Using quasidifferentiable functions for the separation of point sets

ANJA HERKLOTZ and BERND LUDERER

Technische Universität Chemnitz, Fakultät für Mathematik, 09107 Chemnitz, Deutschland

Summary: The aim of this paper is to realize some ideas of Demyanov (2001) by means of numeric algorithms designed for quasidifferentiable functions.

Key words: Quasidifferential calculus, identification of point sets, numerical algorithms

Mathematics Subject Classification: 90 C 30, 49 J 52, 49 M 37, 65 K 05

1 Introduction

Quasidifferential calculus provides an important tool for the solution of various nondifferentiable optimization problems. In his papers [3, 4] Demyanov discussed the questions how to separate two convex sets or how to determine as much elements as possible belonging to the one or to the other set or to the intersection of them. He proposed an algorithm that successively approximates the intersection of these sets. To this aim, he studied a quasidifferentiable function which is to be minimized. Special attention is paid to convex polyhedral sets.
In the present paper we intend to realize the ideas of Demyanov. The quasidifferential of the relevant function is described and an algorithm of Luderer and Weigelt [2] is applied to nonsmooth functions resulting from an exact penalty approach and Rosen's gradient projection method (cf. Rosen [6, 7]), resp. It should be noted, however, that the convergence of the mentioned algorithm from [2] can be ensured only for subdifferentiable problems, while the functionals occurring here are quasidifferentiable in general. We show that the original algorithm due to Demyanov et al. [3] has drawbacks in some cases. Therefore we describe some modification. Moreover, first numerical results are reported on.

2 A quasidifferentiable function for separating two sets

In the following we consider the convex compact subsets of $\mathbb{R}^n$

$$\mathcal{A} = \mathrm{co}\,\{a_i \,|\, i \in I = \{1, \ldots, N_1\}\}\,,\; \mathcal{B} = \mathrm{co}\,\{b_j \,|\, j \in J = \{1, \ldots, N_2\}\} \quad (2.1)$$

arising from a finite number of points. To these sets the functions

$$\overline{f}_{\mathcal{A}}(y) = \max_{a \in \mathcal{A}} \langle a, y \rangle, \qquad \overline{f}_{\mathcal{B}}(y) = \max_{b \in \mathcal{B}} \langle b, y \rangle, \quad (2.2)$$

$$\underline{f}_{\mathcal{A}}(y) = \min_{a \in \mathcal{A}} \langle a, y \rangle, \qquad \underline{f}_{\mathcal{B}}(y) = \min_{b \in \mathcal{B}} \langle b, y \rangle \quad (2.3)$$

are associated (cf. [4]). Furthermore, the sets of active points in direction y are introduced:

$$\mathcal{R}_{\mathcal{A}}(y) = \{a \in \mathcal{A} \,|\, \langle a, y \rangle = \overline{f}_{\mathcal{A}}(y)\}, \quad \mathcal{Q}_{\mathcal{A}}(y) = \{a \in \mathcal{A} \,|\, \langle a, y \rangle = \underline{f}_{\mathcal{A}}(y)\},$$

$$\mathcal{R}_{\mathcal{B}}(y) = \{b \in \mathcal{B} \,|\, \langle b, y \rangle = \overline{f}_{\mathcal{B}}(y)\}, \quad \mathcal{Q}_{\mathcal{B}}(y) = \{b \in \mathcal{B} \,|\, \langle b, y \rangle = \underline{f}_{\mathcal{B}}(y)\}.$$

According to the rules of quasidifferential calculus (cf. [1]) the function $\overline{f}_{\mathcal{A}}(y)$ from (2.2) is obviously subdifferentiable with the (nonunique) quasidifferential $\mathcal{D}\overline{f}_{\mathcal{A}}(y) = [\underline{\partial}\overline{f}_{\mathcal{A}}(y), \{\mathbf{0}\}]$, where $\underline{\partial}\overline{f}_{\mathcal{A}}(y) = \mathrm{co} \bigcup\limits_{a \in R_{\mathcal{A}}(y)} a$, while the function $\underline{f}_{\mathcal{A}}(y)$ from (2.3) is superdifferentiable with the (nonunique) quasidifferential $\mathcal{D}\underline{f}_{\mathcal{A}}(y) = [\{\mathbf{0}\}, \overline{\partial}\underline{f}_{\mathcal{A}}(y)]$ with $\overline{\partial}\underline{f}_{\mathcal{A}}(y) = \mathrm{co} \bigcup\limits_{a \in Q_{\mathcal{A}}(y)} a$. Analogous assertions apply to the functions $\overline{f}_{\mathcal{B}}(y)$ and $\underline{f}_{\mathcal{B}}(y)$, too.
On the basis of (2.2) and (2.3), the function

$$F_{\mathcal{AB}}(y) = |\max_{a \in \mathcal{A}} \langle a, y \rangle - \max_{b \in \mathcal{B}} \langle b, y \rangle| + |\min_{a \in \mathcal{A}} \langle a, y \rangle - \min_{b \in \mathcal{B}} \langle b, y \rangle|, \quad (2.4)$$

which describes the difference between the width of the set $\mathcal{A} \cup \mathcal{B}$ and the width of the set $\mathcal{A} \cap \mathcal{B}$ in direction y is constructed. It provides some kind of measure of how many points can be identified in direction y. The largest number of identifiable points can be expected in that direction for which the function (2.4) achieves a maximum over $\mathcal{G}_0 = \{y \in \mathbb{R}^n \,|\, \|y\| = 1\}$ (here $\|\cdot\|$ means the Euclidean norm). Of course, the actual number of identifiable points depends on the special form of $\mathcal{A}$ and $\mathcal{B}$.
Thus, the following optimization problem is to be solved:

$$F_{\mathcal{AB}}(y) \longrightarrow \max_{y \in \mathcal{G}_0}, \qquad \mathcal{G}_0 = \{y \in \mathbb{R}^n \,|\, \|y\| = 1\}. \quad (2.5)$$

If for any $y \in \mathbb{R}^n$ the value $F_{\mathcal{AB}}(y)$ is equal to zero, then the relation $\mathcal{A} = \mathcal{B}$ holds.

3 Algorithms for maximizing $\mathbf{F}_{\mathcal{AB}}(\mathbf{y})$

3.1 An exact penalty approach

Due to the suggestion given in [3], problem (2.5) will be solved by an exact penalty approach (cf. [5]). Let us rewrite the constraints into the form $\mathcal{G}_0 = \{y \in \mathbb{R}^n : \varphi(y) = 0\}$ with $\varphi(y) = |\|y\| - 1|$. From the theory of exact penalties it is well-known that there exists a number $\lambda^* \geq 0$ such that for any $\lambda > \lambda^*$ the maximum problem (2.5) is equivalent to the unconstrained minimum problem

$$\overline{F}_{\mathcal{AB}}(y) = -F_{\mathcal{AB}}(y) + \lambda\varphi(y) \longrightarrow \min_{y \in \mathbb{R}^n}. \tag{3.1}$$

For solving (3.1) it is necessary to know the quasidifferential of $\overline{F}_{\mathcal{AB}}$ at an arbitrary point y. It can be derived from the rules of quasidifferential calculus (cf. [1]), when studying four scenarios concerning the location of the sets $\mathcal{A}$ and $\mathcal{B}$ (cf. the figures in [3]):

$$\begin{array}{ll}
\overline{f}_{\mathcal{A}}(y) \geq \overline{f}_{\mathcal{B}}(y), & \underline{f}_{\mathcal{A}}(y) \geq \underline{f}_{\mathcal{B}}(y), \\
\overline{f}_{\mathcal{A}}(y) < \overline{f}_{\mathcal{B}}(y), & \underline{f}_{\mathcal{A}}(y) \geq \underline{f}_{\mathcal{B}}(y), \\
\overline{f}_{\mathcal{A}}(y) \geq \overline{f}_{\mathcal{B}}(y), & \underline{f}_{\mathcal{A}}(y) < \underline{f}_{\mathcal{B}}(y), \\
\overline{f}_{\mathcal{A}}(y) < \overline{f}_{\mathcal{B}}(y), & \underline{f}_{\mathcal{A}}(y) < \underline{f}_{\mathcal{B}}(y).
\end{array}$$

Using (2.4) and the definition of φ, the function $\overline{F}_{\mathcal{AB}}(y)$ takes the form

$$\overline{F}_{\mathcal{AB}}(y) = -\Big(|\overline{f}_{\mathcal{A}}(y) - \overline{f}_{\mathcal{B}}(y)| + |\underline{f}_{\mathcal{A}}(y) - \underline{f}_{\mathcal{B}}(y)|\Big) + \lambda\varphi(y).$$

The function $\varphi(y) = |\|y\| - 1| = \max\{\|y\| - 1, 1 - \|y\|\}$ is obviously subdifferentiable, where the subdifferential is of the form

$$\underline{\partial}\varphi(y) = \begin{cases} \frac{y}{\|y\|} & \text{if } \|y\| > 1, \\ -\frac{y}{\|y\|} & \text{if } \|y\| < 1, \\ \operatorname{co}\{\frac{y}{\|y\|}, -\frac{y}{\|y\|}\} & \text{if } \|y\| = 1. \end{cases}$$

For the evaluation of the quasidifferentials of the functions $\overline{f}_{\mathcal{A}}$, $\overline{f}_{\mathcal{B}}$, $\underline{f}_{\mathcal{A}}$, $\underline{f}_{\mathcal{B}}$ we take into account the fact that according to the rules of quasidifferential calculus the subdifferential and the superdifferential changes their places when turning from a function f to the function $-f$. In this way, we get the following representations of the quasidifferential of $F_{\mathcal{AB}}(y)$.

Case 1: $F_{\mathcal{AB}}(y) = \overline{f}_{\mathcal{A}}(y) - \overline{f}_{\mathcal{B}}(y) + \underline{f}_{\mathcal{A}}(y) - \underline{f}_{\mathcal{B}}(y) + \lambda\varphi(y)$

$$\underline{\partial}\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{b \,|\, b \in \mathcal{R}_{\mathcal{B}}(y)\} + \mathrm{co}\,\{-a \,|\, a \in \mathcal{Q}_{\mathcal{A}}(y)\} + \lambda\underline{\partial}\varphi(y)$$
$$\overline{\partial}\,\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{-a_i \,|\, i \in \mathcal{R}_{\mathcal{A}}(y)\} + \mathrm{co}\,\{b_j \,|\, j \in \mathcal{Q}_{\mathcal{B}}(y)\}$$

Case 2: $F_{\mathcal{AB}}(y) = \overline{f}_{\mathcal{B}}(y) - \overline{f}_{\mathcal{A}}(y) + \underline{f}_{\mathcal{A}}(y) - \underline{f}_{\mathcal{B}}(y) + \lambda\varphi(y)$

$$\underline{\partial}\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{a \,|\, a \in \mathcal{R}_{\mathcal{A}}(y)\} + \mathrm{co}\,\{-a \,|\, a \in \mathcal{Q}_{\mathcal{A}}(y)\} + \lambda\underline{\partial}\varphi(y)$$
$$\overline{\partial}\,\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{-b \,|\, b \in \mathcal{R}_{\mathcal{B}}(y)\} + \mathrm{co}\,\{b \,|\, b \in \mathcal{Q}_{\mathcal{B}}(y)\}$$

Case 3: $F_{\mathcal{AB}}(y) = \overline{f}_{\mathcal{A}}(y) - \overline{f}_{\mathcal{B}}(y) + \underline{f}_{\mathcal{B}}(y) - \underline{f}_{\mathcal{A}}(y) + \lambda\varphi(y)$

$$\underline{\partial}\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{b \,|\, b \in \mathcal{R}_{\mathcal{B}}(y)\} + \mathrm{co}\,\{-b \,|\, b \in \mathcal{Q}_{\mathcal{B}}(y)\} + \lambda\underline{\partial}\varphi(y)$$
$$\overline{\partial}\,\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{-a \,|\, a \in \mathcal{R}_{\mathcal{A}}(y)\} + \mathrm{co}\,\{a \,|\, a \in \mathcal{Q}_{\mathcal{A}}(y)\}$$

Case 4: $F_{\mathcal{AB}}(y) = \overline{f}_{\mathcal{B}}(y) - \overline{f}_{\mathcal{A}}(y) + \underline{f}_{\mathcal{B}}(y) - \underline{f}_{\mathcal{A}}(y) + \lambda\varphi(y)$

$$\underline{\partial}\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{a \,|\, a \in \mathcal{R}_{\mathcal{A}}(y)\} + \mathrm{co}\,\{-b \,|\, b \in \mathcal{Q}_{\mathcal{B}}(y)\} + \lambda\underline{\partial}\varphi(y)$$
$$\overline{\partial}\,\overline{F}_{\mathcal{AB}}(y) = \mathrm{co}\,\{-b \,|\, b \in \mathcal{R}_{\mathcal{B}}(y)\} + \mathrm{co}\,\{a \,|\, a \in \mathcal{Q}_{\mathcal{A}}(y)\}.$$

Algorithm 1:

1. Set $k := 0$. Choose a starting point $y_0 \in \mathbb{R}^n$.
2. Determine a quasidifferential $\mathcal{D}\overline{F}_{\mathcal{AB}}(y_k)$ of $\overline{F}_{\mathcal{AB}}$ in y_k.
   If $-\overline{\partial}\,\overline{F}_{\mathcal{AB}}(y_k) \subset \underline{\partial}\overline{F}_{\mathcal{AB}}(y_k)$, i.e. $\overline{F}'_{\mathcal{AB}}(y_k; r) \geq 0$ for any $r \in \mathbb{R}^n$, then stop.
3. Find a direction of steepest descent $r_k = -\frac{v_k + w_k}{\|v_k + w_k\|}$ with $\|v_k + w_k\| = \max\limits_{w \in \overline{\partial}\,\overline{F}_{\mathcal{AB}}(y_k)} \min\limits_{v \in \underline{\partial}\overline{F}_{\mathcal{AB}}(y_k)} \|v + w\|$.
4. Find a step size α_k such that $\overline{F}_{\mathcal{AB}}(y_k + \alpha_k r_k) = \min\limits_{\alpha > 0} \overline{F}_{\mathcal{AB}}(y_k + \alpha r_k)$ and set $y_{k+1} := y_k + \alpha_k r_k$. Go to 2.

An implementation of this algorithm is described in [2]. When searching for a direction of descent, in the present paper we only use the points of the sets $\mathcal{A}_k$ and $\mathcal{B}_k$ generated by the Algorithm IDENT instead of the whole quasidifferentials which are described by convex hulls.

3.2 A gradient projection method approach

We consider the minimum problem

$$\widetilde{F}_{\mathcal{AB}}(y) = -F_{\mathcal{AB}}(y) \longrightarrow \min_{y \in \mathcal{G}_0}, \qquad \mathcal{G}_0 = \{y \in \mathbb{R}^n : \|y\| = 1\}. \tag{3.2}$$

In order to find a direction of descent in the gradient projection method for (3.2), one has to know a quasidifferential of the function $\widetilde{F}_{\mathcal{AB}}$. This task has been studied in detail in the previous section.
Furthermore, it is easy to determine the projection onto the constraints, since due to their simple structure one has $P_{\mathcal{G}_0}(y) = \frac{y}{\|y\|}$.

Algorithm 2:

1. Choose $y_0 \in \mathcal{G}_0$, set $k := 0$.
2. Determine some step size γ_k, starting from y_k in direction $r_k = -\frac{v_k + w_k}{\|v_k + w_k\|}$ with $\|v_k + w_k\| = \max_{w \in \overline{\partial} \widetilde{F}_{\mathcal{AB}}(y_k)} \min_{v \in \underline{\partial} \widetilde{F}_{\mathcal{AB}}(y_k)} \|v + w\|$.
3. Evaluate the projection $P_{\mathcal{G}_0}(\tilde{y}_k) = \frac{\tilde{y}_k}{\|\tilde{y}_k\|}$, where $\tilde{y}_k = y_k + \gamma_k r^k$.
4. If $y_k = P_{\mathcal{G}_0}(\tilde{y}_k)$, then stop. Otherwise, set $y_{k+1} = P_{\mathcal{G}_0}(\tilde{y}_k)$, $k := k + 1$, and go to step 2.

4 An algorithm for separating convex sets

In the paper Demyanov et al. [3] an algorithm for separating two convex closed compact sets $\mathcal{A} \in \mathbb{R}^n$ and $\mathcal{B} \in \mathbb{R}^n$ (in the case $\mathcal{A} \cap \mathcal{B} = \emptyset$) or for identifying the elements of these sets is proposed and described in detail for the case if the sets $\mathcal{A}$ and $\mathcal{B}$ are of the form (2.1). A point $x \in \mathcal{A} \cap \mathcal{B}$ is identified if it is possible to indicate that either $x \in \mathcal{A} \setminus \mathcal{B}$ or $x \in \mathcal{B} \setminus \mathcal{A}$. In the case $x \in \mathcal{A} \cap \mathcal{B}$ the point x is said to be *essentially unidentifiable*.
Unfortunately, in case that the intersection $\mathcal{A} \cap \mathcal{B}$ is nonempty, it is possible that this intersection is not found or approximated by the mentioned algorithm. This is clearly a drawback of this approach, since there may be practical situations, in which it is important to know the intersection (at least in an approximate manner). The example given in [3] shows this effect (see below Example 1).
To overcome the mentioned drawback of the algorithm from [3], we modify it in the following way.

Algorithm IDENT:

1. Set $k := 1, \mathcal{A}_k := \mathcal{A}$ and $\mathcal{B}_k := \mathcal{B}$.
2. Construct the function $F_k(y) = F_{\mathcal{A}_k\mathcal{B}_k}(y)$.
3. Determine y_k^* in such a way that $F_k(y_k^*) = \max\limits_{\|y\|=1} F_k(y)$.
4. If $F_k(y_k^*) = 0$, then the algorithm terminates.
5. Calculate $\overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)$ and $\underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)$ and set

$$
\begin{array}{rcl}
I_k^* &:=& \{i \in I \,|\, h(a_i, y_k^*) \in [\underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*), \overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)]\},\\
I_k^{**} &:=& \{i \in I \,|\, h(a_i, y_k^*) \notin [\underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*), \overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)]\},\\
J_k^* &:=& \{j \in J \,|\, h(b_j, y_k^*) \in [\underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*), \overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)]\},\\
J_k^{**} &:=& \{j \in J \,|\, h(b_j, y_k^*) \notin [\underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*), \overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)]\},.
\end{array}
$$

   where $h(x, y) = \langle x, y\rangle$ characterizes a separating hyperplane.
6. If $I_k^* = \emptyset$, then set

$$
\begin{array}{rcl}
\widetilde{I}_k^* &:=& \{i \in I \,|\, h(a_i, y_k^*) > \overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y^*)\},\\
\widetilde{I}_k^{**} &:=& \{i \in I \,|\, h(a_i, y_k^*) < \underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y^*)\}.
\end{array}
$$

   The case $J_k^* = \emptyset$ is treated analogously.
7. For any $i \in I_k^*$, $s \in I_k^{**}$ (resp. $i \in \widetilde{I}_k^*$, $s \in \widetilde{I}_k^{**}$ as $I_k^* = \emptyset$), set

$$a_{is}(\alpha) := \alpha a_i + (1 - \alpha) a_s$$

   and find $\underline{\alpha}_{is}$, $\overline{\alpha}_{is}$ such that the relations

$$\langle a_{is}(\underline{\alpha}_{is}), y_k^*\rangle = \underline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*), \quad \langle a_{is}(\overline{\alpha}_{is}), y_k^*\rangle = \overline{f}_{\mathcal{A}_k\mathcal{B}_k}(y_k^*)$$

   are fulfilled. Analogously do with J_k^*, J_k^{**} (resp. $\widetilde{J}_k^*$, $\widetilde{J}_k^{**}$).
8. Construct $\mathcal{A}_{k+1}$ as follows:

$$
\begin{array}{rcl}
\mathcal{A}_{k+1} &=& \mathrm{co}\,\{\{a_i \,|\, i \in I_k^*\} \cup \{a_{is}(\overline{\alpha}_{is}), a_{is}(\underline{\alpha}_{is}) \,|\\
&& \qquad \overline{\alpha}_{is}, \underline{\alpha}_{is} \in (0,1) \ \ \forall\, i \in I_k^*, s \in I_k^{**}\}\}.
\end{array}
$$

   Analogously construct $\mathcal{B}_{k+1}$ from $\mathcal{B}_k$, $\underline{\beta}_{is}$ and $\overline{\beta}_{is}$.
9. Set $k := k + 1$ and go to Step 2.

Remarks: In the following only the sets I_k^* and I_k^{**} are dealt with, for the sets J_k^* and J_k^{**} analogous statements are true, too.
There are four possibilities concerning the index sets I_k^* and I_k^{**}:

Case 1. $I_k^* = \emptyset$, $I_k^{**} = \emptyset$: The set $\mathcal{A}_k$ is empty.

Case 2. $I_k^* \neq \emptyset$, $I_k^{**} = \emptyset$: No point of the set $\mathcal{A}_k$ can be identified in the considered direction.

Case 3. $I_k^* = \emptyset$, $I_k^{**} \neq \emptyset$: All points of $\mathcal{A}_k$ are identifiable.

Case 4. $I_k^* \neq \emptyset$, $I_k^{**} \neq \emptyset$: Some points of the set $\mathcal{A}_k$ can be identified.

While Case 1 is uninteresting, Cases 2 and 4 are correct treated by the original algorithm from [3]. Only Case 3 is problematic as can be seen by the example given in [3] (cf. Example 1 given below). Of course, it is true that every single point can be associated to some set in a unique way. However, it is quite possible that points belonging to the convex hull of $\mathcal{A}$ or $\mathcal{B}$ lie in the intersection of these two sets. This was the reason for introducing Step 6 of Algorithm IDENT, which yields an approach for finding the intersection of $\mathcal{A}$ and $\mathcal{B}$.
For the constructed index sets $\widetilde{I}_k^*$ and $\widetilde{I}_k^{**}$ two cases are possible:

(A) $\widetilde{I}_k^* = \emptyset$ or $\widetilde{I}_k^{**} = \emptyset$: In this case, using the original algorithm from [3], all points can be uniquely associated.

(B) $\widetilde{I}_k^* \neq \emptyset$ and $\widetilde{I}_k^{**} \neq \emptyset$: Some points of the convex hull are elements of the intersection in the considered direction y.
The sets $\widetilde{I}_k^*$ and $\widetilde{I}_k^{**}$ are treated in the same way as the sets I_k^* and I_k^{**}: Linear combinations are formed (as described in Step 7 of Algorithm IDENT) and the so constructed points are added to the subsequent set $\mathcal{A}_{k+1}$. In this way, Algorithm IDENT proceeds with linear combinations of points, although possibly all extreme points of the considered sets could be identified.
To show how Algorithm IDENT works, we reconsider the example from [3].

Example 1: Let the sets $\mathcal{A}$ and $\mathcal{B}$ be given as follows:

$$\mathcal{A} = \mathcal{A}_1 = \mathrm{co}\left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}\right\}, \quad \mathcal{B} = \mathcal{B}_1 = \mathrm{co}\left\{\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right\}.$$

It can be easily seen that $\mathcal{A} \cap \mathcal{B} = \{\mathbf{0}\}$. In the first iteration of Algorithm IDENT we get $I_1^* = \emptyset$, $I_1^{**} = \{1, 2\}$, $J_1^* = \{1, 2\}$, $J_1^{**} = \emptyset$. Thus the situation is that of Case 3. The new index sets $\widetilde{I}_1^*$ ad $\widetilde{I}_1^{**}$ are $\widetilde{I}_1^* = \{1\}$, $\widetilde{I}_1^{**} = \{2\}$. From the demand

$$\left\langle \alpha \begin{pmatrix} 2 \\ 2 \end{pmatrix} + (1-\alpha) \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\rangle = 0$$

the value $\alpha = \frac{1}{2}$ lying in the interval $(0,1)$ is obtained. In accordance with Algorithm IDENT we get the set $\mathcal{A}_2 = \{\mathbf{0}\}$, while the set $\mathcal{B}_2$ remains unchanged compared with $\mathcal{B}_1$, i.e. $\mathcal{B}_2 = \mathcal{B}_1$. In the next iteration for the sets J_2^* and J_2^{**} the same effect occurs as for I_1^*, I_1^{**}. It is not hard to see, that $\beta = \frac{1}{2}$. Therefore, we get $\mathcal{B}_3 = \{\mathbf{0}\}$. Again, the set $\mathcal{A}_3$ remains unchanged in comparison with $\mathcal{A}_2$. In this way, in contrast to the algorithm from [3], after two iterations Algorithm IDENT yields the exact intersection of $\mathcal{A}$ and $\mathcal{B}$.

5 Numerical results

To illustrate the algorithm and to report first (preliminary) numerical results, we consider the following simple example.

Example 2: $\mathcal{A} = \text{co } \widetilde{\mathcal{A}}, \quad \mathcal{B} = \text{co } \widetilde{\mathcal{B}}$, where

$$\widetilde{\mathcal{A}} = \left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix} \right\}, \quad \widetilde{\mathcal{B}} = \left\{ \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix} \right\}.$$

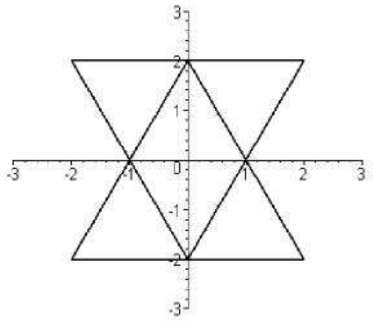

Figure 1:

It can be easily seen that the intersection has the representation

$$\mathcal{A} \cap \mathcal{B} = \text{co} \left\{ \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}.$$

This means that the points $(0,-2)^\top \in \widetilde{\mathcal{A}}$ and $(0,2)^\top \in \widetilde{\mathcal{B}}$ belong to the intersection of the convex hulls of the sets $\mathcal{A}$ and $\mathcal{B}$.

All the following results relate to Example 2. The algorithm terminates if $\hat{\varrho}(\mathcal{M}_k, \mathcal{M}_{k-1}) < 0.1$, where $\mathcal{M}_k = \mathcal{A}_k \cup \mathcal{B}_k$ and $\hat{\varrho}(A,B)$ denotes the modified

Hausdorff distance between the sets A and B considering only the "generating" points of these sets as well as all the points occurring during the iterations, but not all points belonging to the convex hulls. Consequently, the true Hausdorff distance between the sets $\mathcal{M}_k$ and $\mathcal{M}_{k-1}$ or between the sets $\mathcal{M}_k$ and $\mathcal{N} = \mathcal{A} \cap \mathcal{B}$ is (much more) smaller in general (cf. Fig. 3 below). For minimizing the function $\overline{F}_{\mathcal{AB}}$ and $\widetilde{F}_{\mathcal{AB}}$, resp., the generalized steepest descent method from [2] is used. In the exact penalty approach, the parameter λ is set to $\lambda = 3$. For finding an appropriate step size the Armijo rule with $T = 1$, $\beta = 0.5$, $\sigma = 0.25$ is applied. The program is implemented in C++.

The exact penalty approach

k	Points in $\mathcal{A}_k$	Points in $\mathcal{B}_k$	$\hat{\varrho}(\mathcal{M}_{k-1}, \mathcal{M}_k)$	$\hat{\varrho}(\mathcal{M}_k, \mathcal{N})$
0	3	3	–	2.23607
1	4	4	2.01475	2
2	6	6	0.412272	2
3	8	8	7.4476E−16	2

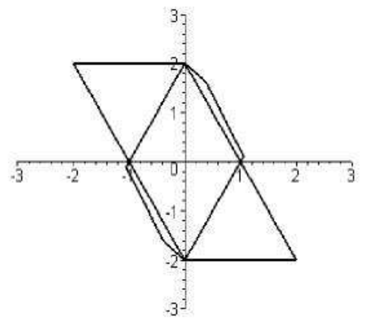

Figure 2:

In Fig. 2 it can be clearly seen that the intersection found considerably differs from the exact intersection. The reason is based on the fact that in minimizing the quasidifferentiable function $\overline{F}_{\mathcal{AB}}$ the evaluated direction is locally optimal. A way out consists in the variation of the starting vector for the minimization problem if necessary. Starting points in $\mathbb{R}^2$ could be e.g. $(1,0)^\top$, $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^\top$, $(0,1)^\top$, $(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^\top$. Of course, in higher dimensions the number of reasonable starting points increases considerably. By means of this modification we get the following results.

k	Points in $\mathcal{A}_k$	Points in $\mathcal{B}_k$	$\hat{\varrho}(\mathcal{M}_{k-1}, \mathcal{M}_k)$	$\hat{\varrho}(\mathcal{M}_k, \mathcal{N})$
0	3	3	–	2.23607
1	4	4	2.01475	2
2	6	6	1.71163	1.00552
3	10	10	0.177509	1.00552
4	18	18	0.000125963	1.00552

Thanks to this modification, the intersection is approached in a much better way (see Fig. 3, left).

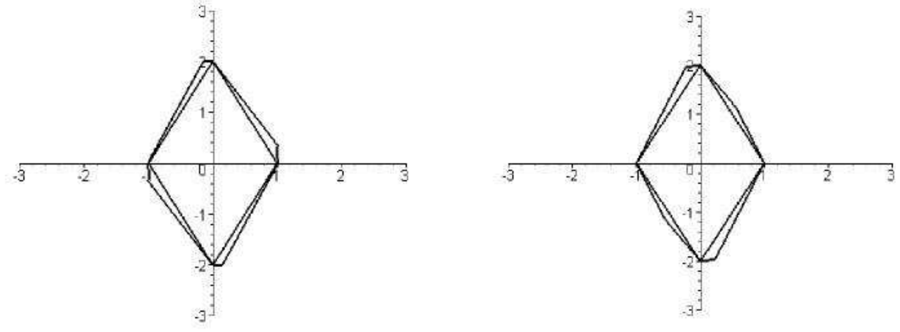

Figure 3:

The gradient projection method

Using this method, very similar results are obtained as in the case of exact penalty methods (cf. Fig. 3, right).

k	Points in $\mathcal{A}_k$	Points in $\mathcal{B}_k$	$\hat{\varrho}(\mathcal{M}_{k-1}, \mathcal{M}_k)$	$\hat{\varrho}(\mathcal{M}_k, \mathcal{N})$
0	3	3	–	2.23607
1	4	4	1.95043	2
2	6	6	1.76337	0.978573
3	10	12	1.02132	1.02132
4	30	30	0.243647	1.02132
5	58	58	0.00432817	1.02132

Summarizing, it can be stated that for sets generated by a relatively small number of points Algorithm IDENT being a modification of the algorithm from [3] works sufficiently well and yields an approximation of the intersection of the considered two sets. At the same time it should be said that a drawback of the algorithm is the rapidly increasing number of points in $\mathcal{A}_k$ and $\mathcal{B}_k$. This

is due to the fact that there occur points which arise from linear combinations of other points. Among them are many "superfluous" points often lying along a straight line (this applies to problems in $\mathbb{R}^2$).
In trying to apply the presented algorithm to real-life problems from breast cancer diagnostic (see www.cs.wisc.edu) containing 569 vectors of dimension 32, we unfortunately had to realize that because of the fast increase of points the used algorithm is not very well-suited for the identification of points. Thus the memory and the CPU time needed are very high. The suggestions of Demyanov in [4] to "clear" the sets $\mathcal{A}_k$ and $\mathcal{B}_k$ (i. e. to delete superfluous points) or to ignore the linear combinations could not significantly improve the behavior of the algorithm. Consequently, further investigations concerning suitable criterion functions and appropriate algorithms of nondifferentiable optimization are necessary.

References

[1] Demyanov V. F., Rubinov A. M.: *Quasidifferential Calculus*, Optimization Software, Inc., Publication Division, New York 1986.

[2] Luderer B., Weigelt J.: *A solution method for a special class of nondifferentiable unconstrained optimization problems*, Computational Optimization and Applications 24 (2003), 83–93.

[3] Demyanov V.F., Astorino A., Gaudioso M.: *Nonsmooth problems in mathematical diagnostics*, Advances in Convex Analysis and Global Optimization (Pythagorion, 2000), Nonconvex Optim. Appl. 54, Kluwer Acad. Publ., Dordrecht 2001, 11–30.

[4] Demyanov V.F.: *On the identification of points of two convex sets*, Vestnik St. Petersburg University, Mathematics 34 (2001) 3, 14–20.

[5] Han S.P., Mangasarian O.L.: *Exact penalty functions in nonlinear programming*, Mathematical Programming 17 (1979), 251–269.

[6] Rosen J. B.: *The gradient projection method for nonlinear programming, Part I. Linear constraints*, J. Soc. Indust. Appl. Math. 8 (1960), 181–217.

[7] Rosen J. B. *The gradient projection method for nonlinear programming, Part II. Nonlinear constraints*, J. Soc. Indust. Appl. Math. 9 (1961), 514–532.