

An approximation algorithm for convex multi-objective programming problems

Matthias Ehrgott · Lizhen Shao · Anita Schöbel

Received: 10 February 2009 / Accepted: 30 July 2010 / Published online: 17 August 2010
© Springer Science+Business Media, LLC. 2010

Abstract In multi-objective convex optimization it is necessary to compute an infinite set of nondominated points. We propose a method for approximating the nondominated set of a multi-objective nonlinear programming problem, where the objective functions and the feasible set are convex. This method is an extension of Benson’s outer approximation algorithm for multi-objective linear programming problems. We prove that this method provides a set of weakly ε -nondominated points. For the case that the objectives and constraints are differentiable, we describe an efficient way to carry out the main step of the algorithm, the construction of a hyperplane separating an exterior point from the feasible set in objective space. We provide examples that show that this cannot always be done in the same way in the case of non-differentiable objectives or constraints.

Keywords Multi-objective optimization · Convex optimization · Approximation algorithm · ε -nondominated point

1 Introduction

In multi-objective optimization, several objective functions have to be minimized. The objectives are usually conflicting so that a feasible solution optimizing all objectives simultaneously does not exist in most cases. Therefore, the purpose of multi-objective optimization is to

M. Ehrgott
Department of Engineering Science, The University of Auckland, Auckland, New Zealand
e-mail: m.ehrgott@auckland.ac.nz

L. Shao (✉)
School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China
e-mail: lshao@ustb.edu.cn

A. Schöbel
Fakultät für Mathematik, Georg-August Universität Göttingen, Göttingen, Germany
e-mail: schoebel@math.uni-goettingen.de

obtain nondominated points rather than optimal values. A nondominated point in objective space is the image of an efficient solution in variable space. An efficient solution of a multi-objective optimization problem is defined as a feasible solution for which an improvement in one objective will always lead to a deterioration in at least one of the other objectives. The set of all nondominated points forms the nondominated set in objective space. It conveys trade-off information to a decision maker who prefers less to more in each objective. For an introduction to multi-objective optimization and surveys of recent developments we refer the reader to [6,8] and [5].

For multi-objective linear programming (MOLP) problems, researchers have developed a variety of methods for generating the efficient set or the nondominated set such as multi-objective simplex methods, interior point methods, and objective space methods. Multi-objective simplex methods and interior point methods work in variable space to find efficient solutions, see the references in [8]. Since the number of objectives of a multi-objective linear programming problem is often much smaller than the number of variables and typically many efficient solutions in decision space are mapped to a single nondominated point in objective space. Benson [3] argues that generating the nondominated set should require less computation than generating the efficient set. Moreover, it is reasonable to assume that a decision maker will choose a solution based on the objective values rather than variable values. Therefore, finding the nondominated set in objective space instead of the efficient set in decision space is more important for the decision maker. Once the decision maker has chosen a nondominated point, it is usually possible to calculate a feasible solution mapping to this point.

Benson has proposed an outer approximation method [3] to find the extended feasible set in objective space (and thereby the nondominated set) of an MOLP. In [15] and [9] some improvements for this algorithm have been suggested.

Although it is theoretically possible to identify the complete nondominated set with the methods mentioned above, finding an exact description of this set often turns out to be practically impossible or at least computationally too expensive (see the examples in [15]). Therefore, many researchers focus on approximating the nondominated set, see [14] for a survey. In the literature, the concept of ε -nondominated points has been suggested as a mathematical way of dealing with approximate nondominated sets to account for modeling limitations or computational inaccuracies.

A multi-objective *nonlinear* programming (MONP) problem is more difficult to solve than an MOLP. Since the problem is nonlinear, the nondominated set is hard to describe exactly. Therefore, we focus on approximating the nondominated set of the MONP. In [15], we have proposed an approximation version of Benson's algorithm to sandwich the extended feasible set of an MOLP with an outer approximation and an inner approximation. The nondominated set of the inner approximation is proved to be a set of weakly ε -nondominated points. In this paper, we extend this algorithm to approximately solve convex multi-objective nonlinear programming problems. We show that the method we propose yields a set of weakly ε -nondominated points of the original MONP.

In Sect. 2 we present some definitions for convex multi-objective programming problems. In Sect. 3 we review the version of Benson's algorithm given in [9] and prove a generalization of one of the results of [3]. Our algorithm for the convex case is given in Sect. 4 along with the proof that it constructs a set of weakly ε -nondominated points. In Sect. 5 we explain details of some steps of the algorithm for the differentiable case and show some illustrative examples. In Sect. 6 we address the non-differentiable case. We provide two examples that show that it cannot be guaranteed that a hyperplane separating a given extreme point of the outer approximation from the feasible set can always be found if objectives or constraints are not differentiable. We draw some conclusions in Sect. 7.

2 Convex multi-objective programming problem

Let $y^1, y^2 \in \mathbb{R}^p$. In this paper we use the notation $y^1 \leqq y^2$ to indicate $y_i^1 \leqq y_i^2$ for all $i = 1, \dots, p$. Moreover, $y^1 \leq y^2$ indicates $y^1 \leqq y^2$ but $y^1 \neq y^2$ whereas $y^1 < y^2$ means $y_i^1 < y_i^2$ for all $i = 1, \dots, p$.

Let $\mathcal{A} \subseteq \mathbb{R}^p$. We denote the interior of \mathcal{A} by $\text{int } \mathcal{A}$, the convex hull of \mathcal{A} by $\text{conv } \mathcal{A}$ and the boundary of \mathcal{A} by $\text{bd } \mathcal{A}$. Let $\mathbb{R}_{\geq}^p = \{x \in \mathbb{R}^p : x_i \geq 0, i = 1, \dots, p\}$. A point $y \in \mathcal{A}$ is called weakly nondominated if $(\{y\} - \text{int } \mathbb{R}_{\geq}^p) \cap \mathcal{A} = \emptyset$. It is called nondominated if $(\{y\} - \mathbb{R}_{\geq}^p \setminus \{0\}) \cap \mathcal{A} = \emptyset$. The set of (weakly) nondominated points of \mathcal{A} is given by

$$\begin{aligned} \mathcal{A}_{\text{WN}} &:= \left\{ y \in \mathcal{A} : (\{y\} - \text{int } \mathbb{R}_{\geq}^p) \cap \mathcal{A} = \emptyset \right\}, \\ \mathcal{A}_N &:= \left\{ y \in \mathcal{A} : (\{y\} - \mathbb{R}_{\geq}^p) \setminus \{0\} \cap \mathcal{A} = \emptyset \right\}. \end{aligned}$$

Consider a multi-objective programming problem (MOP)

$$\begin{aligned} \min f(x) &= (f_1(x), \dots, f_p(x))^T \\ \text{s. t. } x \in \mathcal{X} &= \{x \in \mathbb{R}^n : g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T \leqq 0\}. \end{aligned} \tag{1}$$

\mathcal{X} is the feasible set in decision space \mathbb{R}^n , we assume it is nonempty. Feasible solutions $x \in \mathcal{X}$ such that $f(x)$ is a (weakly) nondominated point of $f(\mathcal{X})$ are frequently called (weakly) efficient solutions of (1) in the literature. The feasible set \mathcal{Y} in objective space \mathbb{R}^p is defined by

$$\mathcal{Y} = \{f(x) : x \in \mathcal{X}\}. \tag{2}$$

If $x^1, x^2 \in \mathcal{X}$ and $f(x^1) \leq f(x^2)$ we say x^1 dominates x^2 and $f(x^1)$ dominates $f(x^2)$. If $x^1, x^2 \in \mathcal{X}$ and $f(x^1) < f(x^2)$ we say x^1 strictly dominates x^2 and $f(x^1)$ strictly dominates $f(x^2)$. We denote the set of active constraints at a point $x^*, x^* \in \mathcal{X}$ by

$$J(x^*) = \{j \in \{1, \dots, m\} : g_j(x^*) = 0\}.$$

Definition 2.1 The multi-objective programming problem (1) is convex if all its objective functions are convex and its feasible set \mathcal{X} is convex.

Throughout the paper, we only consider convex multi-objective nonlinear programming problems. More precisely, we require that $f_i(x), i = 1, \dots, p$ and $g_j(x), j = 1, \dots, m$ are convex functions. This ensures that both \mathcal{X} and $\mathcal{Y} + \mathbb{R}_{\geq}^p$ are convex.

We also assume that there exists $\hat{y} \in \mathbb{R}^p$ such that $\hat{y} \leqq f(x)$, for all $x \in \mathcal{X}$ (this means that there exists an optimal solution for each single objective problem, or, equivalently, the ideal point exists).

It is well known [10] that weakly and properly efficient solutions of convex multi-objective programming problems are characterised as optimal solutions to weighted sum problems. Properly efficient solutions are efficient solutions with bounded trade-off.

Theorem 2.2 A feasible solution \hat{x} of the convex multi-objective programme (1) is weakly (properly) efficient if and only if there exists some $\lambda \in \mathbb{R}^p, \lambda \geq 0 (\lambda > 0)$ such that \hat{x} is an optimal solution of the single objective convex programme

$$\min\{\lambda^T f(x) : g(x) \leqq 0\}.$$

However, only in the case of multi-objective *linear* programmes is it possible to use this result to find the sets \mathcal{Y}_N and \mathcal{Y}_{WN} (properly efficient and efficient solutions coincide in the linear case), because a finite set of weight vectors λ is sufficient.

In the case of nonlinear convex problems Theorem 2.2 can in practice only be used to find a finite set of efficient solutions (nondominated points). Hence it is necessary to *approximate* the (weakly) nondominated set. Ruzika and Wiecek [14] provide a good survey of approximation methods for multi-objective programming problems.

In this paper we use the concept of (weakly) ε -efficient solutions and (weakly) ε -nondominated points to measure the quality of approximations. In the literature, additive and multiplicative definitions have been given.

The multiplicative definition states that, given a scalar $\epsilon > 0$, $y \in \mathcal{Y}$ is ϵ -nondominated if there is no other $\hat{y} \in \mathcal{Y}$ such that $(1 + \epsilon)\hat{y} \leq y$ (assuming, without loss of generality, that $y \geq 0$ for all $y \in \mathcal{Y}$). Papadimitriou and Yannakakis [13] have shown that for any multi-objective programming problem and any scalar $\epsilon > 0$ there exists a polynomial size ϵ -nondominated set.

Throughout this paper we adopt the additive definition of ε -nondominated point. Let $\varepsilon \in \mathbb{R}_{\leq}^p$. A point $y \in \mathcal{Y}$ is called (weakly) ε -nondominated if there does not exist any $\hat{y} \in \mathcal{A}$ such that $\hat{y} \leq (<) y - \varepsilon$. Feasible solutions $x \in \mathcal{X}$ such that $f(x)$ is a (weakly) ε -nondominated point of $f(\mathcal{X})$ are frequently called (weakly) ε -efficient solutions of (1).

In the next section we review Benson’s outer approximation algorithm [2,3], which has been shown to construct a set of ε -nondominated points for multi-objective linear programmes in [9].

3 Benson’s algorithm

Benson has proposed an outer approximation algorithm (see [2,3]) to solve MOLPs in objective space, i.e.,

$$f(x) = Cx, g(x) = b - Ax, \tag{3}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $C \in \mathbb{R}^{p \times n}$. The algorithm is further developed in [9]. We describe this version of the algorithm.

We use

$$\mathcal{P} = \{y \in \mathbb{R}^p : Cx \leq y \text{ for some } x \in \mathcal{X}\} = \mathcal{Y} + \mathbb{R}_{\leq}^p, \tag{4}$$

which we call the extended feasible set in objective space of the MOP (1).

According to the following theorem, instead of directly finding \mathcal{Y}_N , the version of Ehrgott et al. [9] is dedicated to finding \mathcal{P}_{WN} .

Theorem 3.1 *The following statements hold.*

1. *The set $\mathcal{P} \subset \mathbb{R}^p$ is a nonempty polyhedron of dimension p and \mathcal{P} is \mathbb{R}_{\leq}^p -bounded from below.*
2. *$\mathcal{Y}_N = \mathcal{P}_N$.*
3. *Every vertex of \mathcal{P} belongs to \mathcal{Y}_N .*
4. *$\mathcal{P}_{WN} = \text{bd } \mathcal{P}$.*

Proofs of 1., 2., and 3. can be found in [9] and [3]. To see 4. notice that $\mathcal{P}_{WN} \subset \text{bd } \mathcal{P}$ is always true and let $y \in \text{bd } \mathcal{P}$. If there is $y' \in \mathcal{P}$ with $y' < y$ it follows that $y = y' + d$ for some $d \in \text{int } \mathbb{R}_{\geq}^p$ contradicting $y \in \text{bd } \mathcal{P}$.

Theorem 3.1 is the basis of Ehrgott et al.’s version of Benson’s outer approximation algorithm [9]. It works on \mathcal{P} to find all nondominated extreme points of \mathcal{Y} . The idea of the algorithm can be summarized as follows. First, a polyhedron that contains \mathcal{P} is constructed and an interior point $\hat{p} \in \mathcal{P}$ is found. Then, for each vertex of the polyhedron, the algorithm checks whether it is in \mathcal{P} or not. If not, it constructs a hyperplane separating the vertex from \mathcal{P} and updates the polyhedron until all its vertices are contained in \mathcal{P} .

In order to construct the cut separating a point $s \notin \mathcal{P}$ from \mathcal{P} , Benson suggests the following procedure: The point s and \hat{p} are connected by a line segment. This line segment intersects the boundary of \mathcal{P} at a unique point y . The idea is to construct a hyperplane which supports \mathcal{P} in y and hence separates s and \mathcal{P} . The following primal dual pair $P(y)$ and $D(y)$ of linear programmes depending on $y \in \mathbb{R}^p$ is needed for that purpose.

$$\begin{aligned}
 P(y) \quad & \min\{z : Ax \geq b, Cx - ez \leq y\}, \\
 D(y) \quad & \max\{b^T u - y^T \lambda : A^T u - C^T \lambda = 0; e^T \lambda = 1; u, \lambda \geq 0\}.
 \end{aligned}$$

Note that the objective value of $P(y)$ is strictly greater than zero if $y \notin \mathcal{P}$, is equal to zero if $y \in \mathcal{P}_{WN}$ and is strictly less than zero if $y \in \mathcal{P} \setminus \mathcal{P}_{WN}$. Furthermore, Benson [3] has shown that an optimal solution of $D(y)$ can be used to obtain the required supporting hyperplane if y is a weakly nondominated point of \mathcal{P} .

Proposition 3.2 [2,3] *Let $y^* \in \mathcal{P}_{WN}$ and let (u^*, λ^*) be an optimal solution of $D(y^*)$. Then the following holds:*

- (i) $y^{*T} \lambda^* = b^T u^*$,
- (ii) $y^T \lambda^* \geq b^T u^*$ for all $y \in \mathcal{P}$,

i.e. $H := \{y \in \mathbb{R}^p : y^T \lambda^ = b^T u^*\}$ is a supporting hyperplane of \mathcal{P} at y^* .*

From Proposition 3.2, we know that y^* is a weakly nondominated point of \mathcal{P} if and only if the optimal values of $P(y^*)$ and $D(y^*)$ are both equal to zero. The algorithm can now be stated.

Algorithm 3.3 (Benson’s outer approximation algorithm)

Initialization.

- (i1) Set $S^0 = y^I + \mathbb{R}_{\geq}^p (y_i^I = \min\{y_i : y \in \mathcal{Y}\})$.
- (i2) $\text{vert } S^0 = \{y^I\}$. Store the inequality representation of S^0 .
- (i3) Compute a point $\hat{p} \in \text{int } \mathcal{P}$, set $k = 0$ and go to iteration k .

Iteration steps.

- (k1) If, for each $s \in \text{vert } S^k, s \in \mathcal{P}$, then go to Step k5: $\mathcal{P} = S^k$. Otherwise, choose any $s^k \in \text{vert } S^k$ such that $s^k \notin \mathcal{P}$ and continue.
- (k2) Find the unique value ρ_k of $\rho, 0 < \rho < 1$, such that $\rho s^k + (1 - \rho)\hat{p} \in \text{bd } \mathcal{P}$ and set $y^k = \rho_k s^k + (1 - \rho_k)\hat{p}$.
- (k3) Set $S^{k+1} = S^k \cap \{y \in \mathbb{R}^p : \lambda^{kT} y \geq b^T u^k\}$, where (u^k, λ^k) can be found by solving LP $D(y^k)$ (see Proposition 3.2).
- (k4) Using $\text{vert } S^k$ and the definition of S^{k+1} given in Step (k3), determine $\text{vert } S^{k+1}$. Set $k = k + 1$ and go to iteration (k1).
- (k5) The nondominated extreme points of \mathcal{P} are $\mathcal{P}_{NE} := \text{vert } S^k$. $\mathcal{Y}_{NE} := \mathcal{P}_{NE}$ is the set of all nondominated extreme points of \mathcal{Y} . Stop.

Benson [3] proves that the algorithm is finite and it terminates with finding all the non-dominated extreme points of \mathcal{Y} in step (k5).

We conclude this section by presenting an extension of Proposition 3.2 which will be helpful for Sect. 6. It shows that we can start with any point \hat{y} (it need not be weakly efficient and even need not be inside \mathcal{P}) and the dual $D(\hat{y})$ will yield a supporting hyperplane for \mathcal{P} .

Proposition 3.4 *Let $\hat{y} \in \mathbb{R}^p$ and let (u^*, λ^*) be an optimal solution of $D(\hat{y})$. Then*

- (i) $y^T \lambda^* \geq b^T u^*$ for all $y \in \mathcal{P}$.
- (ii) $y^T \lambda^* = b^T u^*$ for $y = Cx'$, where x' is an optimal solution of $P(\hat{y})$,

i.e. the hyperplane $H := \{y \in \mathbb{R}^p : y^T \lambda^ = b^T u^*\}$ is a supporting hyperplane of \mathcal{P} at Cx' .*

Proof (i) Let (u^*, λ^*) be an optimal solution of $D(\hat{y})$ and assume to the contrary that $\tilde{y}^T \lambda^* < b^T u^*$ for some $\tilde{y} \in \mathcal{P}$. Consider

$$D(\tilde{y}) \quad \max\{b^T u - \tilde{y}^T \lambda : A^T u - C^T \lambda = 0; e^T \lambda = 1; u, \lambda \geq 0\}.$$

Since (u^*, λ^*) is feasible for $D(\hat{y})$ it is also feasible for $D(\tilde{y})$ and hence $D(\tilde{y})$ has a feasible solution with objective value strictly greater than zero. The dual to $D(\tilde{y})$ is

$$P(\tilde{y}) \quad \min\{z : Ax \geq b, Cx - ez \leq \tilde{y}\}.$$

From weak duality we conclude that every feasible solution of $P(\tilde{y})$ has objective value strictly greater than zero. On the other hand, $\tilde{y} \in \mathcal{P}$. Hence, there exists $\tilde{x} \in \mathbb{R}^n$ such that $A\tilde{x} \geq b$ and $C\tilde{x} \leq \tilde{y}$. We obtain that $x := \tilde{x}, z := 0$ is a feasible solution to $P(\tilde{y})$ with objective value equal to zero, a contradiction.

- (ii) Let (x', z') be an optimal solution of $P(\hat{y})$. We want to show that $(Cx')^T \lambda^* = b^T u^*$. To this end, we calculate

$$\begin{aligned} (Cx')^T \lambda^* &= \underbrace{(\lambda^*)^T}_{\geq 0} Cx' \\ &\leq (\lambda^*)^T (\hat{y} + ez') \text{ since } (x', z') \text{ is feasible for } P(\hat{y}) \\ &= (\lambda^*)^T \hat{y} + z' \underbrace{(\lambda^*)^T e}_{=1} = (\lambda^*)^T \hat{y} + z' \\ &= b^T u^* \text{ due to strong duality.} \end{aligned}$$

□

4 An extension of Benson’s algorithm to convex MOPs

We extend Benson’s algorithm to solve convex multi-objective programming problems. We assume that the objective functions $f_i(x), i = 1, \dots, p$ and the constraint functions $g_j(x), j = 1, \dots, m$ are convex.

Our extended Benson’s algorithm uses the same idea as the approximation version of Benson’s algorithm in [15], i.e., we approximate the set

$$\mathcal{P} = \{y \in \mathbb{R}^p : f(x) \leq y \text{ for some } x \in \mathcal{X}\}. \tag{5}$$

with an outer approximation and an inner approximation.

We first remark that most of the results of Theorem 3.1 also hold for the convex case because their proofs do not use the linearity.

Proposition 4.1 1. *The set $\mathcal{P} \subset \mathbb{R}^p$ is a non-empty set of dimension p and \mathcal{P} is \mathbb{R}^p_{\geq} -bounded from below.*

2. $\mathcal{Y}_N = \mathcal{P}_N$.
3. $\mathcal{P}_{WN} = \text{bd } \mathcal{P}$.

Let $\epsilon > 0$ be an approximation error. Our algorithm starts with a polyhedron \mathcal{S} , covering the set \mathcal{P} . Then an interior point \hat{p} of \mathcal{P} is found. In each iteration, we check vertices s of \mathcal{S} to see if they are contained in \mathcal{P} or not. If not, we calculate the corresponding boundary point y of \mathcal{P} on the line segment between s and \hat{p} , and then measure the distance $d(s, y)$ between s and y . If the vertex $s \in \mathcal{P}$ or if $d(s, y) \leq \epsilon$, we add s to the outer approximation \mathcal{O} and its corresponding boundary point y to the inner approximation \mathcal{I} (note: if $s \in \mathcal{P}$, its corresponding boundary point is s itself). On the other hand, if a vertex of the polyhedron \mathcal{S} is found at a distance greater than ϵ from its corresponding boundary point, a hyperplane separating it from \mathcal{P} is added to the description of \mathcal{S} and the procedure is repeated.

We now state the algorithm in its general form.

Algorithm 4.2 (Outer approximation algorithm for convex MOP)

Initialization.

- (i1) Set $\mathcal{S}^0 = y^I + \mathbb{R}_{\geq}^p (y_i^I = \min\{y_i : y \in \mathcal{Y}\})$, then $\text{vert } \mathcal{S}^0 = \{y^I\}$.
- (i2) Find an interior point \hat{p} of \mathcal{P} .
- (i3) Let $\mathcal{O} := \emptyset, \mathcal{I} := \emptyset$ and $k = 0$.

Iteration steps.

- (k1) If, for each $s \in \text{vert } \mathcal{S}^k, s \in \mathcal{P} \cup \mathcal{O}$ is satisfied, then go to Step (k6). Otherwise, choose any $s^k \in \text{vert } \mathcal{S}^k \setminus (\mathcal{O} \cup \mathcal{P})$ and continue.
- (k2) Compute the unique boundary point $y^k := \rho_k s^k + (1 - \rho_k)\hat{p} \in \mathcal{P}$, where $\rho_k \in (0, 1)$.
- (k3) If the distance $d(s^k, y^k)$ from s^k to y^k is at most ϵ , then add s^k to \mathcal{O} and add y^k to \mathcal{I} . Go to (k1).
- (k4) If the distance $d(s^k, y^k) > \epsilon$, then determine a hyperplane $H := \{y \in \mathbb{R}^p : y^T \lambda^k \geq b^T u^k\}$ separating s^k from \mathcal{P} and set $\mathcal{S}^{k+1} = \mathcal{S}^k \cap \{y \in \mathbb{R}^p : y^T \lambda^k \geq b^T u^k\}$.
- (k5) Determine $\text{vert } \mathcal{S}^{k+1}$. Set $k = k + 1$ and go to (k1).
- (k6) Define the set of points of the outer approximation $V_o(\mathcal{S}^k) = \text{vert } \mathcal{S}^k$ and define the set of points of the inner approximation $V_i(\mathcal{S}^k) = (\text{vert } \mathcal{S}^k \setminus \mathcal{O}) \cup \mathcal{I}$.

Results.

- (r1) Let $\mathcal{P}^i = \text{conv } V_i(\mathcal{S}^K) + \mathbb{R}_{\geq}^p$. \mathcal{P}^i represents the inner approximation of \mathcal{P} ($\mathcal{P}^i \subseteq \mathcal{P}$).
- (r2) Let $\mathcal{P}^o = \text{conv } V_o(\mathcal{S}^K) + \mathbb{R}_{\geq}^p$. \mathcal{P}^o represents the outer approximation of \mathcal{P} ($\mathcal{P} \subseteq \mathcal{P}^o$).

Theorem 4.3 Let $\epsilon = \epsilon e, e = (1, \dots, 1)^T \in \mathbb{R}^p$. Then \mathcal{P}_N^i is a set of weakly ϵ -nondominated points of \mathcal{P} .

Proof Shao and Ehrgott [15] have proved this for the approximation version of Benson’s algorithm to solve MOLPs. The proof also holds for our approximation algorithm to solve convex MONPs. The general idea of the proof is described as follows: Due to convexity we obtain that $\mathcal{P}^i \subseteq \mathcal{P} \subseteq \mathcal{P}^o$. Since the vertices of the two polyhedra \mathcal{P}^i and \mathcal{P}^o have a one to one corresponding relationship and the distances between all corresponding vertices are at most ϵ , the result follows. For details, the reader is referred to [15]. □

The execution of Steps **(k2)** (finding the unique boundary point) and **(k4)** (finding the separating hyperplane) is different from the approximation version of Benson’s algorithm [15] since the problem is nonlinear. Step **(k2)** can be done by solving the following nonlinear programme

$$\rho^k := \max\{\rho : g(x) \leq 0, 0 < \rho < 1, \rho s^k + (1 - \rho)\hat{p} \geq f(x), x \in \mathbb{R}^n, \rho \in \mathbb{R}\} \quad (6)$$

and setting $y^k := \rho^k s^k + (1 - \rho^k)\hat{p}$.

Note that ρ^k exists due to the intermediate value theorem since $f_i(x), i = 1, \dots, p$ are continuous. Next we prove some results about optimal solutions of (6).

Proposition 4.4 *Let $\hat{p} \in \text{int}\mathcal{P}$ and $s^k \notin \mathcal{P}$. Let (x^k, ρ^k) be an optimal solution of the nonlinear programme (6). Then x^k is a weakly efficient solution of MOP (1).*

Proof Suppose, to the contrary, that there exists $\bar{x}, g(\bar{x}) \leq 0$ such that $f(\bar{x}) < f(x^k) \leq y^k = \rho^k s^k + (1 - \rho^k)\hat{p}$. Then we can increase ρ^k until $(\rho^k + \delta)s^k + (1 - \rho^k - \delta)\hat{p}$ and $f(\bar{x})$ have at least one component the same. This contradicts the assumption that ρ^k is the optimal value of the nonlinear programme (6). □

Proposition 4.5 *Let $\hat{p} \in \text{int}\mathcal{P}$ and $s^k \notin \mathcal{P}$. Let (x^k, ρ^k) be an optimal solution of the nonlinear programme (6) and let $y^k = \rho^k s^k + (1 - \rho^k)\hat{p}$. Then $y^k \in \mathcal{P}_{\text{WN}}$.*

Proof Clearly, $y^k \in \text{bd } \mathcal{P}$. The result follows from Proposition 4.1. □

Theorem 4.3 shows that, given any scalar $\epsilon > 0$, Algorithm 4.2 constructs a set of weakly $\epsilon\epsilon$ -nondominated points for the multi-objective convex programming problem (1). The guarantee of the approximation quality is possible because the algorithm constructs both an outer and an inner approximation of \mathcal{P} , i.e. $\mathcal{P}^i \subseteq \mathcal{P} \subseteq \mathcal{P}^o$. In the process, the algorithm finds a finite number of nondominated points y^k .

One may ask how this compares to computing a finite number of nondominated points using a scalarization method such as the weighted sum method (as indicated at the end of Sect. 2) or the elastic constraint method (see, e.g. [7]). Due to convexity of \mathcal{P} , these nondominated points can be used to construct an inner approximation of \mathcal{P} in the same way as Algorithm 3.3. However, it would not be possible to assure the quality of the approximation, i.e. whether the nondominated set of the inner approximation is a set of weakly ϵ -nondominated points, due to the lack of an outer approximation to compare with.

5 Constructing a separating hyperplane in the differentiable case

We now additionally assume that the convex objective functions $f_i(x), i = 1, \dots, p$ and the convex constraint functions $g_j(x), j = 1, \dots, m$ are continuously differentiable and that the problem satisfies the Kuhn-Tucker constraint qualification at all weakly efficient solutions.

Under these assumptions we will present an efficient way of performing **(k3)**, i.e., of constructing a hyperplane cutting off s from \mathcal{P} .

The idea of constructing a hyperplane is to use the dual programme as in the linear case. Therefore, based on the optimal solution x^k of the nonlinear programme (6) and its corresponding point $f(x^k)$ we approximate the multi-objective nonlinear programme by an MOLP at the fixed point x^k . We then construct a hyperplane that supports \mathcal{P} in y^k .

Let

$$\nabla f(x^k) = (\nabla f_1(x^k), \dots, \nabla f_p(x^k)) = \begin{pmatrix} \frac{\partial f_1(x^k)}{\partial x_1}, \dots, \frac{\partial f_p(x^k)}{\partial x_1} \\ \vdots \\ \frac{\partial f_1(x^k)}{\partial x_n}, \dots, \frac{\partial f_p(x^k)}{\partial x_n} \end{pmatrix}$$

and

$$\nabla g(x^k) = (\nabla g_1(x^k), \dots, \nabla g_m(x^k)) = \begin{pmatrix} \frac{\partial g_1(x^k)}{\partial x_1}, \dots, \frac{\partial g_m(x^k)}{\partial x_1} \\ \vdots \\ \frac{\partial g_1(x^k)}{\partial x_n}, \dots, \frac{\partial g_m(x^k)}{\partial x_n} \end{pmatrix}$$

be the gradient of $g(x)$ and of $f(x)$ at $x^k \in \mathbb{R}^n$ respectively. We define the following MOLP

$$\begin{aligned} \text{MOLP}(x^k) \quad & \min f(x^k) + \nabla f(x^k)^T(x - x^k) \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^T(x - x^k) \leq 0. \end{aligned}$$

Given x^k let us denote

$$\mathcal{P}^L(x^k) = \{y \in \mathbb{R}^p : f(x^k) + \nabla f(x^k)^T(x - x^k) \leq y \text{ for some } x \text{ with } g(x^k) + \nabla g(x^k)^T(x - x^k) \leq 0\}.$$

If it is clear to which point x^k we refer we will only write \mathcal{P}^L . Note that

$$\mathcal{P}^L(x^k) \supseteq \mathcal{P} \tag{7}$$

since $f_i(x), i = 1, \dots, p$ and $g_j(x), j = 1, \dots, m$ are convex functions.

We furthermore define

$$\begin{aligned} f_{x^k} &:= f(x^k) - \nabla f(x^k)^T x^k, \\ C_{x^k} &:= \nabla f(x^k)^T, \\ A_{x^k} &:= -\nabla g(x^k)^T, \\ b_{x^k} &:= g(x^k) - \nabla g(x^k)^T x^k, \end{aligned} \tag{8}$$

and rewrite MOLP(x^k) as

$$\min\{C_{x^k}x + f_{x^k} : A_{x^k}x \geq b_{x^k}\}.$$

Apart from the constant f_{x^k} this is a linear multi-objective programme as in Sect. 3. The primal and dual pair that we use to construct the supporting hyperplane for MOLP(x^k) are

$$\begin{aligned} \text{LP}(x^k, y^k) \quad & \min\{z : A_{x^k}x \geq b_{x^k}, C_{x^k}x - ez \leq y^k - f_{x^k}\}, \\ \text{LD}(x^k, y^k) \quad & \max\{b_{x^k}^T u - (y^k - f_{x^k})^T \lambda : A_{x^k}^T u - C_{x^k}^T \lambda = 0; e^T \lambda = 1; u, \lambda \geq 0\}. \end{aligned}$$

According to Proposition 3.2 we know: In case that $y^k \in \mathcal{P}_{\text{WN}}^L(x^k)$ is a weakly nondominated point for MOLP(x^k), an optimal solution (u^*, λ^*) of LD(x^k, y^k) will yield a supporting hyperplane $H = \{y \in \mathbb{R}^p : (y - f_{x^k})^T \lambda^* = b_{x^k}^T u^*\}$ of $\mathcal{P}^L(x^k)$ at y^k . This hyperplane corresponds to a hyperplane $H' = \{y \in \mathbb{R}^p : y^T \lambda^* = f(x^k)^T \lambda^*\}$ which we will show to be supporting \mathcal{P} at y^k in case of differentiable constraints and objectives. We hence can use the following procedure to perform Step (k4) in Algorithm 4.2:

Algorithm 5.1 (Step (k4) for differentiable functions)

| | |
|-------------|--|
| (k4) | Determine gradients $\nabla f(x^k)$ and $\nabla g(x^k)$. Solve $LD(x^k, y^k)$ and let (u^k, λ^k) be an optimal solution. Set $S^{k+1} = S^k \cap \{y \in \mathbb{R}^p : y^T \lambda^k \geq b^T u^k\}$. |
|-------------|--|

Let (x^k, ρ^k) be an optimal solution of (6) and let $y^k = \rho^k s^k + (1 - \rho^k) \hat{p}$. In the following we show step by step that solving the dual problem $LD(x^k, y^k)$ will give us a supporting hyperplane of \mathcal{P} at y^k (Theorem 5.6). We have already shown that $f(x^k), y^k \in \mathcal{P}_{WN}$ in Propositions 4.4 and 4.5.

Next we show the main intuition, namely that $f(x^k)$ is contained in $\mathcal{P}_{WN}^L(x^k)$. To do this we need Proposition 5.2.

Proposition 5.2 *Let $x^k \in \mathcal{X}$ be a weakly efficient solution of MOP (1) and let MOP (1) satisfy the Kuhn-Tucker constraint qualification at x^k . Then there does not exist any $d \in \mathbb{R}^p$ such that*

$$\begin{aligned} \nabla f_i(x^k)^T d &< 0, \quad \text{for all } i = 1, \dots, p \text{ and} \\ \nabla g_j(x^k)^T d &\leq 0, \quad \text{for all } j \in J(x^k). \end{aligned}$$

Proof See [12] for a proof. □

This result is useful for the following Proposition.

Proposition 5.3 *Let $x^k \in \mathcal{X}$ be a weakly efficient solution of MOP (1) and let MOP (1) satisfy the Kuhn-Tucker constraint qualification at x^k . Then x^k is a weakly efficient solution of MOLP(x^k).* □

Proof Suppose, to the contrary, that x^k is strictly dominated by \bar{x} . Therefore we have

$$\begin{aligned} f_i(x^k) + \nabla f_i(x^k)^T (\bar{x} - x^k) &< f_i(x^k), \quad \text{for all } i = 1, \dots, p \text{ and} \\ g_j(x^k) + \nabla g_j(x^k)^T (\bar{x} - x^k) &\leq 0, \quad \text{for all } j = 1, \dots, m. \end{aligned}$$

This means

$$\begin{aligned} \nabla f_i(x^k)^T (\bar{x} - x^k) &< 0, \quad \text{for all } i = 1, \dots, p \text{ and} \\ \nabla g_j(x^k)^T (\bar{x} - x^k) &\leq 0, \quad \text{for all } j \in J(x^k). \end{aligned}$$

This contradicts Proposition 5.2. Therefore, x^k is weakly efficient for MOLP(x^k). □

Proposition 5.3 can also be explained by the Karush-Kuhn-Tucker (KKT) necessary conditions for weakly efficient solutions (Theorem 3.1.5 in [12]). Since x^k is a weakly efficient solution of MOP (1), according to the KKT necessary conditions, there exist vectors $0 \leq \lambda \in \mathbb{R}^p$ and $0 \leq \mu \in \mathbb{R}^m$ such that

$$\begin{aligned} \sum_{i=1}^p \lambda_i \nabla f_i(x^k) + \sum_{j=1}^m \mu_j \nabla g_j(x^k) &= 0 \\ \mu_j g_j(x^k) &= 0 \quad \text{for all } j = 1, \dots, m. \end{aligned}$$

Therefore the optimal value of $LD(x^k, f(x^k))$ is equal to zero, which means that $f(x^k)$ is a weakly nondominated point of $\mathcal{P}^L(x^k)$.

This result shows that in the case of $f(x^k) = y^k$ the supporting hyperplane obtained from solving $LD(x^k, y^k)$ is also a supporting hyperplane of \mathcal{P} at y^k . However, there may

exist weakly efficient solutions with $f(x^k) \neq y^k$. To be able to prove our main result (Theorem 5.6) we show in Proposition 5.5 that y^k is contained in $\mathcal{P}_{\text{WN}}^L(x^k)$ even when $f(x^k) \neq y^k$.

Proposition 5.4 *Let $\bar{y}, f(\bar{x}) \in \mathcal{P}_{\text{WN}}$ such that $\bar{y} \geq f(\bar{x})$. Then there exists $\lambda \geq 0$, $\lambda^T \bar{y} = \lambda^T f(\bar{x})$ such that \bar{x} is an optimal solution of the weighted sum problem $\min\{\lambda^T f(x), g(x) \leq 0\}$.*

Proof Since $f(\bar{x}), \bar{y} \in \mathcal{P}_{\text{WN}}$, there exist $\lambda_1 \geq 0, \lambda_2 \geq 0$ such that

$$\begin{aligned} \lambda_1^T f(\bar{x}) &= c_1 \lambda_1^T y \geq c_1, \text{ for all } y \in \mathcal{P} \\ \lambda_2^T \bar{y} &= c_2 \lambda_2^T y \geq c_2, \text{ for all } y \in \mathcal{P}. \quad (*) \end{aligned}$$

Let $1 > \alpha > 0$. For any convex combination of $\lambda_1^T y \geq c_1$ and $\lambda_2^T y \geq c_2$ we have

$$\alpha \lambda_1^T y + (1 - \alpha) \lambda_2^T y \geq \alpha c_1 + (1 - \alpha) c_2, \text{ for all } y \in \mathcal{P}.$$

Substituting $c_1 = \lambda_1^T f(\bar{x})$ and $c_2 = \lambda_2^T \bar{y}$ into the above inequality, we have

$$\alpha \lambda_1^T y + (1 - \alpha) \lambda_2^T y \geq \alpha \lambda_1^T f(\bar{x}) + (1 - \alpha) \lambda_2^T \bar{y}, \text{ for all } y \in \mathcal{P}.$$

Because $f(\bar{x}) \in \mathcal{P}$, we have

$$\alpha \lambda_1^T f(\bar{x}) + (1 - \alpha) \lambda_2^T f(\bar{x}) \geq \alpha \lambda_1^T f(\bar{x}) + (1 - \alpha) \lambda_2^T \bar{y},$$

i.e.,

$$(1 - \alpha) \lambda_2^T (f(\bar{x}) - \bar{y}) \geq 0.$$

Thus $\lambda_2^T (f(\bar{x}) - \bar{y}) = 0$ as $\bar{y} \geq f(\bar{x})$. This means there exists $\lambda_2 \geq 0, \lambda_2^T (f(\bar{x}) - \bar{y}) = 0$ such that \bar{x} is an optimal solution of the weighted sum problem $\min\{\lambda_2^T f(x), g(x) \leq 0\}$. Otherwise we have a contradiction to (*). \square

Proposition 5.5 *If $f(\bar{x}), \bar{y} \in \mathcal{P}_{\text{WN}}$, that $\bar{y} \geq f(\bar{x})$ and that the constraint qualification holds at \bar{x} . Then $\bar{y} \in \mathcal{P}_{\text{WN}}^L(\bar{x})$.*

Proof Since $\bar{y}, f(\bar{x}) \in \mathcal{P}_{\text{WN}}$, according to Proposition 5.4, there exists $\lambda \geq 0, \lambda^T \bar{y} = \lambda^T f(\bar{x})$ such that \bar{x} is an optimal solution of the weighted sum problem $\min\{\lambda^T f(x), g(x) \leq 0\}$.

Define

$$\begin{aligned} F_0 &= \{d : \sum_{i=1}^p \lambda_i \nabla f_i(\bar{x})^T d < 0\} \quad \text{and} \\ G'_0 &= \{d : \nabla g_j(\bar{x})^T d \leq 0 \text{ for each } j \in J(\bar{x})\}. \end{aligned}$$

From the constraint qualification it follows that \bar{x} is a KKT solution of the weighted sum problem. According to Theorem 4.2.15 in [1] this implies that $F_0 \cap G'_0 = \emptyset$.

We denote $I = \{i : \bar{y}_i = f_i(\bar{x})\}$ (remember that $\bar{y} \geq f(\bar{x})$). Now suppose to the contrary that \bar{y} is strictly dominated by $f(\bar{x}) + \nabla f(\bar{x})^T (x^* - \bar{x})$ for MOLP(\bar{x}). Then

$$\begin{aligned} f_i(\bar{x}) + \nabla f_i(\bar{x})^T (x^* - \bar{x}) &< \bar{y}_i, \quad i = 1, \dots, p, \\ g_j(\bar{x}) + \nabla g_j(\bar{x})^T (x^* - \bar{x}) &\leq 0, \quad j = 1, \dots, m. \end{aligned}$$

i.e.,

$$\nabla f_i(\bar{x})^T (x^* - \bar{x}) < 0, \text{ for all } i \in I \text{ and } \nabla g_j(\bar{x})^T (x^* - \bar{x}) \leq 0, \text{ for all } j \in J(\bar{x}).$$

Now note that $I \neq \emptyset$ (otherwise y would not be weakly nondominated). Furthermore, there exists $i \in I$ such that $\lambda_i > 0$ since $\lambda \geq 0$, $\lambda^T(\bar{y} - f(\bar{x})) = 0$. We hence obtain that

$$\sum_{i=1}^p \lambda_i \nabla f_i(\bar{x})^T (x^* - \bar{x}) = \sum_{i \in I} \lambda_i \nabla f_i(\bar{x})^T (x^* - \bar{x}) < 0$$

Using $d = x^* - \bar{x}$ this contradicts $F_0 \cap G'_0 = \emptyset$. □

The fact that \bar{y} is a weakly nondominated point of $\mathcal{P}^L(\bar{x})$ can also be explained by the KKT necessary conditions of nonlinear programming (see [1]). Since \bar{x} is an optimal solution of the weighted sum problem $\min\{\lambda^T f(x), g(x) \leq 0\}$ ($\lambda \geq 0$, $\lambda^T \bar{y} = \lambda^T f(\bar{x})$), the KKT necessary conditions imply that there exists a vector $0 \leq \mu \in \mathbb{R}^m$ such that

$$\begin{aligned} \sum_{i=1}^p \lambda_i \nabla f_i(\bar{x}) + \sum_{j=1}^m \mu_j \nabla g_j(\bar{x}) &= 0 \\ \mu_j g_j(\bar{x}) &= 0 \quad \text{for all } j = 1, \dots, m \end{aligned}$$

This indicates that the optimal value of $LD(\bar{x}, \bar{y})$ is equal to zero, which means that \bar{y} is a weakly nondominated point of $\mathcal{P}^L(\bar{x})$.

Finally we can show that solving $LD(x^k, y^k)$ yields a supporting hyperplane of \mathcal{P} .

Theorem 5.6 *Let $\hat{p} \in \text{int}\mathcal{P}$ and $s^k \notin \mathcal{P}$. Let x^k, ρ^k be an optimal solution of nonlinear programme (6) and let $y^k = \rho^k s^k + (1 - \rho^k)\hat{p}$. Let H be the hyperplane obtained by solving $LD(x^k, y^k)$. Then H is a supporting hyperplane of \mathcal{P} at y^k .*

Proof We have $y^k \in \mathcal{P}_{\text{WN}}$ due to Proposition 4.5 and $f(x^k) \in \mathcal{P}_{\text{WN}}$ due to Proposition 4.4. By (6) it is clear that $f(x^k) \leq y^k$. We hence can apply Proposition 5.5 and obtain that $y^k \in \mathcal{P}_{\text{WN}}^L(x^k)$. (Note that in the case $f(x^k) = y^k$, Proposition 5.3 would have been sufficient.) From Proposition 3.2 we hence know that H is a supporting hyperplane of $\mathcal{P}^L(x^k)$ at the point y^k . Since $y^k \in \mathcal{P}$ and $\mathcal{P}^L(x^k) \supseteq \mathcal{P}$ (see (7)) we obtain that H is also a supporting hyperplane of \mathcal{P} at y^k and hence cuts off s^k from \mathcal{P} . □

Illustrative examples

First we give an example to illustrate the extension of Benson’s algorithm.

Example 5.7 Consider the following MONP problem.

$$\begin{aligned} \min \quad & (f_1(x), f_2(x)) & &= (x_1, x_2) \\ \text{s.t.} \quad & (x_1 - 2)^2 + (x_2 - 2)^2 & \leq & 4 \\ & 0 & \leq & x_1 \\ & 0 & \leq & x_2 \end{aligned}$$

Here, $\mathcal{P} = \{y \in \mathbb{R}^2 : f(x) \leq y, g(x) \leq 0\}$, $y^I = (0, 0)$. Figure 1 shows $\mathcal{Y}, \mathcal{P}, S^0$ and the interior point \hat{p} . We set the approximation error $\epsilon = 0.1$.

Figures 2, 3, 4 show the first, second and third cut. The vertex s^k being cut off, its corresponding boundary point y^k , the distance $d(s^k, y^k)$ between s^k and y^k and the cutting plane for each iteration are listed in Table 1. After three cuts, there are four points outside \mathcal{P} , they are (0,1.6155), (0.2950,0.8766), (1.6155,0), and (0.8766,0.2950). Their corresponding boundary points are (0.0343,1.6313), (0.3280,0.9025), (1.6313,0.0343) and (0.9025,0.3280).

The distances between the four infeasible points and the corresponding boundary points are less than the approximation error 0.1, so we accept these four infeasible points for the outer approximation. The algorithm terminates.

We want to compare the results with the method of finding a finite set of nondominated points by the weighted sum and elastic constraint methods. Since in Example 5.7 the inner approximation of Benson’s algorithm is constructed with four nondominated points, we will find four nondominated points with the two other methods as well.

The weighted sum problem is

$$\min\{\lambda x_1 + (1 - \lambda)x_2 : (x_1 - 2)^2 + (x_2 - 2)^2 \leq 4; x_1 \geq 0; x_2 \geq 0\}$$

and we use $\lambda = 1, \lambda = \frac{2}{3}, \lambda = \frac{1}{3}$ and $\lambda = 0$. The elastic constraint method is

$$\min\{x_2 + 2s_1 : x_1 - s_1 \leq \mu; (x_1 - 2)^2 + (x_2 - 2)^2 \leq 4; x_1 \geq 0; x_2 \geq 0; s_1 \geq 0\}$$

and we use $\mu = 0, \mu = \frac{1}{3}, \mu = \frac{5}{3}$, and $\mu = 2$.

Table 2 shows the resulting nondominated points for all three methods and Fig. 5 illustrates the resulting inner approximations of \mathcal{P} .

From Theorem 4.3, the quality of the approximation of Benson’s algorithm is guaranteed, it is a $0.1e$ -weakly nondominated set of \mathcal{P} . In this example, however, both the weighted sum and the elastic constraint method deliver worse approximations, in fact they are not $0.1e$ -weakly nondominated sets of \mathcal{P} . This indicates that the values chosen as weights

Fig. 1 $\mathcal{Y}, \mathcal{P}, S^0$ and \hat{p}

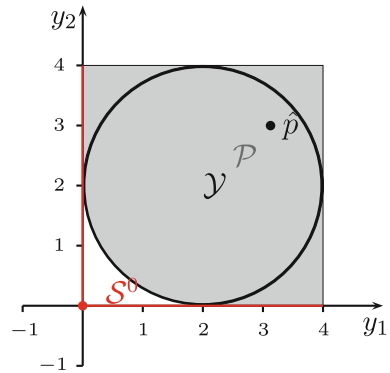


Fig. 2 First cut

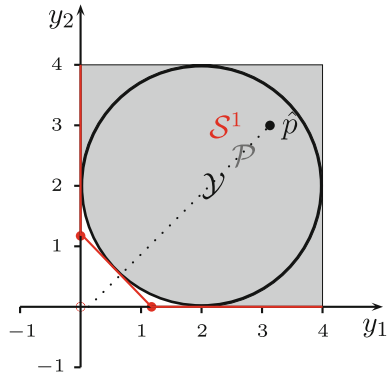


Fig. 3 Second cut

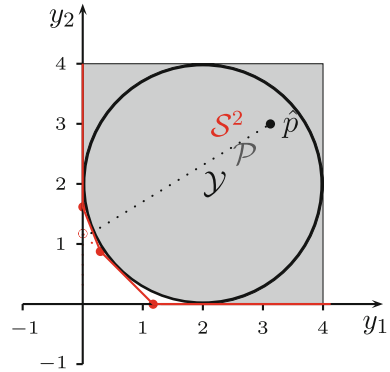


Fig. 4 Third cut

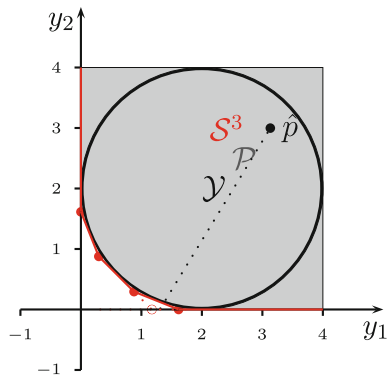


Table 1 $s^k, y^k, d(s^k, y^k)$ and the cutting planes in the first three iterations of the algorithm

| Iteration | s^k | y^k | $d(s^k, y^k)$ | Cut |
|-----------|------------|-----------------|---------------|-----------------------------------|
| $k = 0$ | (0,0) | (0.5858,0.5858) | 0.8284 | $y_1 + y_2 = 1.1716$ |
| $k = 1$ | (0,1.1776) | (0.1426,1.2585) | 0.1669 | $0.7147y_1 + 0.2853y_2 = 0.4609$ |
| $k = 2$ | (1.1716,0) | (1.2585,0.1426) | 0.1669 | $0.2853 y_1 + 0.7147y_2 = 0.4609$ |

Table 2 Nondominated points found by Benson’s algorithm, the weighted sum method, and the elastic constraint method

| Method | y^1 | y^2 | y^3 | y^4 |
|--------------------|-----------------|-----------------|-----------------|-----------------|
| Benson | (0.0343,1.6313) | (0.3280,0.9025) | (0.9025,0.3280) | (1.6313,0.0343) |
| Weighted sum | (0,2) | (0.2111,1.1056) | (1.1056,0.2111) | (2,0) |
| Elastic constraint | (0.2111,1.1056) | (0.3333,0.8945) | (1.6667,0.0280) | (2,0) |

or as right hand sides in the two methods are not appropriate to find $0.1e$ -weakly non-dominated sets of \mathcal{P} . Hence the methods need to be extended by mechanisms to control the parameters λ and μ in such a way that the quality of the approximation can be guaranteed.

In order to perform some preliminary computational experiments, we have implemented our algorithm in Matlab 7.3 using MOSEK as NP and LP solver. At Step (k4), the method of [4] for on-line vertex enumeration by adjacency lists was used to calculate a vertex

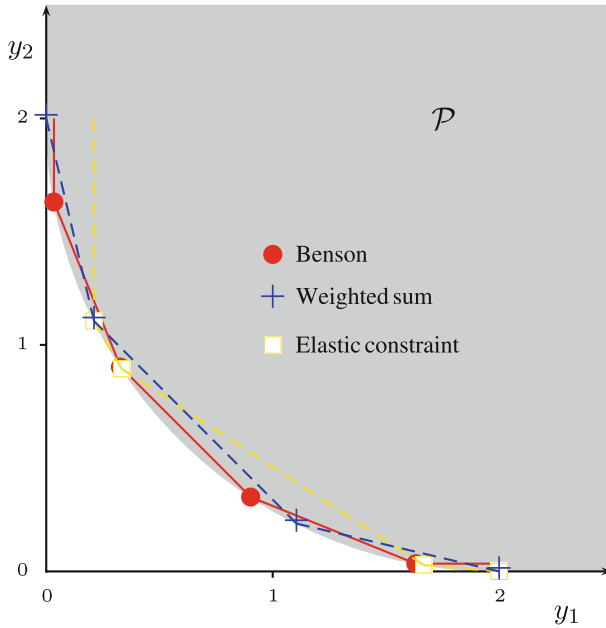


Fig. 5 The inner approximations constructed from the nondominated points of the three methods

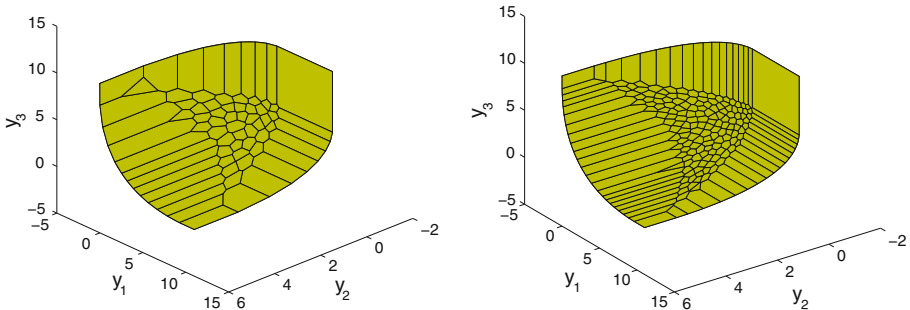


Fig. 6 \mathcal{P}^o with approximation errors 0.05 and 0.01

representation from the inequality representation of \mathcal{S}^k . We then applied this code to three convex multi-objective nonlinear programmes listed in the test examples in [11]. The tests were run on a dual processor CPU with 1.8 GHz and 1 GB RAM.

Example 5.8

$$\begin{aligned}
 \min \quad & f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 \\
 & f_2(x) = (x_1 - 2)^2 + (x_2 - 3)^2 \\
 & f_3(x) = (x_1 - 4)^2 + (x_2 - 2)^2 \\
 \text{s.t.} \quad & x_1 + 2x_2 \leq 10 \\
 & 0 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 4
 \end{aligned}$$

Figure 6 shows the result of solving the problem with approximation errors 0.05 and 0.01, respectively.

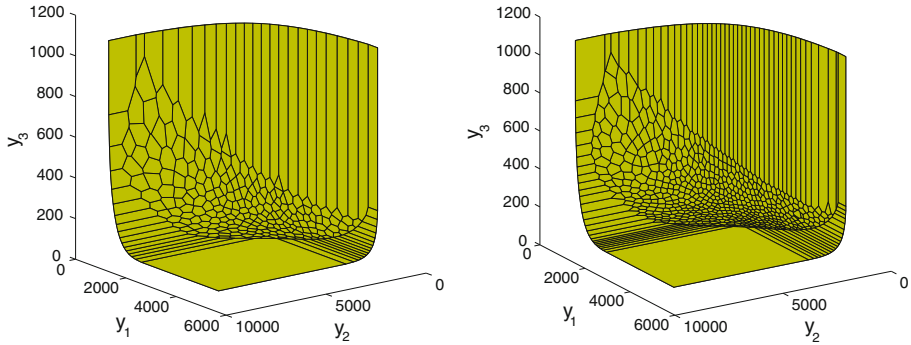


Fig. 7 \mathcal{P}^o with approximation errors 5 and 2

Example 5.9

$$\begin{aligned}
 \min \quad & f_1(x) && = 50x_1^4 + 10x_2^4 \\
 & f_2(x) && = 30(x_1 - 5)^4 + 100(x_2 - 3)^4 \\
 & f_3(x) && = 70(x_1 - 2)^4 + 20(x_2 - 4)^4 \\
 \text{s.t.} \quad & (x_1 - 2)^2 + (x_2 - 2)^2 && \leq 1 \\
 & 0 && \leq x_1 \leq 3 \\
 & 0 && \leq x_2 \leq 3
 \end{aligned}$$

Figure 7 shows the result of solving the problem with approximation errors 5 and 2, respectively.

Example 5.10

$$\begin{aligned}
 \min \quad & f_1(x) && = x_1^2 + x_2^2 + x_3^2 + 10x_2 - 120x_3 \\
 & f_2(x) && = x_1^2 + x_2^2 + x_3^2 + 80x_1 - 448x_2 + 80x_3 \\
 & f_3(x) && = x_1^2 + x_2^2 + x_3^2 - 448x_1 + 80x_2 + 80x_3 \\
 \text{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 && \leq 100 \\
 & 0 && \leq x_1 \leq 10 \\
 & 0 && \leq x_2 \leq 10 \\
 & 0 && \leq x_3 \leq 10
 \end{aligned}$$

Figure 8 shows the result of solving the problem with approximation errors 10 and 5, respectively.

Summarizing information comparing the number of vertices, the number of cutting planes and the computation time with various values of ϵ is given in Table 3.

Table 3 and the figures clearly show the effect of the choice of ϵ . The smaller the error parameter, the more cutting planes and the more vertices are generated and the longer the computation time.

Note that we chose different approximation errors for the different examples. This is because the approximation error is an absolute value, hence it should be chosen according to the scale of the objectives.

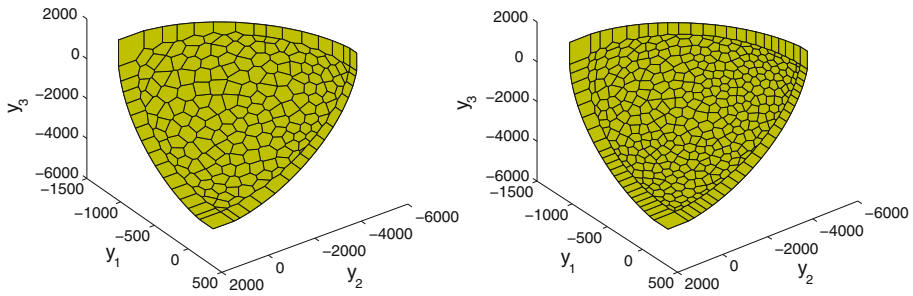


Fig. 8 \mathcal{P}^o with approximation errors 10 and 5

Table 3 Running time and number of vertices and cutting planes for the three examples with different values of ϵ

| Example | ϵ | Time (seconds) | Vertices of \mathcal{P}^o | Cutting planes |
|---------|------------|----------------|-----------------------------|----------------|
| 5.8 | 0.05 | 3.562 | 110 | 69 |
| | 0.01 | 7.859 | 346 | 201 |
| 5.9 | 5 | 41.344 | 479 | 267 |
| | 2 | 113.626 | 1170 | 625 |
| 5.10 | 10 | 25.860 | 391 | 218 |
| | 5 | 37.610 | 748 | 407 |

6 Approaches for the non-differentiable case

If the functions involved are convex but not differentiable, we can still use Algorithm 4.2 since all the results of Sect. 4 remain true. The problem is how to perform Step (k4) in the non-differentiable case.

At a first glance one might think that replacing each of the gradients by a subgradient might allow to use the construction of a separating hyperplane of the previous section also for the non-differentiable (convex) case. Unfortunately, this is not that easy as the following two examples demonstrate.

Example 6.1 Consider the following (MOP) with differentiable constraints but non-differentiable objective functions:

$$\begin{aligned} \min \quad & f_1(x) = |x_1| + |x_2| \\ & f_2(x) = |x_1 - 2| + |x_2| \\ \text{s.t.} \quad & x_1^2 + x_2^2 - 100 \leq 0 \end{aligned}$$

The feasible solution $\hat{x} = (1, 0)$ with $f(\hat{x}) = (1, 1)$ is efficient. We linearize at \hat{x} . To this end we determine the subgradients of the two objective functions

$$\begin{aligned} \nabla f_1(\hat{x}) &= (1, \alpha)^T \quad \text{with } -1 \leq \alpha \leq 1 \\ \nabla f_2(\hat{x}) &= (-1, \beta)^T \quad \text{with } -1 \leq \beta \leq 1 \end{aligned}$$

and $\nabla g(\hat{x}) = (2, 0)^T$, $g(\hat{x}) = -99$. The resulting MOLP(\hat{x}) hence is

$$\begin{aligned} \min \quad & \bar{f}_1(x) &= x_1 + \alpha x_2 \\ & \bar{f}_2(x) &= -x_1 + \beta x_2 + 2 \\ \text{s.t.} \quad & 2x_1 - 101 &\leq 0. \end{aligned}$$

Considering this linearization we see that $(1, 1)$ is dominated since choosing

$$\begin{aligned} x_1 &= \frac{2\alpha}{\alpha + \beta} \\ x_2 &= \frac{-2}{\alpha + \beta} \end{aligned}$$

leads to $\bar{f}(x) = (0, 0) < (1, 1)$ whenever $\alpha + \beta \neq 0$.

Even worse, this example shows that it might be impossible to cut off a point s : Suppose we started Algorithm 4.2 with the interior point $\hat{p} = (2, 2)$ and at some iteration we want to cut off $s = (0, 0) \notin \mathcal{P}$. Then $y = f(\hat{x})$ is the boundary point obtained in Step (k2) of Algorithm 4.2. But as shown above, the linearization at point $y = (1, 1)$ will in most cases contain s such that it cannot be cut off.

In the second example we consider an MOP with differentiable objective functions and a non-differentiable constraint. We again demonstrate that it can happen that the linearization $\mathcal{P}^L(x^k)$ contains the point s^k such that it is not possible to find a separating hyperplane by looking at the corresponding MOLP(x^k).

Example 6.2

$$\begin{aligned} \min \quad & f_1(x) &= (x_1 - 3)^2 + (x_2 - 1)^2 \\ & f_2(x) &= (x_1 - 1)^2 + (x_2 - 1)^2 \\ \text{s.t.} \quad & |x_1| + 2|x_2| - 2 &\leq 0 \end{aligned}$$

Let $s = (6 - 4\sqrt{2}, 2) \notin \mathcal{P}$ and $\hat{p} = (10, 2) \in \text{int } \mathcal{P}$. Step (k2) of Algorithm 4.2 then yields the point $y = (2, 2)$ as the unique point on the line through s and \hat{p} which is on the boundary of \mathcal{P} . The corresponding feasible solution \hat{x} which is obtained when solving (6) is $\hat{x} = (2, 0)$ with $f(\hat{x}) = (2, 2) = y$. Linearizing (MOP) at \hat{x} (using the gradients at \hat{x} for the objective function and a subgradient for the constraint) yields the MOLP(\hat{x}) given as

$$\begin{aligned} \min \quad & \bar{f}_1(x) &= -2x_1 - 2x_2 + 6 \\ & \bar{f}_2(x) &= 2x_1 - 2x_2 - 2 \\ \text{s.t.} \quad & x_1 + 2\alpha x_2 - 2 &\leq 0 \end{aligned}$$

with some α satisfying $0 \leq \alpha \leq 1$.

Consider $\tilde{x} = (\sqrt{2} + 1, \sqrt{2} - 1)$ and note that \tilde{x} is feasible for MOLP(\hat{x}) if $\alpha \leq -\frac{1}{2}$. But

$$\bar{f}(\tilde{x}) = (6 - 4\sqrt{2}, 2) = s,$$

hence $s \in \mathcal{P}^L(\hat{x})$. Consequently, there cannot exist a hyperplane separating $\mathcal{P}^L(\hat{x})$ from s . This means, whenever $\alpha \leq -\frac{1}{2}$, the linearization MOLP(\hat{x}) is useless for Step (k4) of the algorithm.

The two examples show that we cannot guarantee that for any linearization a separating hyperplane can be obtained. We nevertheless proceed similar to the approach in Sect. 5 as follows:

Let $\nabla g(x)$ and $\nabla f(x)$ be subgradients at $x^k \in \mathbb{R}^n$ for g and f , respectively. We linearize at x^k by using the subgradients and obtain the following MOLP

$$\begin{aligned} \text{MOLP}(x^k) \quad & \min f(x^k) + \nabla f(x^k)^T(x - x^k) \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^T(x - x^k) \leq 0. \end{aligned}$$

Using the same notation as in (8) we consider the following pair of dual linear programmes, but instead of y^k we use the point s^k we want to cut off:

$$\begin{aligned} \text{LP}(x^k, s^k) \quad & \min\{z : A_{x^k}x \geq b_{x^k}, C_{x^k}x - ez \leq s^k - f_{x^k}\}, \\ \text{LD}(x^k, s^k) \quad & \max\{b_{x^k}^T u - (s^k - f_{x^k})^T \lambda : A_{x^k}^T u - C_{x^k}^T \lambda = 0\}. \end{aligned}$$

Applying Proposition 3.4 with $\hat{y} = s^k$ yields the following corollary:

Corollary 6.3 *Let (u^*, λ^*) be an optimal solution of $\text{LD}(x^k, s^k)$ and (x^*, z^*) be an optimal solution of $\text{LP}(x^k, s^k)$. Then*

- (i) $(y - f_{x^k})^T \lambda^* \geq b_{x^k}^T u^*$ for all $y \in \mathcal{P}^L(x^k)$,
- (ii) $(y - f_{x^k})^T \lambda^* = b_{x^k}^T u^*$ for $y = Cx' + f_{x^k}$ and x' is an optimal solution of $\text{LP}(x^k, s^k)$,
- (iii) $(s^k - f_{x^k})^T \lambda^* \geq b_{x^k}^T u^*$ if and only if $z^* > 0$ if and only if $s^k \notin \mathcal{P}^L(x^k)$.

I.e., the hyperplane $H := \{y \in \mathbb{R}^p : (y - f_{x^k})^T \lambda^ = b_{x^k}^T u^*\}$ is a supporting hyperplane of \mathcal{P} at $Cx^* + f_{x^k}$. It separates s^k from $\mathcal{P}^L(x^k)$ if and only if the objective value of $\text{LD}(x^k, s^k)$ (or equivalently, of $\text{LP}(x^k, s^k)$) is strictly greater than 0.*

Proof (i) and (ii) are a direct consequence of Proposition 3.4. For (iii), recall that the objective value of $\text{LP}(x^k, s^k)$ is strictly greater than zero if and only if $s^k \notin \mathcal{P}^L(x^k)$. □

This gives us on the one hand the possibility to efficiently test if s^k is cut off or not, namely by checking if the objective value of $\text{LD}(x^k, s^k)$ is greater than zero or negative. On the other hand, since the objective function of $\text{LD}(x^k, s^k)$ is to maximize $b_{x^k}^T u^* - (s^k - f_{x^k})^T \lambda^*$ we will end up with a separating hyperplane if there exists one for the respective linearization.

Note that if a separating hyperplane is found by solving the dual $\text{LD}(x^k, y^k)$ it is a supporting hyperplane for $\mathcal{P}^L(x^k)$ (so in some sense the best we can expect) but need not be supporting for \mathcal{P} . Summarizing, we suggest the following Step (k4) in our approximation algorithm 4.2 for the non-differentiable case:

Algorithm 6.4 (Step (k4) for non-differentiable functions)

-
- (k4) (a) Determine subgradients $\nabla f(x^k)$ and $\nabla g(x^k)$.
 - (b) Solve $\text{LD}(x^k, s^k)$, let (u^k, λ^k) be an optimal solution.
 - (c) If the objective value of $\text{LD}(x^k, s^k)$ is smaller than zero then set $S^{k+1} = S^k \cap \{y \in \mathbb{R}^p : y^T \lambda^k \geq b^T u^k\}$.
 - (d) Otherwise choose new subgradients $\nabla f(x^k)$ and $\nabla g(x^k)$ and go back to (k4) (b) or choose other vertex s^k and go back to (k2).
-

We hence can perform Algorithm 4.2 using the specifications for (k4) as in Sect. 5 but checking in each step if the hyperplane H found cuts off s^k or not. If it cuts off s^k we use it and proceed as in the differentiable case. Otherwise we have to change the subgradient or the vertex s^k and continue.

7 Conclusion

In this paper, we have extended Benson's outer approximation method to solve convex multi-objective nonlinear programming problems. We have assumed that the objective functions and the constraint functions are continuously differentiable and proved that the algorithm guarantees to find weakly ε -nondominated points with a specified approximation error ϵ ($\varepsilon = \epsilon e$). We have tested the algorithm with three small examples. The results suggest that different approximation error should be chosen for different problems according to the objective scales since the approximation error is an absolute value. Further research is needed to investigate the trade-off between approximation quality and computational effort. It is clear that as ε decreases, the number of iterations will increase, but it is worth investigating if a bound on the number of iterations can be obtained.

If the problem is not differentiable, then the boundary point may not be weakly non-dominated and we cannot use the dual programme of the MOLP to find the supporting hyperplane. Therefore, it is not obvious how a separating hyperplane can be found efficiently and a heuristic procedure is suggested. Further evaluating the theory and behaviour of the extension to non-differentiable convex multi-objective nonlinear programming problems is under research.

References

1. Bazaraa, M.S.: *Nonlinear Programming Theory and Algorithms*. John Wiley and Sons Inc, (1993)
2. Benson, H.P.: Hybrid approach for solving multiple-objective linear programs in outcome space. *J. Optim. Theory Appl.* **98**, 17–35 (1998)
3. Benson, H.P.: An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *J. Global Optim.* **13**, 1–24 (1998)
4. Chen, P.C., Hansen, P., Jaumard, B.: On-line and off-line vertex enumeration by adjacency lists. *Oper. Res. Lett.* **10**, 403–409 (1991)
5. Chinchuluun, A., Pardalos, M.: A survey of recent developments in multiobjective optimization. *Ann. Oper. Res.* **154**, 29–50 (2007)
6. Ehrgott, M.: *Multicriteria Optimization*. Springer, (2005)
7. Ehrgott, M., Ruzika, S.: Improved ε -constraint method for multiobjective programming. *J. Optim. Theory Appl.* **138**(3), 375–396 (2008)
8. Ehrgott, M., Wiecek, M.: Multiobjective programming. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) *Multicriteria Decision Analysis: State of the Art Surveys*, pp. 667–722. Springer Science + Business Media, New York (2005)
9. Ehrgott, M., Löhne, A., Shao, L.: A dual variant of Benson's outer approximation algorithm. Report 654, Department of Engineering Science, The University of Auckland (2007)
10. Geoffrion, A.: Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.* **22**, 618–630 (1968)
11. Miettinen, K., Makela, M.M., Kaario, K.: Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *Eur. J. Oper. Res.* **175**, 931–947 (2006)
12. Miettinen, K.M.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, (1999)
13. Papadimitriou, C., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: Blum, A. (ed) *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pp. 86–92. IEEE Computer Society (2000)
14. Ruzika, S., Wiecek, M.M.: Approximation methods in multiobjective programming. *J. Optim. Theory Appl.* **126**(3), 473–501 (2005)
15. Shao, L., Ehrgott, M.: Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Math. Methods Oper. Res.* **68**, 257–276 (2008). doi:[10.1007/s00186-008-0220-2](https://doi.org/10.1007/s00186-008-0220-2)