. Then, MDBDC stops after a finite number of iterations with the solution $\mathbf{x}^* \in X$ being a weakly Pareto stationary point for the problem* (1).

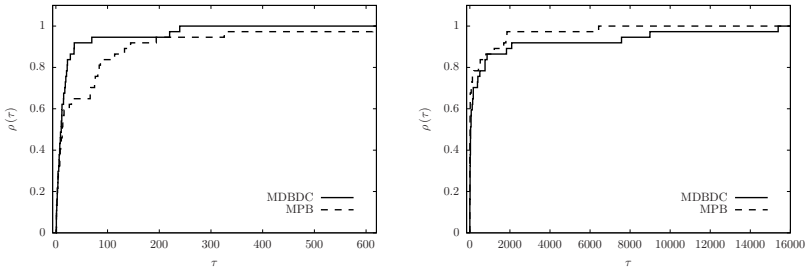
Proof. Consider minimization of an improvement function (4). By Theorem 3, after a finite number of iterations MDBDC finds a solution $\mathbf{x}^* \in \mathbb{R}^n$ such that $\mathbf{0} \in \partial H(\mathbf{x}^*, \mathbf{x}^*)$. According to Theorem 2 (iii), the solution $\mathbf{x}^* \in X$ is weakly Pareto stationary for the problem (1). □

4 Numerical behaviour of MDBDC

We discuss about the numerical behaviour of MDBDC by utilizing the computational experiments provided in [30], where the performances of MDBDC and MPB are compared. MPB has been selected as a reference method due to its somehow similar structure, but unlike MDBDC, it is designed for a general nonconvex problem. The 53 test problems are formed such that they all have either two or three objectives and the objectives are collected from academic single-objective DC problems. Some of the problems also include

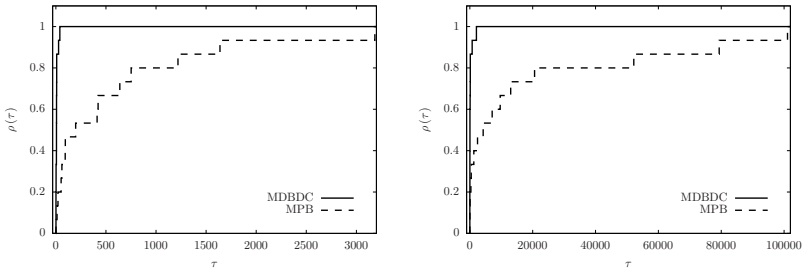
either a DC or concave constraint. The dimension of the problems varies from 2 to 500 such that 37 of them are small ($2 \leq n \leq 100$) and 16 are large ($250 \leq n \leq 500$). Since MDBDC and MPB both failed at one large test problem, we exclude this case from the discussion.

The numerical performance is illustrated in Figures 1 and 2 where the performance profiles [7] for small and large test problems are given, respectively. In the performance profiles, subgradient evaluations and CPU times are compared. In the small test problems, MDBDC wins MPB in terms of subgradient evaluations but in terms of CPU times MPB performs slightly better than MDBDC. Nevertheless, in the larger test problems, MDBDC beats MPB both in subgradient evaluations and CPU times. Therefore, we can conclude that, in the computational point of view, MDBDC is a good alternative for MPB in the case of DC problems.



(a) Subgradient evaluations (b) CPU

Fig. 1 Small test problems ($2 \leq n \leq 100$)



(a) Subgradient evaluations (b) CPU

Fig. 2 Large test problems ($250 \leq n \leq 500$)

Not only to compare the execution of the algorithms, another main goal in the numerical experiments in [30] was to emphasize the difference in the

solutions obtained. To compare solutions, we say that a solution is better than the other if it has better function values for every objective. In practice, it is possible that one method finds a better solution than the other even if they both find theoretically equally good solutions, namely weak Pareto stationary points. This is due to the fact that both local and global optima satisfy the condition (2) and the feasible set in the objective space is nonconvex. An interesting observation is that by taking into account the DC structure of the problem, MDBDC finds a better solution than MPB around 30% of the tests performed in [30], even though both methods find theoretically equally good solutions. Moreover, in half of the cases where MPB uses less computational efforts, MDBDC finds a better solution. This shows that the model used in MPB is more easily attracted to local optima.

5 Conclusions

We have discussed about the multiobjective double bundle method for DC optimization (MDBDC) being a method for multiobjective DC problems with inequality DC constraints. The method is descent and under mild assumptions it is proved to be finitely convergent to a weakly Pareto stationary solution. MDBDC has shown to behave well numerically and it is observed to be profitable to use a method taking into account a DC structure instead of a general nonconvex method.

MDBDC can be used in several ways. First, it can be used to solve only one weakly Pareto stationary solution, or execute it with different starting points to obtain an approximation of the set of local weak Pareto optima. Due to the descent property, the starting point is projected to the set of local weak Pareto optima in the decision space such that the solution obtained lies in the negative orthant from the starting point. Another possibility is to use MDBDC as a component of some interactive method as MPB was used in [29]. In addition, MDBDC is suitable to solve single-objective DC problems with DC constraints.

Acknowledgements This work was financially supported by the University of Turku and the Academy of Finland (Project 294002).

References

1. Astorino, A., Miglionico, G.: Optimizing sensor cover energy via DC programming. Optimization Letters **10**(2), 355–368 (2016)
2. Bagirov, A., Yearwood, J.: A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. European Journal of Operational Research **170**(2), 578–596 (2006)

3. Bello Cruz, J.Y., Iusem, A.N.: A strongly convergent method for nonsmooth convex minimization in Hilbert spaces. Numerical Functional Analysis and Optimization **32**(10), 1009–1018 (2011)
4. Bonnel, H., Iusem, A.N., Svaiter, B.F.: Proximal methods in vector optimization. SIAM Journal on Optimization **15**(4), 953–970 (2005)
5. Carrizosa, E., Guerrero, V., Romero Morales, D.: Visualizing data as objects by DC (difference of convex) optimization. Mathematical Programming **169**(1), 119–140 (2018)
6. Craft, D., Halabi, T., Shih, H.A., Bortfeld, T.: An approach for practical multiobjective IMRT treatment planning. International Journal of Radiation Oncology*Biology*Physics **69**(5), 1600–1607 (2007)
7. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**(2), 201–213 (2002)
8. Ehrgott, M.: Multicriteria Optimization, second edn. Springer, Berlin (2005)
9. Gadhi, N., Metrane, A.: Sufficient optimality condition for vector optimization problems under d.c. data. Journal of Global Optimization **28**(1), 55–66 (2004)
10. Gaudioso, M., Giallombardo, G., Miglionico, G., Bagirov, A.: Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations. Journal of Global Optimization **71**(1), 37–55 (2018)
11. Gaudioso, M., Gruzdeva, T.V., Strekalovsky, A.S.: On numerical solving the spherical separability problem. Journal of Global Optimization **66**(1), 21–34 (2016)
12. Gutjahr, W.J., Nolz, P.C.: Multicriteria optimization in humanitarian aid. European Journal of Operational Research **252**(2), 351–366 (2016)
13. Hartman, P.: On functions representable as a difference of convex functions. Pacific Journal of Mathematics **9**(3), 707–713 (1959)
14. Hiriart-Urruty, J-.B.: Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. In: J. Ponstein (ed.) Convexity and Duality in Optimization, vol. 256, pp. 37–70. Springer, Berlin (1985)
15. Holmberg, K., Tuy, H.: A production-transportation problem with stochastic demand and concave production costs. Mathematical Programming **85**(1), 157–179 (1999)
16. Horst, R., Thoai, N.V.: DC programming: Overview. Journal of Optimization Theory and Applications **103**(1), 1–43 (1999)
17. Ji, Y., Goh, M., De Souza, R.: Proximal point algorithms for multi-criteria optimization with the difference of convex objective functions. Journal of Optimization Theory and Applications **169**(1), 280–289 (2016)
18. Ji, Y., Qu, S.: Proximal point algorithms for vector DC programming with applications to probabilistic lot sizing with service levels. Discrete Dynamics in Nature and Society (2017). DOI 10.1155/2017/5675183
19. Joki, K., Bagirov, A., Karmitsa, N., Mäkelä, M.M.: A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes. Journal of Global Optimization **68**(3), 501–535 (2017)
20. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M., Taheri, S.: Double bundle method for finding Clarke stationary points in nonsmooth DC programming. SIAM Journal on Optimization **28**(2), 1892–1919 (2018)
21. Kiwiel, K.C.: A descent method for nonsmooth convex multiobjective minimization. Large Scale Systems **8**(2), 119–129 (1985)
22. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable optimization. Mathematical Programming **46**(1–3), 105–122 (1990)
23. Le Thi, H.A., Pham Dinh, T.: Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms. Journal of Global Optimization **11**(3), 253–285 (1997)

24. Le Thi, H.A., Pham Dinh, T.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations research **133**(1–4), 23–46 (2005)
25. Mäkelä, M.M., Eronen, V.-P., Karmitsa, N.: On nonsmooth multiobjective optimality conditions with generalized convexities. In: T.M. Rassias, C.A. Floudas, S. Butenko (eds.) Optimization in Science and Engineering, pp. 333–357. Springer (2014)
26. Mäkelä, M.M., Karmitsa, N., Wilppu, O.: Proximal bBundle method for nonsmooth and nonconvex multiobjective optimization. In: T. Tuovinen, S. Repin, P. Neittaanmäki (eds.) Mathematical Modeling and Optimization of Complex Structures, *Computational Methods in Applied Sciences*, vol. 40, pp. 191–204. Springer (2016)
27. Mäkelä, M.M., Neittaanmäki, P.: Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control. World Scientific Publishing Co., Singapore (1992)
28. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
29. Miettinen, K., Mäkelä, M.M.: Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. Optimization **34**(3), 231–246 (1995)
30. Montonen, O., Joki, K.: Bundle-based descent method for nonsmooth multiobjective DC optimization with inequality constraints. Journal of Global Optimization **72**(3), 403–429 (2018)
31. Montonen, O., Karmitsa, N., Mäkelä, M.M.: Multiple subgradient descent bundle method for convex nonsmooth multiobjective optimization. Optimization **67**(1), 139–158 (2018)
32. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to DC programming: Theory, algorithms and applications. Acta Mathematica Vietnamica **22**(1), 289–355 (1997)
33. Poirion, F., Mercier, Q., Désidéri, J.-A.: Descent algorithm for nonsmooth stochastic multiobjective optimization. Computational Optimization and Applications **68**(2), 317–331 (2017)
34. Qu, S., Goh, M., Wu, S.-Y., De Souza, R.: Multiobjective DC programs with infinite convex constraints. Journal of Global Optimization **59**(1), 41–58 (2014)
35. Qu, S., Liu, C., Goh, M., Li, Y., Ji, Y.: Nonsmooth multiobjective programming with quasi-Newton methods. European Journal of Operational Research **235**(3), 503–510 (2014)
36. Rangaiah, G.P.: Multi-Objective Optimization: Techniques and Applications in Chemical Engineering. Advances in process systems engineering. World Scientific, Singapore (2009)
37. Schramm, H., Zowe, J.: A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. SIAM Journal on Optimization **2**(1), 121–152 (1992)
38. Sun, W.Y., Sampaio, R.J.B., Candido, M.A.B.: Proximal point algorithm for minimization of DC functions. Journal of Computational Mathematics **21**(4), 451–462 (2003)
39. Taa, A.: Optimality conditions for vector optimization problems of a difference of convex mappings. Journal of Global Optimization **31**(3), 421–436 (2005)
40. Toland, J.F.: On subdifferential calculus and duality in nonconvex optimization. Mémoires de la Société Mathématique de France **60**, 177–183 (1979)
41. Wang, S.: Algorithms for Multiobjective and Nonsmooth Optimization. In: P. Kleinschmidt, F.J. Radermacher, W. Sweitzer, H. Wildermann (eds.) Methods of Operations Research, 58, pp. 131–142. Athenaum Verlag (1989)