

On numerical solving the spherical separability problem

M. Gaudioso¹ · T. V. Gruzdeva² · A. S. Strekalovsky²

Received: 30 October 2014 / Accepted: 3 June 2015 / Published online: 12 July 2015
© Springer Science+Business Media New York 2015

Abstract The separation problem of two sets, whose convex hulls have a nonempty intersection, is considered. In order to find a solution of the problem algorithms of local and global search are developed. The efficiency of the algorithms is demonstrated by computational simulations on test examples.

Keywords Spherical separation · D.C. function · Nonsmooth optimization · Local search · Global search

1 Introduction

During the last decades the attention of specialists has been attracted to theoretical and practical problems of systems analysis, control and information processing from various application areas, where a necessity to classify (to decompose) finite sets of data arises that can be often reduced to a solution of separability problems.

The problem of separation of sets, whose convex hulls have a nonempty intersection, represents, probably, one of the attractive objects of real-life interest. Such problems necessitate some more general and complicated concepts of separability than this one of linear splitting. Various definitions of generalized separability have been proposed to satisfy the demands from application areas [1–9]. Among others let us mention the employing k -functions and

✉ T. V. Gruzdeva
gruzdeva@icc.ru

M. Gaudioso
gaudioso@deis.unical.it

A. S. Strekalovsky
strekal@icc.ru

¹ Department of Computer Science, Modeling, Electronics and Systems Engineering (DIMES), University of Calabria, 87036 Rende, CS, Italy

² Matrosov Institute for Systems Dynamics and Control Theory, SB of RAS, 664033 Irkutsk, Russia

the majority committees [1,4]. Polyhedral separability is shown itself as well rather relevant from theoretical view-point and suitable for practitioners [2,3].

Among such approaches we recall the seminal contributions by Mangasarian [2] and Rosen [5], and the fundamental breakthrough due to the introduction the Support Vector Machine (SVM) method [6,7]. Moreover, the field of data sets classification is enlarging ceaselessly and grasps new areas [4–11]. In addition, all separation problems turn out to be nonconvex, and therefore they need new mathematical apparatus to find a global solution, in particular, to escape local pits provided by local search algorithms including the classical ones (conjugate gradients methods, Newtonian ones, SQP ones, IMP etc).

On the other hand, all theoretical proposals have to be verified by a practical relevance and a possible issue to solve numerically the corresponding nonconvex extremum's problems. A comparison of computational efficiency of global search algorithms such as Brunch & Bounds methods, cuts schemes committee's methodology etc, gives rise to new ideas for constructing special procedures for solving classification and separation problems.

In the paper we consider the problem of spherical binary separability [8,9,11], which consists in finding a sphere, dissectioning the sets \mathcal{A} and \mathcal{B} in the finite-dimensional space \mathbb{R}^n . Since it is unknown in advance whether the sets may be separated with a sphere or not, the minimization of the classification error function arises here naturally [10]. This minimization problem turns out to be nonsmooth and nonconvex [12–16], that leads us to a nonconvex variational problem.

Nonconvex problems with nondifferentiable functions are of special interest from the viewpoint of mathematical investigations because of their additional complexity and, on the other side, since they have a wide field of applications. Lately, the modelling of many real-life problems [17–19] have led to nondifferentiable nonconvex optimization problems.

At the same time, in nonconvex problems, the direct application of standard methods may have unpredictable consequences [13,15,16,20,21], and sometimes may even distract one from the desired solution. So, it seems to be quite natural (but hardly ever grounded) the reaction of the specialists propagating methods of direct selection—such as the method of branches and bounds (and cuts methods), which, as known, suffer the curse of dimension, when the volume of computations grows exponentially side by side with the growth of the problems dimension [13,15,16]. We are sure, there exists also another way of solving nonconvex problems of high dimension.

Further the spherical separability problem for two sets can be viewed as a problem of D.C. minimization [13–15]:

$$F(x) = g(x) - h(x) \downarrow \min_x, \quad x \in D, \quad (\mathcal{P})$$

where $g(\cdot)$, $h(\cdot)$ are convex and not necessarily differentiable functions, and set $D \subset \mathbb{R}^n$ is also convex. From now on let us suppose that the function $F(\cdot)$ over \mathbb{R}^n is bounded from below:

$$\inf(F, \mathbb{R}^n) > -\infty. \quad (\mathcal{H})$$

As well-known [13,15,16], the A.D.Alexandrov's functions (or the (D.C.) functions, which may be represented as a difference of two convex functions), form a linear space, which is dense in the space of continuous functions (in the topology of homogeneous convergence on compacts). So, problems of D.C. programming represent a rather large and, besides, very attractive class of optimization problems, for which we developed the theory of global search [13,15].

According to the theory developed on the base of global optimality conditions (GOC), the process of seeking for a global solution in nonconvex optimization problems consists of the two principal stages: (i) a special local search, which takes into account the structure of the problem under scrutiny [14], and (ii) the procedures (based on the global optimality conditions [13, 15]) of escaping critical points (provided by the local search).

The paper is organized as follows. In Sect. 3 we consider the generalization of the special local search method for solving D.C. minimization problems for the nonsmooth case. In addition, the convergence conditions of the method under investigation, and the stopping criteria are also substantiated. In Sect. 4 the proposed method is verified on the test problems of classification [22–24]. After that, on the base of global optimality conditions [13, 15] we develop a global search strategy for nonsmooth problems of D.C. minimization. In Sect. 6 we present the global search algorithm and the average of the tenfold cross-validation results of its computational testing on classification problems from [22–24].

2 Problem statement

Consider two nonempty and disjoint finite sets $\mathcal{A} = \{a^1, \dots, a^M\}$ and $\mathcal{B} = \{b^1, \dots, b^N\}$ in the space \mathbb{R}^n . Further let us denote the sphere $S(x, r)$ of radius $r > 0$ centered at a point $x \in \mathbb{R}^n$.

Then the sets \mathcal{A} and \mathcal{B} are defined to be separated by a sphere $S(x, r)$, if the following inequalities hold

$$\| a^i - x \| \leq r \quad \forall i = 1, \dots, M; \tag{1}$$

$$\| b^j - x \| \geq r \quad \forall j = 1, \dots, N; \tag{2}$$

where $\| \cdot \|$ is the Euclidean norm in the space \mathbb{R}^n .

The points of the sets \mathcal{A} and \mathcal{B} for which inequalities (1) and (2) are respectively satisfied, are called the well-classified points.

In the case, where one or more of the inequalities (1) or (2) are violated, we introduce the classification error, which is defined as follows:

$$\omega(x, r) \triangleq \sum_{i=1}^M \max\{0, \| a^i - x \|^2 - r^2\} + \sum_{j=1}^N \max\{0, r^2 - \| b^j - x \|^2\}. \tag{3}$$

Now one can formulate the problem of spherical separation with a sphere of the minimal radius as the problem of minimization of the classification error function with the variables $x \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$. It can be readily seen, that this problem is nonconvex, and the nonconvexity is generated by the terms $(-r^2)$ and $(-\| b^j - x \|^2)$. Taking into account the definition (3) of the error function, we obtain the problem of unconstrained nondifferentiable nonconvex optimization ($C > 0$):

$$F(x, r) \triangleq r^2 + C\omega(x, r) \downarrow \min_{x,r}. \tag{P_0}$$

In this problem we look for the priority between the two goals such as minimizing the radius and the classification error by means of varying the constant $C > 0$. It is clear that the function $F(x, r)$ is bounded from below by zero. So, the assumption (\mathcal{H}) in Problem (\mathcal{P}_0) is fulfilled.

Further, to perform the investigations we need an exact D.C. representation of the non-convex goal function $F(\cdot)$ in Problem (\mathcal{P}_0) . Using the following equality

$$\max\{0, f_1(x) - f_2(x)\} + f_2(x) - f_2(x) = \max\{f_1(x), f_2(x)\} - f_2(x),$$

we immediately obtain the D.C. representation as follows [10]:

$$F(x, r) = g(x, r) - h(x, r), \tag{4}$$

where

$$g(x, r) \triangleq r^2 + C \sum_{i=1}^M \max\{r^2, \|a^i - x\|^2\} + C \sum_{j=1}^N \max\{r^2, \|b^j - x\|^2\},$$

$$h(x, r) \triangleq CMr^2 + C \sum_{j=1}^N \|b^j - x\|^2. \tag{5}$$

In the next section we consider a special local search method employing this D.C. decomposition.

3 A special local search method

It is well-known [13–15,25,26] that in nonconvex problems, in particular, in D.C. programming problems, the classical convex optimization methods (as conjugate gradient, Newtonian, trust region methods etc.) turn out to be inefficient, in general, for reaching a global solution. In order to develop a search for a local solution to Problem (\mathcal{P}) we are going to apply the well-known DC-Algorithm (DCA) [13–15,27,28]. As known, it consists, first, in linearizing, at a current point, the function $h(\cdot)$ which defines the basic non-convexity of Problem (\mathcal{P}) , and minimizing so constructed convex linearized problem. After that we carry out consecutive solution of these linearized problems. This algorithm provides critical points, that can be proved by using only tools of convex analysis [13–15,29–31].

Now, let us describe the special local search method (SLSM) for a nonsmooth D.C. minimization Problem (\mathcal{P}) .

Let us given a starting iterate $x^0 \in \mathbb{R}^n$. Suppose, a point $x^s \in D$ and a subgradient $x_s^* \in \partial h(x^s)$ are provided. Then we find $x^{s+1} \in D$ as an approximate solution to the linearized problem as follows

$$\Phi_s(x) = g(x) - \langle x_s^*, x \rangle \downarrow \min_x, \quad x \in D, \tag{P\mathcal{L}_s}$$

so that the linearization is produced at the point x^s . More precisely, it means that the next iterate x^{s+1} satisfies the inequality as follows

$$g(x^{s+1}) - \langle x_s^*, x^{s+1} \rangle \leq \inf_{x \in D} \{g(x) - \langle x_s^*, x \rangle\} + \delta_s, \tag{6}$$

where a sequence $\{\delta_s\}$ fulfils the following conditions

$$\delta_s \geq 0, \quad s = 0, 1, 2, \dots; \quad \sum_{s=0}^{\infty} \delta_s < \infty. \tag{7}$$

Note, that the linearized problem $(\mathcal{P}\mathcal{L}_s)$ turns out to be convex, meanwhile Problem (\mathcal{P}) was a nonconvex one.

Let us denote the optimal value of the linearized problem $(\mathcal{P}\mathcal{L}_s)$ by $\mathcal{V}(\mathcal{P}\mathcal{L}_s)$, so that

$$\mathcal{V}(\mathcal{P}\mathcal{L}_s) := \inf_x \{g(x) - \langle x_s^*, x \rangle \mid x \in D\}. \tag{8}$$

Theorem 1 [13, 15] *Suppose, that the goal function $F(\cdot)$ of Problem (\mathcal{P}) is bounded from below, and one can find a subgradient $x^* \in \partial h(x)$ of the function $h(\cdot)$ at any point $x \in D$, where D is a closed convex set from \mathbb{R}^n .*

Then

i) *the sequence $\{x^s\}$ generated by the rule (6), (7), where $x_s^* \in \partial h(x^s)$, satisfies the condition:*

$$\lim_{s \rightarrow \infty} \{ \inf_x [g(x) - g(x^{s+1}) - \langle x_s^*, x - x^{s+1} \rangle] \} = 0, \tag{9}$$

or what is the same

$$\lim_{s \rightarrow \infty} [\mathcal{V}(\mathcal{P}\mathcal{L}_s) - \Phi_s(x^{s+1})] = 0.$$

ii) *Furthermore, if the sequences $\{x^s\}$ and $\{x_s^*\}$ are converging in concordance, i.e.*

$$x^s \rightarrow z \in D, \quad x_s^* \rightarrow z^* \in \partial h(z), \quad x_s^* \in \partial h(x^s), \tag{10}$$

then the limit point z of the sequence $\{x^s\}$ satisfies the condition

$$\inf_{x \in D} \{g(x) - g(z) - \langle z^*, x - z \rangle\} = 0, \quad z^* \in \partial h(z), \tag{11}$$

or what is the same $\mathcal{V}(\mathcal{P}\mathcal{L}_z) = \Phi_(z) \triangleq g(z) - \langle z^*, z \rangle$.*

Thus, the point z in Theorem 1 is a solution to the linearized problem

$$g(x) - \langle z^*, x \rangle \downarrow \min_x \quad x \in D, \quad z^* \in \partial h(z). \tag{P}\mathcal{L}_z$$

Definition 1 i) A point z is called critical if it satisfies condition (11), i.e. it is a solution to the linearized problem $(\mathcal{P}\mathcal{L}_z)$ with a subgradient $z^* \in \partial h(z)$.

ii) A point x^s is an approximate τ -critical if it is a τ -solution to Problem $(\mathcal{P}\mathcal{L}_s)$ with a subgradient $x_s^* \in \partial h(x^s)$, i.e.

$$g(x^s) - \langle x_s^*, x^s \rangle \leq \inf_{x \in D} \{g(x) - \langle x_s^*, x \rangle\} + \tau.$$

Now let us investigate the issue related to a stopping criterion for the presented method. From (6) due to convexity of the function $h(\cdot)$ it follows

$$\begin{aligned} -\delta_s &\leq \inf_{x \in D} \{g(x) - \langle x_s^*, x \rangle\} - g(x^{s+1}) + \langle x_s^*, x^{s+1} \rangle \\ &\leq g(x^s) - g(x^{s+1}) + \langle x_s^*, x^{s+1} - x^s \rangle \\ &\leq g(x^s) - g(x^{s+1}) + h(x^{s+1}) - h(x^s) = F(x^s) - F(x^{s+1}), \end{aligned}$$

which can be considered as a hint for the convergence proof of the method. Moreover, it suggests that one of the following inequalities can be employed as a stopping criterion for SLSM (together with $\delta_s \leq \tau/2$):

$$\left. \begin{aligned} F(x^s) - F(x^{s+1}) &\leq \tau/2, \\ \Phi_s(x^s) - \Phi_s(x^{s+1}) &\triangleq g(x^s) - g(x^{s+1}) + \langle x_s^*, x^{s+1} - x^s \rangle \leq \tau/2, \end{aligned} \right\} \tag{12}$$

where $F(\cdot)$ and $\Phi_s(\cdot)$ are the objective functions of Problems (\mathcal{P}) and $(\mathcal{P}\mathcal{L}_s)$, respectively, and τ is a given accuracy, and $x_s^* \in \partial h(x^s)$.

If one of the inequalities (12) is fulfilled, it can be easily shown that point x^s turns out to be τ -critical to Problem (P), when $\delta_s \leq \tau/2$. Indeed, (12) together with inequality (6) imply that

$$g(x^s) - \langle x_s^*, x^s \rangle \leq \tau/2 + g(x^{s+1}) - \langle x_s^*, x^{s+1} \rangle \leq \inf_{x \in D} \{g(x) - \langle x_s^*, x \rangle\} + \tau/2 + \delta_s.$$

Therefore, if $\delta_s \leq \tau/2$, then the point x^s is a τ -solution to Problem $(\mathcal{P}\mathcal{L}_s)$.

Thus, in the nonsmooth optimization Problem (P) SLSM described above provides for an approximate critical point. Henceforth below we assume $D = \mathbb{R}^n$.

4 Testing the local search method (DCA)

The local search algorithm just presented above has been tested on nine test problems from [22–24]. The computational experiment has been conducted on the computer Intel Pentium 4 CPU 3 GHz.

Further the barycenter of set \mathcal{A} :

$$x^0 = (x_1^0, \dots, x_n^0), \quad x_i^0 = \frac{a_i^1 + \dots + a_i^M}{M}, \quad i = 1, \dots, n, \tag{13}$$

has been chosen as a starting point for the center of a possibly separating sphere, and the solution to Problem (\mathcal{P}_0) with the fixed initial center x^0 (corresponding to the minimization of the radius) has been chosen as the starting radius r_0 .

On the other hand, the SLSM’s framework requires to solve at each iteration a convex Problem $(\mathcal{P}\mathcal{L}_s)$. To this end we have employed Shor’s r -algorithm [18]. It is based on the operation of stretching of the space in the direction of the difference of two sequential subgradients.

In addition, we have adopted the tenfold cross-validation protocol, which consists in splitting the dataset of interest into ten equally sized pieces. Nine of them are in turn used as a training set and the remaining one as a testing set. The correctness of the following computational simulation can be estimated by the total percentage of well classified points (of both sets \mathcal{A} and \mathcal{B}) when the algorithm stops.

Table 1 represents the average of the tenfold cross-validation results of computational testing of the local search algorithm, where we used the notations as follows: n is the space dimension; M and N stand for the numbers of points to be classified in the sets \mathcal{A} and \mathcal{B} , respectively; C stands for the values of parameter C in the goal function of Problem (\mathcal{P}_0) ; $F_0 = F(x^0, r_0)$ is the starting value of the goal function of Problem (\mathcal{P}_0) ; $F(z)$ is the value of the function at the critical point ($z = (x, r)$) provided by SLSM; $iter$ is the number of Linearized Problems solved (iterations of SLSM); $time$ is the CPU time of computing solutions (seconds); % stands for the percentage of well classified points; % $[10]$ stands for the percentage of well-classified points in the paper [10].

As the stopping criterion for SLSM we have used the following inequalities (see (12)):

$$|\Phi_s(x^s) - \Phi_s(x^{s+1})| \leq \frac{\tau}{2}, \quad \delta_s \leq \frac{\tau}{2};$$

with the precision of $\tau = 10^{-3}$ (likewise in [10]).

The average number of iterations turned out to be 41, which corresponds to the number of linearized problems solved, and the average CPU time for solving one problem was 1.22 s.

From Table 1 it can be readily seen, that we have managed to enlarge the percentage of well-classified points in problems “ionosphere”, “g50c”, “g10n” only at the stage of local

Table 1 Local search

Name	n	M	N	C	F_0	F(z)	Iter	Time	%	% [10]
Cancer	9	23	443	0.1	2.15×10^2	1.70×10^2	42	10.59	94.71	95.71
				1	1.14×10^3	6.39×10^2	56	5.40	95.36	
				10	1.03×10^4	5.65×10^3	54	4.67	95.51	
				100	1.02×10^5	5.24×10^4	44	5.47	95.22	
Diagnost	30	375	212	0.1	5.26×10^5	5.13×10^5	23	74.14	85.97	89.82
				1	4.24×10^6	3.92×10^6	37	247.52	87.89	
				10	4.12×10^7	3.82×10^7	33	396.13	87.89	
				100	4.13×10^8	3.81×10^8	33	196.38	87.89	
Heart	13	214	83	0.1	4.02×10^1	3.43×10^1	8	7.47	75.00	80.33
				1	2.63×10^2	1.79×10^2	12	6.72	79.00	
				10	2.49×10^3	1.63×10^3	17	1.43	78.33	
				100	2.47×10^4	1.66×10^4	14	0.36	77.00	
Pima	8	268	500	0.1	4.34×10^5	3.75×10^5	59	31.99	61.43	68.70
				1	4.28×10^6	3.63×10^6	60	21.95	61.95	
				10	4.27×10^7	3.64×10^7	54	12.94	61.40	
				100	4.27×10^8	3.65×10^8	35	6.76	61.04	
Ionosphere	34	225	126	0.1	4.72×10^1	2.34×10^1	15	82.24	82.22	72.00
				1	3.83×10^2	8.32×10^1	20	136.50	89.17	
				10	3.74×10^3	6.60×10^2	19	78.67	89.44	
				100	3.73×10^4	6.82×10^3	18	32.37	89.16	
Sonar	60	97	126	0.1	6.14×10^0	4.68×10^0	14	13.92	66.67	69.05
				1	4.76×10^1	2.38×10^1	26	9.83	69.52	
				10	4.61×10^2	2.09×10^2	33	3.59	70.00	
				100	4.61×10^3	2.09×10^3	29	2.97	70.95	
Galaxy	14	2082	2110	0.1	1.10×10^1	7.58×10^0	73	37.27	87.41	93.79
				1	1.09×10^2	6.75×10^1	201	102.50	88.81	
				10	1.08×10^3	6.66×10^2	185	60.12	88.79	
				100	1.08×10^4	7.10×10^3	46	19.66	87.10	
g50c	50	275	275	0.1	1.58×10^2	1.09×10^2	16	57.94	83.09	72.96
				1	1.09×10^3	4.43×10^2	33	65.00	85.17	
				10	1.04×10^4	4.40×10^3	25	20.42	83.64	
				100	1.03×10^5	5.08×10^4	22	3.37	80.36	
g10n	10	274	276	0.1	7.23×10^1	4.61×10^1	24	20.13	80.00	81.04
				1	6.29×10^2	2.11×10^2	33	18.25	85.82	
				10	6.20×10^3	1.81×10^3	34	11.07	86.36	
				100	6.19×10^4	2.25×10^4	33	3.19	82.55	

search. It is easy to see that the results of Table 1 can be viewed as more promising than these ones from [10]. It may be assessed so, in particular, because of the choice of Shor’s r -algorithm [18] for solving Problem $(\mathcal{P}\mathcal{L}_s)$ instead of the subroutine NCVX [8], which

implements a bundle type algorithm for solving the nonsmooth unconstrained optimization problems as it was done in [10]. On the other hand, the principal effect is attained due to the employing of the general framework of SLSM.

5 Optimality conditions and the global search strategy

Let us recall the fundamental result of global search theory for d.c. programming.

Theorem 2 [13, 15] *Suppose that*

$$\exists v \in \mathbb{R}^n : F(v) > F(z) \stackrel{\Delta}{=} \zeta. \tag{14}$$

Then point $z \in \mathbb{R}^n$ is a global solution to Problem (P) if and only if

$$\left. \begin{aligned} (a) \quad & \forall (y, \beta) \in \mathbb{R}^{n+1} : \beta = h(y) + \zeta, \\ (b) \quad & g(y) \leq \beta \leq \sup(g, \mathbb{R}^n), \quad \forall y^* \in \partial h(y) : \\ (c) \quad & g(y) - \beta \geq \langle y^*, x - y \rangle \quad \forall x \in \mathbb{R}^n. \end{aligned} \right\} \tag{E}$$

Conditions (E) are related to classical optimality conditions, employ the idea of linearization with respect to the basic nonconvexity and possess a constructive (algorithmic) property (in case when variational inequality (c) of conditions (E) does not hold, there exists a rule of constructing a feasible point, which is better than the point z under scrutiny)[13, 15].

Now introduce the following function

$$\begin{aligned} \varphi(z) = \sup_{x, y, \beta, y^*} \{ & \langle y^*, x - y \rangle + \beta - g(x) \mid \beta = h(y) + \zeta, \\ & g(y) \leq \beta \leq \sup(g, \mathbb{R}^n), \quad y^* \in \partial h(y) \}. \end{aligned} \tag{15}$$

Since $\beta = \beta_0 \stackrel{\Delta}{=} g(z)$ for $y = z$, the following inequality obviously holds

$$0 = \beta_0 + \langle z^*, z - z \rangle - g(z) \leq \varphi(z) \quad \forall z \in \mathbb{R}^n \quad \forall z^* \in \partial h(z).$$

Hence

$$\varphi(z) \geq 0 \quad \forall z \in \mathbb{R}^n. \tag{16}$$

Therefore, conditions (E) are equivalent to the equality $\varphi(z) = 0$. Furthermore, optimality conditions for the minimizing sequences can be presented in the following form.

Theorem 3 [13, 15] *If a sequence $\{z^k\}$ is minimizing to Problem (P) ($\{z^k\} \in \mathcal{M}$), then*

$$\lim_{k \rightarrow \infty} \varphi(z^k) = 0. \tag{17}$$

If, in addition, the following condition holds

$$\exists v \in \mathbb{R}^n, \exists \chi > 0 : F(v) \geq F(z^k) + \chi, \quad k = 0, 1, 2, \dots, \tag{18}$$

then the optimality condition (17) becomes sufficient for the sequence $\{z^k\}$ to be minimizing ($\{z^k\} \in \mathcal{M}$) in Problem (P).

The proof of this theorem does not principally differ from the proof in the differentiable case [13].

The properties of Optimality Conditions (E) allow one to construct algorithms for solving D.C. minimization problems, which presume two principal stages, as follows:

- (a) local search, which provides for an approximately critical point z^k with the corresponding value of the goal function $\zeta_k \triangleq F(z^k)$;
- (b) procedures of escaping critical points, which are based on the Global Optimality Conditions (GOC) (\mathcal{E}).

The latter procedure can be represented as a chain of the following operations [13, 15, 29]:

- (1) Choose a number $\beta : \inf(g, \mathbb{R}^n) \leq \beta \leq \sup(g, \mathbb{R}^n)$. It is possible to choose an initial β_0 , for instance, which is equal to $g(z^k)$ ($\beta_0 = g(z^k)$, $\zeta_k = F(z^k) = g(z^k) - h(z^k)$).
- (2) Construct a finite approximation

$$\mathcal{A}_k(\beta) = \{v^1, \dots, v^{N_k} \mid h(v^i) = \beta - \zeta_k, i = 1, \dots, N_k, N_k = N_k(\beta)\}$$

of the level surface $\{h(x) = \beta - \zeta_k\}$ of the function $h(\cdot)$, which generates the basic nonconvexity in Problem (\mathcal{P}).

- (3) Find a δ_k -solution \bar{u}^i of the following Linearized Problem:

$$g(x) - \langle v_i^*, x \rangle \downarrow \min_x, x \in \mathbb{R}^n, \tag{P\mathcal{L}i}$$

where $v_i^* \in \partial h(v^i)$ is a subgradient of $h(\cdot)$ at the point v^i , so that

$$g(\bar{u}^i) - \langle v_i^*, \bar{u}^i \rangle - \delta_k \leq \inf_x \{g(x) - \langle v_i^*, x \rangle\}. \tag{19}$$

After that, apply SLSM starting at the point \bar{u}^i . Let u^i be a point provided by SMLP.

- (4) Find a δ_k -solution w^i ($h(w^i) = \beta - \zeta_k$) of the level problem, i.e.

$$\langle w_i^*, u^i - w^i \rangle + \delta_k \geq \sup_{v, v^*} \{ \langle v^*, u^i - v \rangle \mid h(v) = \beta - \zeta_k, v^* \in \partial h(v) \}, \tag{20}$$

where $w_i^* \in \partial h(w^i)$.

- (5) Compute the value $\eta_k(\beta) := \eta_k^0(\beta) + \beta$, where

$$\eta_k^0(\beta) := \langle w_j^*, u^j - w^j \rangle - g(u^j) \triangleq \max_{i \in I_k} \{ \langle w_i^*, u^i - w^i \rangle - g(u^i) \}, \tag{21}$$

$w_i^* \in \partial h(w^i)$, $i \in I_k \triangleq \{i \in \{1, \dots, N_k\} \mid g(v^i) \leq \beta\}$.

If $\eta_k(\beta) > 0$ then it can be readily seen that the point u^j is better than z with respect to the goal function. In this case, one can pass to the next iteration of the algorithm and again execute the local search by starting from the point u^j . If still $\eta_k(\beta) \leq 0$, then a new value of β should be chosen, and the iteration should be repeated.

6 The global search algorithm and its numerical testing

Now let us describe the principal stages of the algorithm developed on the basis of the global search strategy (GSS) developed above for nonsmooth problem of D.C. minimization.

Note that the SLSM already described in Sect. 3 and tested in Sect. 4 has been applied as a local search method.

Further, it is necessary to determine the boundaries for the number β within the interval $[\beta_-, \beta_+]$, where $\beta_- \triangleq \inf(g, \mathbb{R}^n)$, $\beta_+ \triangleq \sup(g, \mathbb{R}^n)$.

However, it is clear that the search for numbers β_-, β_+ is equivalent to an unconstrained minimization and maximization of the function $g(\cdot)$, respectively. It can be readily seen from (5), that the function $g(\cdot)$ is convex and takes only nonnegative values. Therefore, one can

take $\beta_- = 0, \beta_+ = +\infty$, i.e. the interval of variation of parameter β is not bounded from above.

On the other side, it is possible to estimate the upper bound β_+ in another way, for example $\beta_+ := g(x^0, \bar{r})$, where x^0 is the barycenter of set \mathcal{A} from (13), and $\bar{r}^2 = \max_j \|b^j\|^2$. So, we have restricted the interval of variation of parameter β and we can change the number β , for example, as follows $\beta_{p+1} := \beta_p + \Delta\beta, \Delta\beta := (\beta_+ - \beta_-)/10$. In other words, the procedure of choosing the value $\Delta\beta$ can be carried out by means of one of well-known methods of one-dimensional optimization.

Further, the crucial element of the Global Search procedures consists in construction of an approximation of the level surface of the convex function $h(\cdot)$ which generates the basic nonconvexity in Problem (\mathcal{P}_0) . In particular, an approximation $\mathcal{A}_k(\beta)$ for each pair $(\beta, \zeta_k), \zeta_k = F(z^k)$, can be constructed by the rule as follows

$$v^l = z^k - \mu_l e^l, \quad l = 1, \dots, n, \tag{22}$$

where z^k is a current critical point; e^l is the l th unit vector of Euclidean basis of \mathbb{R}^n . A search for μ_l turns out to be rather simple and moreover analytical for the quadratic function $h(\cdot)$, because it reduces itself to the solution of a quadratic equation in one variable. The set (approximation) (22) has shown itself rather competitive [13] during the computational simulations.

Repeat, that later on stage 3 of GSS, instead of solving only one Linearized Problem, we additionally initiate SLSM, i.e. we solve a sequence of Linearized Problems until obtaining an approximate critical point. The structure of GSS shall not be destroyed by this change, and we can obtain a feasible point with some better and attractive properties. On stage 4 of GSS, instead of solving the level problem and the subsequent forming the value η_k , we execute direct comparison of the value of the goal function at the point u^i provided by the local search, with the value $\zeta_k \triangleq F(z^k)$. Besides, it can be readily seen that $\eta_k > 0$ if and only if point the point u^i is better than z^k with respect to the objective function [13].

Let us describe now the global search algorithm (GSA) step by step.

Let us given a starting point x^0 , and the accuracy $\tau = 10^{-3}$.

- Step 0. Set $k := 0, x^k := x^0, l := 1, \tau_k := 0.1$.
- Step 1. Starting from the point x^k and applying SLSM construct a τ_k -critical point z^k , such that $\zeta_k \triangleq F(z^k) \leq F(x^k)$.
- Step 2. Set $\beta := g(z^k)$.
- Step 3. Construct the point v^l of the approximation $\mathcal{A}_k(\beta)$ according to the rule (22), $h(v^l) = \beta - \zeta_k$.
- Step 4. Starting from the point v^l construct by means of the SLSM a $2\tau_k$ -critical point u^l .
- Step 5. If $\zeta_k > F(u^l)$ then set $x^{k+1} := u^l, \zeta_{k+1} := F(u^l), l := l + 1$. If $\tau_k > \tau$ then set $\tau_{k+1} := \frac{\tau_k}{2}$. After that set $k := k + 1$ and loop to Step 1.
- Step 6. If $\zeta_k \leq F(u^l)$ and $l < N$ then set $l := l + 1$ and loop to Step 3.
- Step 7. If $\zeta_k \leq F(u^l)$ and $l = N$, and $\tau_k > \tau$ then set $\tau_k := \frac{\tau_k}{2}, l := 1$ and loop to Step 1.
- Step 8. If $\zeta_k \leq F(u^l)$ and $l = N$, and $\tau_k \leq \tau$, and $\beta < \beta_+$ then set $\tau_k := \tau, \beta := \beta + \Delta\beta, l := 1$ and loop to Step 3.
- Step 9. If $l = N, \zeta_k \leq F(u^l) \forall \beta \in [\beta_-, \beta_+]$ (i.e. the one-dimensional search with respect to parameter β on the segment $[\beta_-, \beta_+]$ is over) then **Stop**: z^k is a feasible point provided by GSA.

Table 2 Global search

Name	C	F_0	F_*	$\omega(x^*, r_*)$	PL	Loc	Iter	Time
Cancer	0.1	2.15×10^2	1.65×10^2	577.29	896	90	4	1:25:53.15
	1	1.14×10^3	6.51×10^2	476.65	769	171	14	0:04:51.71
	10	1.03×10^4	4.33×10^3	474.05	1685	207	16	0:16:22.28
	100	1.02×10^5	4.39×10^4	481.89	551	234	15	0:07:13.16
Diagnost	0.1	5.26×10^5	4.94×10^5	3986749.01	2733	921	21	2:54:02.65
	1	4.24×10^6	3.59×10^6	3222243.79	2040	1786	52	1:25:36.09
	10	4.12×10^7	3.02×10^7	2905610.64	2334	1507	69	2:40:28.01
	100	4.13×10^8	3.17×10^8	3213153.73	1377	1245	32	0:52:11.48
Heart	0.1	4.02×10^1	3.37×10^1	174.68	176	104	5	0:14:15.02
	1	2.63×10^2	1.68×10^2	158.23	260	125	4	0:18:59.88
	10	2.49×10^3	1.59×10^3	157.98	638	143	4	0:12:08.27
	100	2.47×10^4	1.58×10^4	158.13	752	169	5	0:01:45.24
Pima	0.1	4.34×10^5	3.73×10^5	3544741.12	6450	200	18	0:19:01.97
	1	4.28×10^6	3.41×10^6	3312008.33	6673	336	35	0:04:58.33
	10	4.27×10^7	3.31×10^7	3292801.64	5878	304	32	0:03:19.89
	100	4.27×10^8	3.32×10^8	3314252.72	1477	152	9	0:02:07.73
Ionosphere	0.1	4.72×10^1	2.31×10^1	93.91	1210	272	3	5:43:11.91
	1	3.83×10^2	7.16×10^1	40.28	1173	510	10	1:29:43.21
	10	3.74×10^3	3.88×10^2	31.52	845	748	16	0:19:31.25
	100	3.73×10^4	4.26×10^3	42.32	816	235	17	0:14:59.94
Sonar	0.1	6.14×10^0	4.66×10^0	24.56	4522	588	5	3:49:07.66
	1	4.76×10^1	2.14×10^1	15.92	2495	1026	14	0:43:49.48
	10	4.61×10^2	1.49×10^2	14.09	2102	1404	16	0:07:03.32
	100	4.61×10^3	1.56×10^3	15.61	1662	1132	21	0:10:03.98
Galaxy	0.1	1.10×10^1	7.56×10^0	69.71	2581	84	2	6:52:16.41
	1	1.09×10^2	6.73×10^1	65.99	1885	112	3	2:33:52.38
	10	1.08×10^3	6.59×10^2	65.83	815	154	5	1:02:45.08
	100	1.08×10^4	6.70×10^3	67.03	401	140	5	0:03:11.88
g50c	0.1	1.58×10^2	1.05×10^2	342.64	2573	880	14	14:44:43.04
	1	1.09×10^3	3.08×10^2	215.93	10973	1850	31	2:52:26.15
	10	1.04×10^4	2.50×10^3	245.51	20178	2795	52	12:00:14.31
	100	1.03×10^5	3.01×10^4	299.19	2345	1813	43	0:46:58.29
g10n	0.1	7.23×10^1	4.57×10^1	240.59	154	103	6	0:02:45.91
	1	6.29×10^2	1.89×10^2	138.25	220	142	10	0:01:03.45
	10	6.20×10^3	1.36×10^3	130.05	379	178	14	0:00:30.81
	100	6.19×10^4	1.39×10^4	139.21	407	239	17	0:00:15.48

Table 2 represents the average results of computational testing of the GSA (using adopted tenfold cross-validation protocol as well as in the local search testing), where F_0 is the starting value of the cost function of Problem (\mathcal{P}_0); F_* is the best value of the goal function provided

Table 3 Average percentage of testing correctness and comparison

Name	n	M	N	Well class %		Win/loss
				GSA	DCA [10]	
Cancer	9	239	443	96.23	95.71	0
Diagnost	30	375	212	90.70	89.82	0
Heart	13	214	83	79.35	80.33	0
Pima	8	268	500	68.05	68.70	0
Ionosphere	34	225	126	90.28	72.00	+1,7
Sonar	60	97	126	80.04	69.05	+1
Galaxy	14	2082	2110	89.91	93.79	-0,4
g50c	50	275	275	87.46	72.96	+1,6
g10n	10	274	276	90.18	81.04	+1

by GSA; $\omega(x^*, r_*)$ is the classification error; PL is the number of Linearized Problems solved during the Global Search; Loc is the number of start-ups of the local search procedure in course of the global search; $iter$ is the number of iterations of GSA; $time$ is the CPU time of the program implementing GSA.

Note, that the the average number of different critical points from column $iter$, by which GSA passed improving, at the same time, the goal function of Problem (\mathcal{P}_0), is 18, while the average number of start-ups of the local search procedure from column Loc is 695.

As expected, in all test problems the biggest classification error $\omega(x^*, r_*)$ has been obtained at $C = 0.1$, what corresponds to the priority of minimizing radius of the sphere.

Next, Table 3 demonstrates a relative efficiency of GSA with adopted tenfold cross-validation protocol. Furthermore, in Table 3 one can observe the data related to the average percentage of well-classified points obtained with the algorithms (i) GSA and (ii) DCA [9, 10]. The last column in the table stands for the relative loss (“-”) or the relative win (“+”) as a percentage of the number of well-classified points compared to the results from [9, 10].

Note that despite the application of a rather simple approximation of the level surface and a rather simplified search of the parameter β , our GSA is able to increase the percentage of well-classified points in all the examples. In problems “ionosphere” and “g50c” the increment has turned out to be rather substantial: it was more than 1.7 and 1.6% well classified points correspondingly with respect to the results from [10]. As to Problems “sonar” and “g10n”, the percentage of well-classified points is more than 1 in comparison with the results from [9, 10]. And for the only one Problem “galaxy” the percentage of well-classified points is less than 0.4 in comparison with the results from [9, 10].

We have added n points $-v^l$, $l = n + 1, \dots, 2n$, in the approximation constructed at the Step 3 of the GSA, and tried to increase the percentage of well-classified points in Problem “galaxy”. The CPU time of the program implemented with the parameter $C = 10$ increased up to 2:29:31.68 and the percentage of well-classified points reached 90.36.

7 Conclusion

In the paper we considered the binary separability problem which consists in splitting two finite-dimensional sets by means of a sphere. This problem is formulated as minimization of the classification error function which is nonconvex and nonsmooth. Besides, the error

function can be represented as a difference of two convex functions, so the problem falls into D.C. optimization area [13, 15].

This feature opens the way of seeking for a global solution to the problem by means of the Theory of Global Search developed in [13–15]. The core of this theory is constructed by Global Optimality Conditions (GOC). The suitable for our case formulation of GOC was mentioned in the paper. On the base of the Theory of Global Search the Global Search Algorithm has been developed.

The computational testing of developed algorithms has been carried out on well known classification test problems [22–24] with the tenfold cross-validation technique.

The comparison of computational efficiency of the developed approach has been done with results from [10]. We plan to continue the comparison in order to have more large field of testing results. Nevertheless, one can see a comparative effectiveness of global search algorithm developed above, the implementation of which was a first attempt. We ponder now about how it can be improved taking into account the promising results of computational experiments. At the same time we plan to enlarge the field of test examples but till present we have found only [32], which together with [10] was apparently sufficient for our first attempt to attack spherical separability.

Acknowledgments The authors expresses gratitude to anonymous referees for their valuable remarks in improving the paper.

References

1. Eremin, I.I.: Theory of linear optimization. Ser. Inverse and Ill-Posed Problems. VSP, Utrecht (2002)
2. Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. *Oper. Res.* **13**(3), 444–452 (1965)
3. Astorino, A., Gaudio, M.: Polyhedral separability through successive LP. *J. Optim. Theory Appl.* **112**(2), 265–293 (2002)
4. Mazurov, V.D., Khachay, M.Yu., Rybin, A.I.: Committee constructions for solving problems of selection, diagnostics and prediction. In: *Proceedings of the Steklov Institute of Mathematics*, vol. 1, pp. S67–S101 (2002)
5. Rosen, J.B.: Pattern separation by convex programming. *J. Math. Anal. Appl.* **10**, 123–134 (1965)
6. Vapnik, V.: The nature of the statistical learning theory. Springer, New York (1995)
7. Mangasarian, O.L., Musicant, D.R.: Data discrimination via nonlinear generalized Support Vector Machines. In: Ferris, M.C., Mangasarian, O.L., Pang, J.S. (eds.) *Applications and Algorithms of Complementarity*. Kluwer, Dordrecht (2000)
8. Astorino, A., Gaudio, M.: A fixed-center spherical separation algorithm with kernel transformations for classification problems. *Comput. Manag. Sci.* **6**, 357–372 (2009)
9. Astorino, A., Fuduli, A., Gaudio, M.: Margin maximization in spherical separation. *Comput. Optim. Appl.* **53**, 301–322 (2012)
10. Astorino, A., Fuduli, A., Gaudio, M.: DC models for spherical separation. *J. Glob. Optim.* **48**(4), 657–669 (2010)
11. Tax, D.M.J., Duin, R.P.W.: Uniform object generation for optimizing one-class classifiers. *J. Mach. Learn. Res.* **2**, 155–173 (2002)
12. Sergeyev, YaD, Strongin, R.G., Lera, D.: *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer, New York (2013)
13. Strekalovsky, A.S.: *Elements of Nonconvex Optimization* (in Russian). Nauka Publ, Novosibirsk (2003)
14. Strekalovsky, A.S.: On local search in d.c. optimization problems. *Appl. Math. Comput.* **255**, 73–83 (2015)
15. Strekalovsky, A.S.: On solving optimization problems with Hidden nonconvex structures. In: Rassias, T.M., Floudas, C.A., Butenko, S. (eds.) *Optimization in Science and Engineering*, pp. 465–502. Springer, New York (2014)
16. Horst, R., Pardalos, P., Thoai, N.V.: *Introduction to Global Optimization*. Kluwer, Dordrecht (1995)

17. Demyanov, V.F., Vasiliev, L.V.: *Nondifferentiable Optimization*. Optimization Software Inc, New York (1985)
18. Shor, N.Z.: *Minimization Methods for Non-differentiable Functions*. Springer, New York (1985)
19. Pshenichny, B.N.: *Convex Analysis and Extremum Problems (in Russian)*. Nauka Publ, Moscow (1980)
20. Mascarenhas, W.F.: On divergence of line search methods. *Comput. Appl. Math.* **26**(1), 129–169 (2007)
21. Mascarenhas, W.F.: The BFGS algorithm with exact line searches fails for nonconvex functions. *Math. Program.* **99**(1), 49–61 (2004)
22. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases. <http://www.ics.uci.edu/mllearn/MLRepository.html>. Accessed 27 October (2014)
23. Odewahn, S., Stockwell, E., Pennington, R., Humphreys, R., Zumach, W.: Automated star/galaxy discrimination with neural networks. *Astron. J.* **103**, 318–331 (1992)
24. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. *Proceedings of the tenth international workshop on artificial intelligence and statistics*, pp. 57–64 (2005)
25. Hiriart-Urruty, J.-B., Lemarshal, C.: *Convex Analysis and Minimization Algorithms*. Springer, Berlin (1993)
26. Nocedal, J., Wright, S.J.: *Numerical optimization*. Springer, New York (1999)
27. Tao, P.D., Souad, L.B.: Algorithms for solving a class of non convex optimization. *Methods of subgradients*. In: Hiriart-Urruty, J.-B. (ed.) *Fermat days 85*, pp. 249–271. Elsevier Science Publishers B.V, North Holland (1986)
28. An, L.T.H., Tao, P.D.: The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems. *Ann. Oper. Res.* **133**, 23–46 (2005)
29. Strekalovsky, A.S., Orlov, A.V., Malyshev, A.V.: On computational search for optimistic solutions in bilevel problems. *J. Glob. Optim.* **48**(1), 159–172 (2010)
30. Gruzdeva, T.V., Strekalovsky, A.S.: Local search in problems with nonconvex constraints. *Comput. Math. Math. Phys.* **47**(3), 381–396 (2007)
31. Gruzdeva, T.V.: On a continuous approach for the maximum weighted clique problem. *J. Glob. Optim.* **56**(3), 971–981 (2013)
32. Astorino, A., Gaudio, M., Khalaf, W.: Edge detection by spherical separation. *Comput. Manag. Sci.* **11**(4), 517–530 (2014)