

COMPUTING MINIMAL ELEMENTS OF FINITE FAMILIES OF SETS W.R.T. PREORDER RELATIONS IN SET OPTIMIZATION

CHRISTIAN GÜNTHER¹, ELISABETH KÖBIS^{1,*}, NICOLAE POPOVICI²

¹*Martin Luther University Halle-Wittenberg, Faculty of Natural Sciences II,
Institute for Mathematics, 06099 Halle (Saale), Germany*

²*Babeş-Bolyai University, Faculty of Mathematics and Computer Science, 400084 Cluj-Napoca, Romania*

Abstract. We propose new algorithms for computing all minimal elements of a nonempty finite family of sets in a real linear space, with respect to a preorder relation defined on the power set of that space. These algorithms are based on a set-valued counterpart of the well-known Graef-Younes reduction procedure, originally conceived for vector optimization. One of our algorithms consists of two subsequent (forward-backward) reduction procedures, similarly to the classical Jahn-Graef-Younes method. Another algorithm involves a pre-sorting procedure with respect to a strongly increasing real-valued function, followed by a single (forward) reduction procedure. Numerical experiments in MATLAB allow us to compare our algorithms for special test families of line segments with respect to ℓ -type, u -type and s -type preorder relations, currently used in set optimization.

Keywords. Preorder relation; Minimal element; External stability; Graef-Younes reduction method; Sorting scalar function.

2010 Mathematics Subject Classification. 26B25, 46A40, 90C29.

1. INTRODUCTION

Set optimization has become an active field of research due to its wide applications (see, e.g., Khan, Tammer and Zălinescu [13] and the references therein). Deriving efficient methods for solving set optimization problems is particularly important, and so far several interesting methods have been developed for computing the minimal elements or some approximations of minimal elements (see, e.g., Jahn [10], and Köbis and Köbis [15]).

The aim of this paper is to develop new algorithms for computing all minimal elements of a nonempty finite family of sets $\mathcal{A} \subseteq \mathcal{P}(E)$ with respect to a preorder relation \preceq defined on the power set $\mathcal{P}(E)$ of a real linear space E . These algorithms are inspired from three methods originally conceived for vector optimization problems:

- (i) *Graef-Younes method*, proposed by Younes [20] and formulated algorithmically by Graef, as mentioned by Jahn [7, Section 12.4];
- (ii) *Jahn-Graef-Younes method*, also called *Graef-Younes method with backward iteration*, proposed by Jahn [7, 12], Jahn and Rathje [8], and reformulated in a more general setting by Eichfelder [4];
- (iii) *Graef-Younes method with a pre-sorting procedure*, proposed by Günther and Popovici [6].

*Corresponding author.

E-mail addresses: christian.guenther@mathematik.uni-halle.de (C. Günther), elisabeth.koebis@mathematik.uni-halle.de (E. Köbis), popovici@math.ubbcluj.ro (N. Popovici).

Received June 15, 2019; Accepted August 1, 2019.

Recently, Köbis, Kuroiwa and Tammer [16] have extended (i) to set optimization problems. We recall their method in our paper as Algorithm 1 (*Generalized Graef-Younes method*). It is an iterative reduction procedure, which consists of a forward iteration loop that excludes a part of the elements which are not minimal in \mathcal{A} . The method then returns a family $\mathcal{B} \subseteq \mathcal{A}$, which contains all minimal elements of \mathcal{A} .

Our Algorithm 2 (*Generalized Jahn-Graef-Younes method*) extends (ii) to set optimization. It returns precisely the set of all minimal elements of \mathcal{A} , by performing two subsequent (forward-backward) reduction procedures, each of them being based on Algorithm 1. We mention that a similar method has been studied under different assumptions (namely when \preccurlyeq is antisymmetric and the set of minimal elements of \mathcal{A} is externally stable) by Köbis, Kuroiwa and Tammer [16].

As a natural extension of (iii) to set optimization, our Algorithm 3 (*Generalized Graef-Younes method with pre-sorting*) also generates the set of all minimal elements of \mathcal{A} , this time by pre-sorting the elements of \mathcal{A} with respect to a strongly \preccurlyeq -increasing real-valued function, prior to the reduction procedure based on Algorithm 1.

Before presenting our algorithms, we will provide in Section 2 some preliminary notions and results on preorder relations defined on a family of sets in a real linear space. Strongly monotone scalar functions with respect to the preorder relations, as well as their role in generating minimal elements of a family of sets, are emphasized in Section 3. Then, in the main Section 4, we formulate our algorithms and we prove their correctness. A comparative analysis of the new algorithms through numerical experiments is provided in Section 5 for a special class of sets and appropriate sorting functions. This paper concludes with a brief summary and an outlook to future work in Section 6.

2. PRELIMINARIES

Throughout this paper E denotes a real linear space, whose power set $\mathcal{P}(E)$ is endowed with a pre-order relation \preccurlyeq , which means that

- \preccurlyeq is reflexive (i.e., $\forall A \in \mathcal{P}(E) : A \preccurlyeq A$) and
- \preccurlyeq is transitive (i.e., $\forall A, A', A'' \in \mathcal{P}(E) : A' \preccurlyeq A \text{ and } A'' \preccurlyeq A' \implies A'' \preccurlyeq A$).

Moreover, $\mathcal{A} \subseteq \mathcal{P}(E)$ will denote a nonempty family of sets.

Definition 2.1. We say that $A \in \mathcal{A}$ is a minimal element of \mathcal{A} (w.r.t. \preccurlyeq) if

$$\forall A' \in \mathcal{A} : A' \preccurlyeq A \implies A \preccurlyeq A'.$$

We denote by $\text{MIN}_{\preccurlyeq}(\mathcal{A})$ the set of all minimal elements of \mathcal{A} , i.e.,

$$\begin{aligned} \text{MIN}_{\preccurlyeq}(\mathcal{A}) &= \{A \in \mathcal{A} \mid \forall A' \in \mathcal{A} : A' \preccurlyeq A \implies A \preccurlyeq A'\} \\ &= \{A \in \mathcal{A} \mid \nexists A' \in \mathcal{A} : A' \preccurlyeq A \text{ and } A \not\preccurlyeq A'\}. \end{aligned}$$

Remark 2.1. If \preccurlyeq is antisymmetric on \mathcal{A} , i.e.,

$$\forall A, A' \in \mathcal{A} : A' \preccurlyeq A \text{ and } A \preccurlyeq A' \implies A = A',$$

then it is easy to check that

$$\begin{aligned} \text{MIN}_{\preccurlyeq}(\mathcal{A}) &= \{A \in \mathcal{A} \mid \forall A' \in \mathcal{A} : A' \preccurlyeq A \implies A' = A\} \\ &= \{A \in \mathcal{A} \mid \nexists A' \in \mathcal{A} \setminus \{A\} : A' \preccurlyeq A\}. \end{aligned}$$

Example 2.1. As usual in set optimization theory and related topics (see, e.g., Jahn and Ha [9], Khan, Tammer and Zălinescu [13], Kuroiwa [17], and Seto, Kuroiwa and Popovici [19]), any convex cone $C \subseteq E$ (i.e., $0 \in C = \mathbb{R}_+ \cdot C = C + C$) induces on $\mathcal{P}(E)$ three preorder relations, \leq_C^ℓ , \leq_C^u and \leq_C^s , defined for any $A, A' \in \mathcal{P}(E)$ by

$$\begin{aligned} A' \leq_C^\ell A &: \iff A \subseteq A' + C \iff A + C \subseteq A' + C; \\ A' \leq_C^u A &: \iff A' \subseteq A - C \iff A' - C \subseteq A - C; \\ A' \leq_C^s A &: \iff A' \leq_C^\ell A \text{ and } A' \leq_C^u A \iff A + C \subseteq A' + C \text{ and } A' - C \subseteq A - C. \end{aligned}$$

Letting \leq_C^ℓ , \leq_C^u and \leq_C^s in the role of \preceq it follows by Definition 2.1 that

$$\begin{aligned} \text{MIN}_{\leq_C^\ell}(\mathcal{A}) &= \{A \in \mathcal{A} \mid \forall A' \in \mathcal{A} : A \subseteq A' + C \implies A' \subseteq A + C\} \\ &= \{A \in \mathcal{A} \mid \nexists A' \in \mathcal{A} : A + C \subseteq A' + C \text{ and } A + C \neq A' + C\}; \\ \text{MIN}_{\leq_C^u}(\mathcal{A}) &= \{A \in \mathcal{A} \mid \forall A' \in \mathcal{A} : A' \subseteq A - C \implies A \subseteq A' - C\} \\ &= \{A \in \mathcal{A} \mid \nexists A' \in \mathcal{A} : A' - C \subseteq A - C \text{ and } A' - C \neq A - C\}; \\ \text{MIN}_{\leq_C^s}(\mathcal{A}) &= \{A \in \mathcal{A} \mid \forall A' \in \mathcal{A} : (A \subseteq A' + C \text{ and } A' \subseteq A - C) \\ &\implies (A' \subseteq A + C \text{ and } A \subseteq A' - C)\} \\ &= \{A \in \mathcal{A} \mid \nexists A' \in \mathcal{A} : A + C \subseteq A' + C, A' - C \subseteq A - C \text{ and} \\ &\quad (A + C \neq A' + C \text{ or } A' - C \neq A - C)\}. \end{aligned}$$

To illustrate these concepts, let us consider the particular framework where $E = \mathbb{R}^2$ is endowed with the standard ordering cone $C = \mathbb{R}_+^2$ and let $\mathcal{A}^* = \{A^1, \dots, A^5\}$ be the family of line segments parallel to the second bisector represented in Figure 1. It is a simple exercise to check that

$$\text{MIN}_{\leq_C^\ell}(\mathcal{A}^*) = \{A^1, A^2\}, \text{MIN}_{\leq_C^u}(\mathcal{A}^*) = \{A^1, A^4\} \text{ and } \text{MIN}_{\leq_C^s}(\mathcal{A}^*) = \{A^1, A^2, A^4, A^5\}.$$

Definition 2.2. We say that a subfamily $\mathcal{B} \subseteq \mathcal{A}$ is externally stable (in \mathcal{A} w.r.t. \preceq) if

$$\forall A \in \mathcal{A} \setminus \mathcal{B}, \exists B \in \mathcal{B} : B \preceq A.$$

Remark 2.2. It is well-known that every nonempty finite subset of a general preordered set is externally stable (see, e.g., Podinovskiĭ and Nogin [18, pp. 21]). Thus, whenever \mathcal{A} is nonempty and finite, the set $\text{MIN}_{\preceq}(\mathcal{A})$ is externally stable.

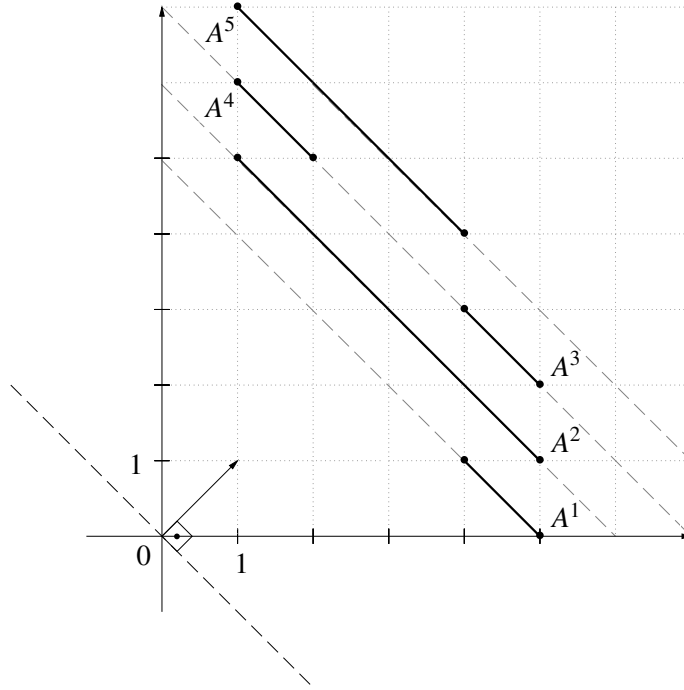
The following lemma will play an important role in the next sections. It extends a result given by Günther and Popovici [6, Lemma 2.8] from vector optimization to set optimization.

Lemma 2.1. Assume that \mathcal{A} is finite. Then, for any nonempty family $\mathcal{B} \subseteq \mathcal{A}$ we have

$$\mathcal{B} \cap \text{MIN}_{\preceq}(\mathcal{A}) \subseteq \text{MIN}_{\preceq}(\mathcal{B}) \subseteq \mathcal{B} \tag{2.1}$$

and the following equivalence holds true:

$$\text{MIN}_{\preceq}(\mathcal{A}) = \text{MIN}_{\preceq}(\mathcal{B}) \iff \text{MIN}_{\preceq}(\mathcal{A}) \subseteq \mathcal{B}. \tag{2.2}$$

FIGURE 1. The family \mathcal{A}^* in Example 2.1.

Proof. The inclusions in (2.1) follow by Definition 2.1.

In order to prove the equivalence (2.2), assume first that $\text{MIN}_{\preccurlyeq}(\mathcal{A}) = \text{MIN}_{\preccurlyeq}(\mathcal{B})$. Then, we get

$$\text{MIN}_{\preccurlyeq}(\mathcal{A}) = \text{MIN}_{\preccurlyeq}(\mathcal{B}) \subseteq \mathcal{B}.$$

Now, for showing the reverse implication, assume that $\text{MIN}_{\preccurlyeq}(\mathcal{A}) \subseteq \mathcal{B}$. Notice that

$$\text{MIN}_{\preccurlyeq}(\mathcal{A}) \subseteq \mathcal{B} \cap \text{MIN}_{\preccurlyeq}(\mathcal{A}) \subseteq \text{MIN}_{\preccurlyeq}(\mathcal{B}) \quad (2.3)$$

in view of (2.1). Now, in order to prove the reverse inclusion, $\text{MIN}_{\preccurlyeq}(\mathcal{B}) \subseteq \text{MIN}_{\preccurlyeq}(\mathcal{A})$, consider an arbitrary element

$$B \in \text{MIN}_{\preccurlyeq}(\mathcal{B}) \subseteq \mathcal{B} \subseteq \mathcal{A}.$$

Suppose by contradiction that $B \in \mathcal{A} \setminus \text{MIN}_{\preccurlyeq}(\mathcal{A})$. Then, there exists some $A \in \mathcal{A}$ such that $A \preccurlyeq B$ and $B \not\preccurlyeq A$. Two cases may occur:

Case 1: Let $A \in \text{MIN}_{\preccurlyeq}(\mathcal{A})$. Then, due to (2.3), we have $A \in \text{MIN}_{\preccurlyeq}(\mathcal{B}) \subseteq \mathcal{B}$. The fact that $A \preccurlyeq B$ and $B \not\preccurlyeq A$ for some $A \in \mathcal{B}$ contradicts the choice of $B \in \text{MIN}_{\preccurlyeq}(\mathcal{B})$.

Case 2: Assume that $A \in \mathcal{A} \setminus \text{MIN}_{\preccurlyeq}(\mathcal{A})$. Since $\text{MIN}_{\preccurlyeq}(\mathcal{A})$ is externally stable by Remark 2.2, there exists some $A' \in \text{MIN}_{\preccurlyeq}(\mathcal{A})$ such that $A' \preccurlyeq A$. In view of (2.3), $\text{MIN}_{\preccurlyeq}(\mathcal{A}) \subseteq \text{MIN}_{\preccurlyeq}(\mathcal{B})$, thus we have $A' \in \text{MIN}_{\preccurlyeq}(\mathcal{B})$. On the one hand, we have $B \not\preccurlyeq A'$, because otherwise we would have $B \preccurlyeq A' \preccurlyeq A$, hence $B \preccurlyeq A$ by transitivity of \preccurlyeq , a contradiction. On the other hand, since $A' \in \mathcal{B}$ and $A' \preccurlyeq A \preccurlyeq B$ we have $A' \preccurlyeq B$, by transitivity of \preccurlyeq . Recalling that $B \in \text{MIN}_{\preccurlyeq}(\mathcal{B})$, we get $B \preccurlyeq A'$, a contradiction. \square

3. (STRONGLY) \preccurlyeq -MONOTONE SCALAR FUNCTIONS

Definition 3.1. A scalar function $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ is said to be:

- \preceq -increasing, if for any $A, A' \in \mathcal{A}$,

$$A' \preceq A \implies \varphi(A') \leq \varphi(A);$$

- strongly \preceq -increasing, if for any $A, A' \in \mathcal{A}$,

$$A' \preceq A \text{ and } A \not\preceq A' \implies \varphi(A') < \varphi(A);$$

- \preceq -decreasing (strongly \preceq -decreasing) if the function $-\varphi$ is \preceq -increasing (strongly \preceq -increasing, respectively).

The next result will allow us to construct an inner approximation (i.e., a subset) of $\text{MIN}_{\preceq}(\mathcal{A})$ by means of the minimizers of certain \preceq -increasing scalar functions. It is a set-valued counterpart of a well-known result concerning (strongly) increasing scalar functions with respect to general preorder relations (see, e.g., Podinovskii and Nogin [18, Lemma 1.2.2]).

Lemma 3.1. *Let $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ be a function and let*

$$\text{argmin}_{A \in \mathcal{A}} \varphi(A) := \{A' \in \mathcal{A} \mid \forall A \in \mathcal{A} : \varphi(A') \leq \varphi(A)\}$$

be the set of its minimizers. If either

- φ is strongly \preceq -increasing or*
- φ is \preceq -increasing and $\text{argmin}_{A \in \mathcal{A}} \varphi(A)$ is a singleton,*

then we have

$$\text{argmin}_{A \in \mathcal{A}} \varphi(A) \subseteq \text{MIN}_{\preceq}(\mathcal{A}).$$

The following lemma generalizes a result obtained in the context of discrete vector optimization by Günther and Popovici [6, Lemma 2.10].

Lemma 3.2. *Assume that \mathcal{A} is finite and let $\mathcal{B} \subseteq \mathcal{A}$ be a nonempty family of sets. If $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ is a strongly \preceq -increasing function, then the following assertions are equivalent:*

- 1° $\mathcal{B} = \text{MIN}_{\preceq}(\mathcal{B})$.
- 2° *For all $B, B' \in \mathcal{B}$ with $\varphi(B) < \varphi(B')$ we have*

$$B \preceq B' \implies B' \preceq B.$$

Proof. The implication $1^\circ \implies 2^\circ$ is obvious, since for all $B, B' \in \mathcal{B}$ (i.e., $B, B' \in \text{MIN}_{\preceq}(\mathcal{B})$) such that $B \preceq B'$, we actually have $B' \preceq B$.

Conversely, assume that 2° holds. Since $\text{MIN}_{\preceq}(\mathcal{B}) \subseteq \mathcal{B}$, we just have to prove that $\mathcal{B} \subseteq \text{MIN}_{\preceq}(\mathcal{B})$. Assume by the contrary that there exists $B' \in \mathcal{B} \setminus \text{MIN}_{\preceq}(\mathcal{B})$. Then, there exists some $B \in \mathcal{B}$ such that $B \preceq B'$ and $B' \not\preceq B$. By the strong \preceq -increasingness of φ , we obtain $\varphi(B) < \varphi(B')$. By assumption 2° , we infer that $B' \preceq B$, a contradiction. Thus the implication $2^\circ \implies 1^\circ$ holds true. \square

Theorem 3.1. *Assume that \mathcal{A} is finite and let $\mathcal{B} \subseteq \mathcal{A}$ be a nonempty family of sets, such that $\text{MIN}_{\preceq}(\mathcal{A}) \subseteq \mathcal{B}$. If $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ is a strongly \preceq -increasing function, then the following assertions are equivalent:*

- 1° $\mathcal{B} = \text{MIN}_{\preceq}(\mathcal{A})$.
- 2° *For all $B, B' \in \mathcal{B}$ with $\varphi(B) < \varphi(B')$, we have*

$$B \preceq B' \implies B' \preceq B.$$

Proof. Directly follows by Lemmas 2.1 and 3.2. \square

Remark 3.1. Theorem 3.1 represents a generalization of a result obtained by Günther and Popovici [6, Theorem 2.11] within vector optimization.

4. ALGORITHMS FOR COMPUTING THE SET $\text{MIN}_{\preceq}(\mathcal{A})$ WHEN \mathcal{A} IS FINITE

In this section, we assume that

$$\mathcal{A} = \{A^1, \dots, A^p\} \subseteq \mathcal{P}(E)$$

is a finite family of sets, where $p \in \mathbb{N}$, $p \geq 2$.

The next algorithm, proposed by Köbis, Kuroiwa and Tammer [16], represents an extension of the classical Graef-Younes method from vector optimization (see, e.g., Younes [20] and Jahn [7, Section 12.4]) to set optimization. It can be seen as a reduction procedure, which consists in eliminating some of the non-minimal elements of \mathcal{A} , by producing in this way a family $\mathcal{B} \subseteq \mathcal{A}$, which still contains all minimal elements of \mathcal{A} .

Algorithm 1: GENERALIZED GRAEF-YOUNES METHOD (REDUCTION PROCEDURE)

Input: The set $\mathcal{A} = \{A^1, \dots, A^p\} \subseteq \mathcal{P}(E)$.

$\mathcal{B} \leftarrow \{A^1\};$

for $j \leftarrow 2$ **to** p **do**

if $\nexists B \in \mathcal{B} : B \preceq A^j$ **and** $A^j \not\preceq B$ **then**

$\mathcal{B} \leftarrow \mathcal{B} \cup \{A^j\};$

end

end

Output: The set \mathcal{B} (containing all minimal elements of \mathcal{A} w.r.t. \preceq).

Remark 4.1. In the worst-case scenario, when $\mathcal{B} = \mathcal{A}$, we need to perform

$$1 + 2 + \dots + (p-1) = \frac{(|\mathcal{A}| - 1) \cdot |\mathcal{A}|}{2} \quad (4.1)$$

pairwise comparisons of elements w.r.t. \preceq (where, a comparison of two sets $A, A' \in \mathcal{A}$ consists in checking both conditions $A' \preceq A$ and $A \not\preceq A'$).

Remark 4.2. As pointed out by Köbis, Kuroiwa and Tammer [16, Theorem 4.3], the set \mathcal{B} generated by Algorithm 1 satisfies the property $\text{MIN}_{\preceq}(\mathcal{A}) \subseteq \mathcal{B} \subseteq \mathcal{A}$. Moreover, by (2.2) in Lemma 2.1, we actually have $\text{MIN}_{\preceq}(\mathcal{B}) = \text{MIN}_{\preceq}(\mathcal{A})$.

In general, the set \mathcal{B} generated by Algorithm 1 may significantly exceed the set $\text{MIN}_{\preceq}(\mathcal{A})$. To overcome this drawback, we consider the following algorithm, which extends the classical “Jahn-Graef-Younes method” (also called “Graef-Younes method with backward iteration” by Jahn [7, 12], Jahn and Rathje [8]) from vector optimization to set optimization with respect to the preorder relation \preceq . Notice that Algorithm 2 has been already formulated by Köbis, Kuroiwa and Tammer [16, Algorithm 4.11], but under different assumptions, namely when \preceq is antisymmetric (not necessarily reflexive or transitive) while $\text{MIN}_{\preceq}(\mathcal{A})$ is nonempty and externally stable.

Algorithm 2: GENERALIZED JAHN-GRAEF-YOUNES METHOD (FORWARD REDUCTION PROCEDURE & BACKWARD REDUCTION PROCEDURE)

Input: The set $\mathcal{A} = \{A^1, \dots, A^p\} \subseteq \mathcal{P}(E)$.
 /* Phase 1 (Forward reduction procedure) */
 $i \leftarrow 1$;
 $\mathcal{B} \leftarrow \{A^i\}$;
for $j \leftarrow 2$ **to** p **do**
 if $\nexists B \in \mathcal{B} : B \preceq A^j$ and $A^j \not\preceq B$ **then**
 $i \leftarrow i + 1$;
 $B^i \leftarrow A^j$;
 $\mathcal{B} \leftarrow \mathcal{B} \cup \{B^i\}$;
 end
end
 /* Phase 2 (Backward reduction procedure) */
 $\mathcal{T} \leftarrow \{B^i\}$;
for $j \leftarrow 1$ **to** $i - 1$ **do**
 if $\nexists T \in \mathcal{T} : T \preceq B^{i-j}$ and $B^{i-j} \not\preceq T$ **then**
 $\mathcal{T} \leftarrow \mathcal{T} \cup \{B^{i-j}\}$;
 end
end
Output: The set \mathcal{T} (representing the set of all minimal elements of \mathcal{A} w.r.t. \preceq).

Remark 4.3. In the worst-case scenario, namely when $\mathcal{B} = \mathcal{A}$, Algorithm 2 needs to perform more than

$$\frac{(|\mathcal{A}| - 1) \cdot |\mathcal{A}|}{2}$$

pairwise comparisons of elements w.r.t. \preceq .

Theorem 4.1. The set \mathcal{T} generated by Algorithm 2 is actually the set $\text{MIN}_{\preceq}(\mathcal{A})$.

Proof. In view of Remark 4.2, we have $\text{MIN}_{\preceq}(\mathcal{A}) \subseteq \mathcal{B} \subseteq \mathcal{A}$ and $\text{MIN}_{\preceq}(\mathcal{B}) \subseteq \mathcal{T} \subseteq \mathcal{B}$. Hence

$$\text{MIN}_{\preceq}(\mathcal{A}) = \text{MIN}_{\preceq}(\mathcal{A}) \cap \mathcal{B} \subseteq \text{MIN}_{\preceq}(\mathcal{B}) \subseteq \mathcal{T}.$$

Thus, inclusion $\text{MIN}_{\preceq}(\mathcal{A}) \subseteq \mathcal{T}$ is satisfied.

Let us prove the reverse inclusion $\text{MIN}_{\preceq}(\mathcal{A}) \supseteq \mathcal{T}$. To this aim, consider an element $T \in \mathcal{T} \subseteq \mathcal{B}$. Then, there exists $j \in \{1, \dots, p\}$ such that $T = A^j$. Moreover, by the forward reduction procedure, we have

$$\nexists B \in \mathcal{B} \cap \{A^k \mid k \in \{1, \dots, p\}, k < j\} : B \preceq A^j \text{ and } A^j \not\preceq B.$$

Hence

$$\nexists B \in \mathcal{T} \cap \{A^k \mid k \in \{1, \dots, p\}, k < j\} : B \preceq A^j \text{ and } A^j \not\preceq B.$$

On the other hand, by the backward reduction procedure, we also have

$$\nexists B \in \mathcal{T} \cap \{A^k \mid k \in \{1, \dots, p\}, k > j\} : B \preceq A^j \text{ and } A^j \not\preceq B.$$

Thus, we infer that

$$\nexists B \in \mathcal{T} : B \preccurlyeq A^j \text{ and } A^j \not\preccurlyeq B,$$

which exactly means that $A^j \in \text{MIN}_{\preccurlyeq}(\mathcal{T})$. Now, taking into account that $\text{MIN}_{\preccurlyeq}(\mathcal{A}) \subseteq \mathcal{T} \subseteq \mathcal{A}$, by applying (2.2) in Lemma 2.1 for \mathcal{T} in the role of \mathcal{B} , we get $\text{MIN}_{\preccurlyeq}(\mathcal{T}) = \text{MIN}_{\preccurlyeq}(\mathcal{A})$. We conclude that $A^j \in \text{MIN}_{\preccurlyeq}(\mathcal{A})$. \square

Observe that, given any function $\varphi : \mathcal{A} \rightarrow \mathbb{R}$, we can always find an enumeration $\{A^{j_1}, \dots, A^{j_p}\}$ of the set $\mathcal{A} = \{A^1, \dots, A^p\}$ such that

$$\varphi(A^{j_1}) \leq \varphi(A^{j_2}) \leq \dots \leq \varphi(A^{j_p}). \quad (4.2)$$

Thus we can pre-sort the elements of \mathcal{A} according to (4.2) and thereafter apply the algorithms stated above to $\{A^{j_1}, \dots, A^{j_p}\}$ instead of the original enumeration $\{A^1, \dots, A^p\}$. When φ is a strongly \preccurlyeq -increasing function we obtain an interesting property, highlighted by the next theorem.

Theorem 4.2. *If $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ is strongly \preccurlyeq -increasing and*

$$\varphi(A^1) \leq \varphi(A^2) \leq \dots \leq \varphi(A^p), \quad (4.3)$$

then Algorithm 1 (Generalized Graef-Younes method) generates the set $\mathcal{B} = \text{MIN}_{\preccurlyeq}(\mathcal{A})$.

Proof. The set \mathcal{B} generated by Algorithm 1 satisfies $\text{MIN}_{\preccurlyeq}(\mathcal{A}) \subseteq \mathcal{B} \subseteq \mathcal{A}$ in view of Remark 4.2. Therefore, in order to prove that $\mathcal{B} = \text{MIN}_{\preccurlyeq}(\mathcal{A})$ it remains to show that \mathcal{B} satisfies the property 2° of Theorem 3.1. Indeed, consider any $B, B' \in \mathcal{B}$ with $\varphi(B) < \varphi(B')$. Taking into account that $B, B' \in \mathcal{A}$, there exist $i, i' \in \{1, \dots, p\}$ such that $B = A^i$ and $B' = A^{i'}$, hence $\varphi(A^i) < \varphi(A^{i'})$. By (4.3) it follows that $i < i'$, which shows that during Algorithm 1, the point $B' = A^{i'}$ is added to \mathcal{B} after $B = A^i$. This ensures that we have $B \preccurlyeq B' \implies B' \preccurlyeq B$. \square

Remark 4.4. In contrast to Theorem 4.2, if $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ is strongly \preccurlyeq -decreasing (instead of being strongly \preccurlyeq -increasing) and (4.3) holds, then Algorithm 1 generates the initial set, i.e., $\mathcal{B} = \mathcal{A}$ (no reduction occurs). Indeed, by construction of \mathcal{B} we have $A^1 \in \mathcal{B}$. Moreover, for any $j \in \{2, \dots, p\}$ we have $A^i \preccurlyeq A^j \implies A^j \preccurlyeq A^i$ for all $i \in \{1, \dots, p\}$ with $i < j$ (otherwise we should have $A^i \preccurlyeq A^j$ and $A^j \not\preccurlyeq A^i$, hence $\varphi(A^i) > \varphi(A^j)$ by the strong \preccurlyeq -decreasingness of φ , a contradiction). Thus we have $A^j \in \mathcal{B}$.

Theorem 4.3. *If $\varphi : \mathcal{A} \rightarrow \mathbb{R}$ is \preccurlyeq -increasing and*

$$\varphi(A^1) < \varphi(A^2) < \dots < \varphi(A^p), \quad (4.4)$$

then Algorithm 1 (Generalized Graef-Younes method) generates the set $\mathcal{B} = \text{MIN}_{\preccurlyeq}(\mathcal{A})$.

Proof. Assume by the contrary that $A^j \in \mathcal{B} \setminus \text{MIN}_{\preccurlyeq}(\mathcal{A})$ for some $j \in \{1, \dots, p\}$. Then, we infer the existence of a set $A' \in \mathcal{A}$ such that $A' \preccurlyeq A^j$ and $A^j \not\preccurlyeq A'$. We distinguish two cases:

Case 1: If $A' \in \text{MIN}_{\preccurlyeq}(\mathcal{A})$, then there exists $i \in \{1, \dots, p\}$ such that $A' = A^i$. Thus, we have $A^i \preccurlyeq A^j$ and $A^j \not\preccurlyeq A^i$.

Case 2: If $A' \in \mathcal{A} \setminus \text{MIN}_{\preccurlyeq}(\mathcal{A})$, then by the external stability of $\text{MIN}_{\preccurlyeq}(\mathcal{A})$ (see Remark 2.2), there is $A^i \in \text{MIN}_{\preccurlyeq}(\mathcal{A})$ with $i \in \{1, \dots, p\}$ such that $A^i \preccurlyeq A'$. By transitivity of \preccurlyeq we infer that in this case we also have $A^i \preccurlyeq A^j$ and $A^j \not\preccurlyeq A^i$.

Since φ is \preceq -increasing and $A^i \preceq A^j$, it follows that $\varphi(A^i) \leq \varphi(A^j)$. Therefore, by (4.4) we deduce that $i \leq j$. Actually, since $A^j \not\preceq A^i$ and \preceq is reflexive, we infer that $i < j$. By the construction of \mathcal{B} and taking into account that $A^i \preceq A^j$ and $A^i \in \mathcal{B}$, we get $A^j \preceq A^i$, a contradiction. \square

Now we are ready to present a new method (Algorithm 3) that generates the set $\text{MIN}_{\preceq}(\mathcal{A})$, by pre-sorting the elements of \mathcal{A} before applying the Generalized Graef-Younes reduction procedure. It extends the method proposed by Günther and Popovici [6] for discrete vector optimization to set optimization.

Algorithm 3: GENERALIZED GRAEF-YOUNES METHOD WITH PRE-SORTING (PRE-SORTING PROCEDURE & REDUCTION PROCEDURE)

Input: The set $\mathcal{A} = \{A^1, \dots, A^p\} \subseteq \mathcal{P}(E)$ and the strongly \preceq -increasing function $\varphi : \mathcal{A} \rightarrow \mathbb{R}$.
 /* Phase 1 (Pre-sorting procedure) */
 Compute an enumeration $\mathcal{A} = \{A^{j_1}, \dots, A^{j_p}\}$ which satisfies (4.2).
 /* Phase 2 (Reduction procedure) */
 $\mathcal{B} \leftarrow \{A^{j_1}\};$
for $k \leftarrow 2$ **to** p **do**
 | **if** $\nexists B \in \mathcal{B} : B \preceq A^{j_k}$ **and** $A^{j_k} \not\preceq B$ **then**
 | | $\mathcal{B} \leftarrow \mathcal{B} \cup \{A^{j_k}\};$
 | **end**
end
 $\mathcal{T} \leftarrow \mathcal{B};$
Output: The set \mathcal{T} (representing the set of all minimal elements of \mathcal{A} w.r.t. \preceq).

Remark 4.5. In Phase 1 of Algorithm 3 we generate an enumeration of \mathcal{A} by rising the values of the sorting function φ . In Phase 2 of Algorithm 3 the Generalized Graef-Younes method (Algorithm 1) is applied to the sorted set $\{A^{j_1}, \dots, A^{j_p}\}$ in the role of \mathcal{A} .

Remark 4.6. In the worst-case scenario, Algorithm 3 needs to perform

$$|\mathcal{T}| \cdot \frac{(2|\mathcal{A}| - |\mathcal{T}| - 1)}{2}$$

pairwise comparisons of elements w.r.t. \preceq , where $\mathcal{T} = \text{MIN}_{\preceq}(\mathcal{A})$. Indeed, since at every iteration in Phase 2 we have $\mathcal{B} \subseteq \text{MIN}_{\preceq}(\mathcal{A}) = \mathcal{T}$, the worst-case scenario occurs when

$$A^{j_1}, \dots, A^{j_{|\mathcal{T}|}} \in \mathcal{T} \quad \text{and} \quad A^{j_{|\mathcal{T}|+1}}, \dots, A^{j_{|\mathcal{A}|}} \notin \mathcal{T},$$

requesting to perform

$$\frac{(|\mathcal{T}| - 1) \cdot |\mathcal{T}|}{2} + (|\mathcal{A}| - |\mathcal{T}|) \cdot |\mathcal{T}| = \frac{|\mathcal{T}| \cdot (2|\mathcal{A}| - |\mathcal{T}| - 1)}{2} \quad (4.5)$$

pairwise comparisons.

Formula (4.5) in particular shows that Algorithm 3 is an output-sensitive method, because its runtime depends on the size of the output set \mathcal{T} . Notice also that the number of pairwise comparisons given by (4.5) for Algorithm 3 may be significantly smaller than the number of pairwise comparisons given by (4.1) for Algorithm 1.

5. NUMERICAL EXPERIMENTS

5.1. Construction of test families of sets and strongly increasing functions. In order to implement and test our Algorithms 2 and 3, in this section we will consider finite families of line segments parallel to the second bisector in $E = \mathbb{R}^2$ (as illustrated in Example 2.1) that can be also degenerated in singletons (as in the vector optimization framework). We denote such a family of segments by

$$\mathcal{A}^* = \{A^1, \dots, A^p\},$$

where $p \in \mathbb{N}$, $p \geq 2$. Actually, for any $i \in \{1, \dots, p\}$, we can represent A^i in the form

$$A^i = a^i + \alpha_i S \quad (5.1)$$

for some point $a^i = (a_1^i, a_2^i) \in \mathbb{R}^2$ and some real number $\alpha_i \geq 0$, where S is the line segment

$$S = \{(-x, x) \mid x \in [-1, 1]\}.$$

By considering the standard ordering cone, $C = \mathbb{R}_+^2$, we will compute the sets $\text{MIN}_{\leq_C^\ell}(\mathcal{A}^*)$, $\text{MIN}_{\leq_C^u}(\mathcal{A}^*)$ and $\text{MIN}_{\leq_C^s}(\mathcal{A}^*)$, by applying our Algorithms 2 and 3 for $\mathcal{A} := \mathcal{A}^*$ and \leq_C^ℓ , \leq_C^u and \leq_C^s in the role of \preceq . More precisely, in order to apply Algorithm 3, we will construct three functions, $\varphi^\ell, \varphi^u, \varphi^s : \mathcal{A}^* \rightarrow \mathbb{R}$, that are strongly increasing with respect to \leq_C^ℓ , \leq_C^u and \leq_C^s , respectively. To this aim, we follow an approach proposed by Jahn [11]. For each direction

$$\lambda \in \Lambda := \{(1, 0), (0, 1), (1, 1)\}$$

we define two auxiliary functions $\psi_\lambda^\ell, \psi_\lambda^u : \mathcal{A}^* \rightarrow \mathbb{R}$ for any $A \in \mathcal{A}^*$ by

$$\psi_\lambda^\ell(A) = \min_{a \in A} \langle \lambda, a \rangle \quad \text{and} \quad \psi_\lambda^u(A) = \max_{a \in A} \langle \lambda, a \rangle.$$

Remark 5.1. By the explicit representation (5.1) it follows that, for any $i \in \{1, \dots, p\}$,

$$\begin{cases} \psi_\lambda^\ell(A^i) = a_1^i - \alpha_i \text{ and } \psi_\lambda^u(A^i) = a_1^i + \alpha_i & \text{if } \lambda = (1, 0) \\ \psi_\lambda^\ell(A^i) = a_2^i + \alpha_i \text{ and } \psi_\lambda^u(A^i) = a_2^i - \alpha_i & \text{if } \lambda = (0, 1) \\ \psi_\lambda^\ell(A^i) = \psi_\lambda^u(A^i) = a_1^i + a_2^i & \text{if } \lambda = (1, 1). \end{cases}$$

Lemma 5.1. For any $A, A' \in \mathcal{A}^*$ we have:

- 1° $A' \leq_C^\ell A \iff \forall \lambda \in \Lambda : \psi_\lambda^\ell(A') \leq \psi_\lambda^\ell(A).$
- 2° $A' \leq_C^u A \iff \forall \lambda \in \Lambda : \psi_\lambda^u(A') \leq \psi_\lambda^u(A).$
- 3° $A' \leq_C^s A \iff \forall \lambda \in \Lambda \setminus \{(1, 1)\} : \psi_\lambda^\ell(A') \leq \psi_\lambda^\ell(A) \text{ and } \psi_\lambda^u(A') \leq \psi_\lambda^u(A).$

Proof. These assertions hold in view of Remark 5.1. □

Lemma 5.2. For any $A, A' \in \mathcal{A}^*$ we have:

- 1° $A \not\leq_C^\ell A' \iff \exists \lambda \in \Lambda : \psi_\lambda^\ell(A') < \psi_\lambda^\ell(A).$
- 2° $A \not\leq_C^u A' \iff \exists \lambda \in \Lambda : \psi_\lambda^u(A') < \psi_\lambda^u(A).$
- 3° $A \not\leq_C^s A' \iff \exists \lambda \in \Lambda \setminus \{(1, 1)\} : \psi_\lambda^\ell(A') < \psi_\lambda^\ell(A) \text{ or } \psi_\lambda^u(A') < \psi_\lambda^u(A).$

Proof. Directly follows by Lemma 5.1. □

Now, we define the functions $\varphi^\ell, \varphi^u, \varphi^s : \mathcal{A}^* \rightarrow \mathbb{R}$ for any $A \in \mathcal{A}^*$ by

$$\begin{aligned}\varphi^\ell(A) &= \sum_{\lambda \in \Lambda} \psi_\lambda^\ell(A), \\ \varphi^u(A) &= \sum_{\lambda \in \Lambda} \psi_\lambda^u(A), \\ \varphi^s(A) &= \sum_{\lambda \in \Lambda \setminus \{(1,1)\}} (\psi_\lambda^\ell + \psi_\lambda^u)(A).\end{aligned}$$

Theorem 5.1. *Function φ^ℓ is strongly \leq_C^ℓ -increasing, function φ^u is strongly \leq_C^u -increasing, while function φ^s is strongly \leq_C^s -increasing.*

Proof. In order to prove that φ^ℓ is strongly \leq_C^ℓ -increasing, let any $A', A \in \mathcal{A}^*$ be such that $A' \leq_C^\ell A$ and $A \not\leq_C^\ell A'$. Then, in view of Lemmas 5.1 and 5.2 (1°), we have $\psi_\lambda^\ell(A') \leq \psi_\lambda^\ell(A)$ for any $\lambda \in \Lambda$ and $\psi_{\lambda_0}^\ell(A') < \psi_{\lambda_0}^\ell(A)$ for some $\lambda_0 \in \Lambda$. Clearly, this implies that $\varphi^\ell(A') < \varphi^\ell(A)$. Thus φ^ℓ is strongly \leq_C^ℓ -increasing. Similarly, we can show that φ^u is strongly \leq_C^u -increasing and φ^s is strongly \leq_C^s -increasing, by using Lemmas 5.1 and 5.2 (2° and 3°, respectively). \square

5.2. Comparative analysis of our algorithms. As suggested by Eichfelder and Gerlach [5], there are several natural ways to construct set-valued test functions in practice: (i) by adding a constant set to a single-valued function, (ii) by multiplying a constant set by a real-valued function or (iii) by considering a sum of set-valued functions given by (i) and (ii). We will use an approach similar to (iii) in order to test our Algorithms 2 and 3. More precisely, by considering a planar curve $\gamma : [0, \tau] \rightarrow \mathbb{R}^2$ (with $\tau > 0$), a nonnegative function $\alpha : [0, \tau] \rightarrow \mathbb{R}_+$ and a finite number of nodes $0 \leq t_1 < \dots < t_p \leq \tau$ ($p \in \mathbb{N}$, $p \geq 2$), we define the family

$$\mathcal{A}^* = \{A^1, \dots, A^p\}$$

of line segments of type (5.1) where, for any $i \in \{1, \dots, p\}$, $a^i = \gamma(t_i)$ and $\alpha_i = \alpha(t_i)$, i.e.,

$$A^i = \gamma(t_i) + \alpha(t_i)S.$$

In Table 1 we present the numerical results obtained by implementing Algorithms 2 and 3 in MATLAB 2016a, on a Core i7-8550U CPU, 16GB RAM computer. They correspond to five families of sets of type \mathcal{A}^* with $p \in \{100, 1000, 10000, 50000, 100000\}$, where

$$\gamma(t) = (t, (t/2) \sin t) \quad \text{and} \quad \alpha(t) = |\sin t| \quad \text{for any} \quad 0 \leq t \leq \tau = 4\pi,$$

while, for every p , the nodes $0 \leq t_1 < \dots < t_p \leq 4\pi$ are generated randomly by means of the MATLAB function `rand`.

We mention that in both Algorithms 2 and 3, the pairwise comparisons of sets is performed by means of Lemmas 5.1 and 5.2, where the values of the auxiliary functions ψ_λ^ℓ and ψ_λ^u on the elements of \mathcal{A}^* are a priori computed as indicated in Remark 5.1. Moreover, in Phase 1 of Algorithm 3 (Pre-sorting procedure), we first compute the values of the sorting functions φ^ℓ , φ^u and φ^s on the elements of \mathcal{A}^* by means of the explicit representations established in Section 5.1, and thereafter we sort these values by using the MATLAB function `sort`.

p	100	1000	10000	50000	100000
$ \text{MIN}_{\leq_C^\ell}(\mathcal{A}^*) $	24	280	2762	13677	27312
$ \text{MIN}_{\leq_C^u}(\mathcal{A}^*) $	28	236	2560	12652	25369
$ \text{MIN}_{\leq_C^s}(\mathcal{A}^*) $	32	311	3245	16059	32078

Algorithm 2 (Generalized Jahn-Graef-Younes method)

Runtime for $\text{MIN}_{\leq_C^\ell}(\mathcal{A}^*)$	0.0128	0.1204	10.7291	173.5408	868.8651
Runtime for $\text{MIN}_{\leq_C^u}(\mathcal{A}^*)$	0.0150	0.0909	8.7809	148.9064	774.1772
Runtime for $\text{MIN}_{\leq_C^s}(\mathcal{A}^*)$	0.0198	0.1799	25.8091	306.1024	1594.7805

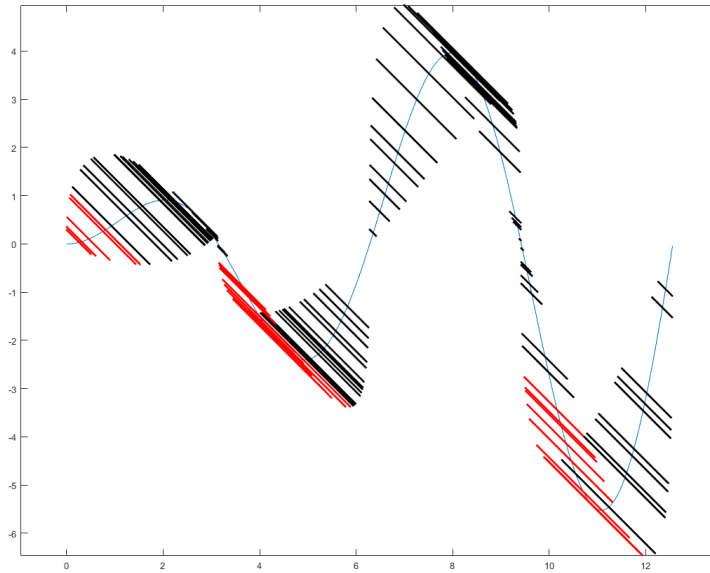
Algorithm 3 (Generalized Graef-Younes method with pre-sorting)

Runtime for $\text{MIN}_{\leq_C^\ell}(\mathcal{A}^*)$	0.0109	0.1019	9.5440	159.0386	849.8282
Runtime for $\text{MIN}_{\leq_C^u}(\mathcal{A}^*)$	0.0116	0.0742	8.4303	134.8084	715.7421
Runtime for $\text{MIN}_{\leq_C^s}(\mathcal{A}^*)$	0.0150	0.1485	24.8091	288.4068	1519.2157

TABLE 1. Computational study for the five test families \mathcal{A}^* in Section 5.2.

As the numerical results listed in Table 1 reveals, both Algorithms 2 and 3 are quite effective even for significantly large p , the latter requesting however a shorter runtime for all five test families, due to the pre-sorting procedure. Notice that a similar behavior has been already observed within vector optimization problems by Günther and Popovici [6].

Finally, we illustrate in Figures 2, 3, and 4 the family \mathcal{A}^* with $p = 100$ along with the corresponding output sets $\text{MIN}_{\leq_C^\ell}(\mathcal{A}^*)$, $\text{MIN}_{\leq_C^u}(\mathcal{A}^*)$ and $\text{MIN}_{\leq_C^s}(\mathcal{A}^*)$ mentioned in Table 1.

FIGURE 2. The set $\text{MIN}_{\leq_C^\ell}(\mathcal{A}^*)$ mentioned in Table 1 for $p = 100$ (red color).

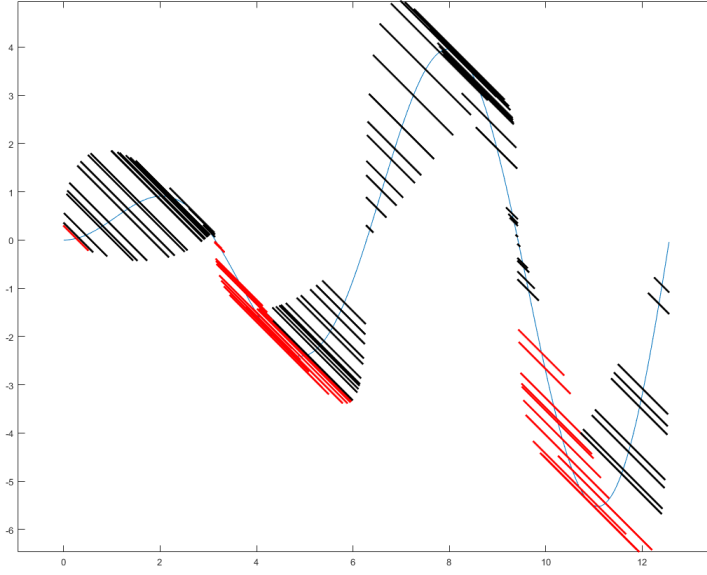


FIGURE 3. The set $\text{MIN}_{\leq_C^u}(\mathcal{A}^*)$ mentioned in Table 1 for $p = 100$ (red color).

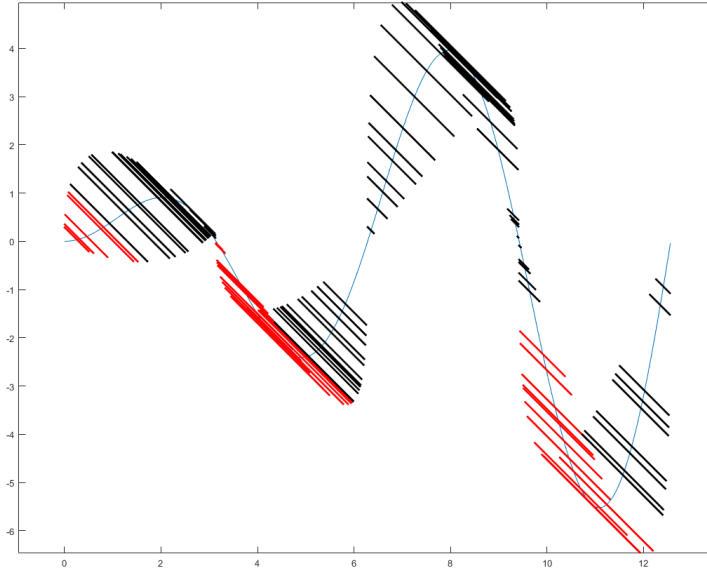


FIGURE 4. The set $\text{MIN}_{\leq_C^s}(\mathcal{A}^*)$ mentioned in Table 1 for $p = 100$ (red color).

6. CONCLUSIONS

Our algorithms are conceived for computing the minimal elements of any nonempty finite family of sets $\mathcal{A} \subseteq \mathcal{P}(E)$ with respect to a general preorder relations on $\mathcal{P}(E)$. Although we have implemented them only for the preorder relations \leq_C^ℓ , \leq_C^u and \leq_C^s and a particular type of sets, for which we succeeded to construct strongly increasing sorting functions, it would be of both theoretical and practical interest for further research to consider other (preorder) relations and different classes of sets, along with appropriate strongly increasing sorting functions. In particular, since a variant of the Jahn-Graef-Younes method for vector optimization problems has been already developed for variable ordering structures by Eichfelder

[3], it is natural to study whether this method could be generalized for set optimization problems in the framework considered by Eichfelder and Pilecka [1, 2]. Other interesting research topics would be to involve a pre-sorting procedure in the numerical methods proposed by Köbis and Le [14] for computing strict, strong and ideal minimal elements of a finite family of sets, as well as in the numerical methods for computing approximations of minimal elements, following the works mentioned in the Introduction.

Acknowledgments

Nicolae Popovici's research was partially supported by a grant of the Romanian Ministry of Research and Innovation, CNCS-UEFISCDI, project number PN-III-P4-ID-PCE-2016-0190, within PNCDI III.

REFERENCES

- [1] G. Eichfelder, M. Pilecka, Set approach for set optimization with variable ordering structures, Part I: Set relations and relationship to vector approach, *J. Optim. Theory Appl.* 171 (2016), 931-946.
- [2] G. Eichfelder, M. Pilecka, Set approach for set optimization with variable ordering structures, Part II: Scalarization approaches, *J. Optim. Theory Appl.* 171 (2016), 947-963.
- [3] G. Eichfelder, *Variable Ordering Structures in Vector Optimization*, Springer, Berlin, Heidelberg, 2014.
- [4] G. Eichfelder, Vector optimization in medical engineering, In: P.M. Pardalos, T.M. Rassias (Ed.), *Mathematics without boundaries. Surveys in interdisciplinary research*, pp. 181-215, Springer, New York, 2014.
- [5] G. Eichfelder, T. Gerlach, On classes of set optimization problems which are reducible to vector optimization problems and its impact on numerical test instances, Chapter 10 in: A. Khan, E. Köbis, Chr. Tammer (Ed.), *Variational analysis and set optimization*, pp. 265-290, CRC Press, 2019.
- [6] C. Günther, N. Popovici, New algorithms for discrete vector optimization based on the Graef-Younes method and cone-monotone sorting functions, *Optimization*, 67 (2018), 975-1003.
- [7] J. Jahn, *Vector optimization - Introduction, theory, and extensions*, Springer, Berlin, Heidelberg, 2011.
- [8] J. Jahn, U. Rathje, Graef-Younes method with backward iteration, In: Chr. Tammer, K. Winkler, K.H. Küfer, H. Rommelfanger (Ed.), *Multicriteria decision making and fuzzy systems - Theory, methods and applications*, pp. 75-81, Shaker Verlag, Aachen, 2006.
- [9] J. Jahn, T.X.D. Ha, New order relations in set optimization, *J. Optim. Theory Appl.* 148 (2011), 209-236.
- [10] J. Jahn, A derivative-free descent method in set optimization, *Comput. Optim. Appl.* 60 (2015), 393-411.
- [11] J. Jahn, Vectorization in set optimization, *J. Optim. Theory Appl.* 167 (2015), 783-795.
- [12] J. Jahn, Multiobjective search algorithm with subdivision technique, *Comput. Optim. Appl.* 35 (2006), 161-175.
- [13] A. Khan, C. Tammer, C. Zălinescu, *Set-valued optimization - An introduction with applications*, Springer, Berlin, Heidelberg, 2015.
- [14] E. Köbis, T.T. Le, Numerical procedures for obtaining strong, strict and ideal minimal solutions of set optimization problems, *Appl. Anal. Optim.* 2 (2018), 423-440.
- [15] E. Köbis, M.A. Köbis, Treatment of set order relations by means of a nonlinear scalarization functional: A full characterization, *Optimization* 65 (2016), 1805-1827.
- [16] D. Kuroiwa, E. Köbis, C. Tammer, Generalized set order relations and their numerical treatment, *Appl. Anal. Optim.* 1 (2017), 45-65.
- [17] D. Kuroiwa, The natural criteria in set-valued optimization, *RIMS Kokyuroku* 1031 (1998), 85-90.
- [18] V.V. Podinovskii, V.D. Nogin, *Pareto optimal solutions of multicriteria optimization problems (in Russian)*, Nauka, Moscow, 1982.
- [19] K. Seto, D. Kuroiwa, N. Popovici, A systematization of convexity and quasiconvexity concepts for set-valued maps, defined by l -type and u -type preorder relations, *Optimization* 67 (2018), 1077-1094.
- [20] Y.M. Younes, *Studies on discrete vector optimization*, Dissertation, University of Damiatta, Egypt, 1993.