Minimizing piecewise-concave functions over polytopes

Manlio Gaudioso
DIMES, Università della Calabria, 87036 Rende (CS), Italia, gaudioso@dimes.unical.it,

Giovanni Giallombardo
DIMES, Università della Calabria, 87036 Rende (CS), Italia, giallo@dimes.unical.it,

Giovanna Miglionico
DIMES, Università della Calabria, 87036 Rende (CS), Italia, gmiglionico@dimes.unical.it,

We introduce an iterative method for solving linearly constrained optimization problems, whose nonsmooth nonconvex objective function is defined as the pointwise maximum of finitely many concave functions. Such problems often arise from reformulations of certain constraint structures (e.g., binary constraints, finite max-min constraints) in diverse areas of optimization. We state a local optimization strategy which exploits piecewise-concavity of the objective function giving rise to a linearized model corrected by a proximity term. In addition we introduce an approximate line-search strategy, based on a curvilinear model, which, similarly to bundle methods, can return either a satisfactory descent or a null-step declaration. Termination at a point satisfying an approximate stationarity condition is proved. We embed the local minimization algorithm into a Variable Neighborhood Search scheme and a Coordinate Direction Search heuristics, whose aim is to improve the current estimate of the global minimizer by exploring different parts of the feasible region. We finally provide computational results obtained on two sets of small- to large-size test problems.

Key words: Nonsmooth optimization; Minmax problems; Nonconvex optimization; Piecewise-concave

MSC2000 subject classification: 90C47; 65K05; 90C26

OR/MS subject classification: Primary: Programming: nondifferentiable; secondary: Nonlinear: Algorithms; Mathematics: Convexity

---

1. Introduction

We consider the following nonconvex nonsmooth optimization problem

$$\min_{x \in X \subset \mathbb{R}^n} f(x) \tag{1}$$

where

$$f(x) \triangleq \max_{i \in I} \; f_i(x),$$

and we assume that

- the *component functions* $f_i(\cdot)$, $i \in I = \{1, \ldots, m\}$, are concave and not necessarily differentiable, hence $f(\cdot)$ is a *piecewise-concave* function [50];
- the set X is a bounded polyhedron, i.e.,

$$X = \{x \in \mathbb{R}^n \,|\, Cx \leq b\}$$

for a given real-matrix C and a real-vector b of appropriate dimensions. We will refer in the sequel to problem (1) as to PCP, the Piecewise Concave Problem.

Such problem often appears in its equivalent piecewise-convex maximization version, and arises from reformulations of certain constraint structures in diverse areas of optimization. Consider, for an example, a combinatorial optimization problem whose binary decision variables u_i, $i = 1, \ldots, m$ are grouped into the vector $u \in \{0, 1\}^m$. Recalling that an equivalent continuous formulation of binary constraints is

$$u_i(1 - u_i) \leq 0, \;\; 0 \leq u_i \leq 1 \quad i = 1, \ldots, m,$$

then a compact reformulation for the binary constraint set can be obtained as

$$\varphi(u) \leq 0, \;\; u \in [0, 1]^m$$

where the function

$$\varphi(u) \triangleq \max_{1 \leq i \leq m} u_i(1 - u_i)$$

is piecewise-concave (see [20, 21] for further developments of this reformulation).

Another interesting example is the min-max-min reformulation of the adjustable robust counterpart of some two-stage stochastic programs with finite uncertainty set [4, 5, 47]. In this framework, two groups of decision variables, x and z, represent "here-and-now" and "wait-and-see" decisions, respectively, namely the decisions made before and after the uncertain parameters of the problem reveal their actual values. In fact, the cost function depends both on such decisions and on the realization of one from finitely many scenarios, i.e.,

$$f(x) = \max_{i \in I} \min_{z \in Z_i(x)} \phi_i(x, z),$$

where $Z_i(x)$ is the space of "wait-and-see" decisions, and $\phi_i(x, z)$ represents the cost associated to the decision couple (x, z) under the ith scenario. In particular, assuming that for each scenario index $i \in I$ the cost functions $\phi_i(x, z)$ are bilinear, i.e.,

$$\phi_i(x, z) \triangleq (b^{(i)} + A^{(i)}x)^\top z,$$

with $b^{(i)} \in \mathbb{R}^m$, $A^{(i)} \in \mathbb{R}^{m\times n}$, and that the sets X and $Z_i(x)$, $i \in I$, are bounded polyhedra, we observe that the objective function of the finite min-max-min problem

$$\min_{x\in X\subset\mathbb{R}^n} \max_{i\in I} \min_{z\in Z_i(x)\subset\mathbb{R}^m} \phi_i(x,z)$$

is the pointwise maximum of finitely many concave functions, hence it is piecewise concave; indeed, $\phi_i(x,z)$ is an affine function in x, and as a consequence $f_i(x) \triangleq \min_{z\in Z_i(x)} \phi_i(x,z)$ is concave, being the minimum of an (infinite) number of affine functions.

The literature on minmax problems is extremely rich. We mention here the historical books by Danskin [8] and by Demyanov and Malozemov [11], and as far as the numerical aspects are concerned, we recall [31, 45, 15]. A number of interesting aspects are treated in [18], and some recent proposals aimed at introducing the incremental/inexact approach in the convex minmax area are presented in [12, 26, 27, 28, 39, 42, 43]. For an up-to-date state of the art in nonsmooth optimization see [1], and for the specific case where inexact function evaluation is allowed see [13].

As far as the minimization of piecewise-concave (or maximization of piecewise-convex) functions is concerned, the literature is mainly focused on approaching the problem from the global optimization viewpoint, thus exploiting appropriate global optimality conditions [19] and iterative model refinements [22]. Localization properties of minima of such problems have been studied in several papers where generalizations of the concept of vertex are introduced [17, 30, 48, 50]. Applications to algorithm design for some relevant classes of combinatorial optimization problems have been described in [20, 21]. From a local optimization perspective, the minimization of piecewise-concave functions falls in the class of nonconvex nonsmooth optimization problems for which several nonsmooth methods have been proposed in the literature, see [2, 24, 25, 29, 33, 34] for some examples.

Our goal is to tackle primarily the local constrained minimization of a piecewise-concave function by means of an optimization algorithm which is inspired by the well known cutting plane approach to nonsmooth optimization. Although we adopt terminology and notations which are typical of the bundle [35] and conjugate subgradient methods [49], the approach we introduce is substantially different and strongly relies on the minmax nature of the problem.

The main feature of our local minimization algorithm is the way we exploit concavity of the component functions, in order to face nonconvexity of the objective function. For such method we prove termination at a point satisfying an approximate stationarity condition. Indeed, the method is based on some classic cutting plane idea, that is on the iterative construction of a model function as the pointwise maximum of finitely many affine pieces. In particular, the piecewise-affine model is obtained by linearizing at the *current point* the component functions that are active at a *bundle of different points* close to it. Due to piecewise concavity, the model locally overestimates the objective function, and this will turn out the key point for designing a bundle updating strategy, as well as a descent test where, unlike traditional approaches, actual descent is checked against that of a nonlinear model rather than a linear one.

The paper is organized as follows. In §2, we introduce the structure and some relevant theoretical foundations of the local minimization algorithm. In §3, we prove finite termination of the algorithm at a point satisfying an approximate stationarity condition. In §4, we report on the computational results obtained on two sets of small- to large-size test problems. Moreover, in order to assess possible embedding of the algorithm into a global optimization scheme, we test two global search methods, based on the Variable Neighborhood Search technique, and on a novel Coordinate Search heuristics, respectively. Finally, in §5 we present some conclusions.

2. Finding local minima of *PCP*

Problem (1) is a linearly-constrained nonlinear optimization problem, whose objective function is piecewise-concave, hence nonconvex and nonsmooth. The local minimization method we propose falls in the mainstream of the linearization methods for minmax [3, 31, 44], and benefits from some ideas which are at the basis of the well known class of bundle type algorithms, primarily devoted to solution of convex nonsmooth optimization problems (see [10, 14, 23, 35, 38, 40, 46] for some variants of bundle methods and [24, 25, 33, 34] for treatment of the nonconvex case).

In the remainder of the paper we adopt the following definitions and notations. By $I(x)$ we define the index set of active component functions at point x, that is

$$I(x) \triangleq \{i \in I |\, f(x) = f_i(x)\}.$$

By $I_\theta(x)$, for any positive scalar θ, we define the index set of the θ-active component functions at x, that is

$$I_\theta(x) \triangleq \{i \in I |\, f(x) - f_i(x) \le \theta\}.$$

By $T_X(x)$ we define the tangent cone (i.e., the cone of feasible directions) at any point x, that is

$$T_X(x) \triangleq \{d \in \mathbb{R}^n |\ c^{(r)\top} d \le 0,\ r \in R(x)\}$$

where $c^{(r)\top}$ is the rth row vector of C and $R(x)$ is the index set of constraints satisfied by equality at x.

We start by setting our problem-specific bundle-type notations. Assume that a set $X^{(J)} \triangleq \{x^{(1)}, x^{(2)}, \dots, x^{(k)}\}$ of feasible points is given, together with the corresponding index set $J \triangleq \{1, 2, \dots, k\}$. Moreover, assume that for every $j \in J$ an active component index $i_j \in I(x^{(j)})$ is selected, and denote by $I^{(J)} \triangleq \{i_1, \dots, i_k\}$ the set of all such indices.

We single out any point $y \in X^{(J)}$ to define our model as the following convex piecewise-affine function

$$\ell_y^{(J)}(x) \triangleq \max_{j\in J}\{f_{i_j}(y) + g^{(j)\top}(x-y)\}. \tag{2}$$

where $g^{(j)} \in \partial f_{i_j}(y)$, $j \in J$. Unlike the traditional bundle approach, we note here that each point $x^{(j)}$, $j \in J$, is used as an *index provider*, rather than a *subgradient provider*, since $g^{(j)}$ is a subgradient of the component function f_{i_j} at y instead of $x^{(j)}$.

In order to face nonconvexity of $f(x)$, our first aim is to exploit concavity of component functions to come up with some useful properties of $\ell_y^{(J)}(x)$. Note that concavity of functions f_is guarantees that for every $j \in J$ it holds:

$$f_{i_j}(x) \le f_{i_j}(y) + g^{(j)\top}(x-y), \quad \forall x \in \mathbb{R}^n. \tag{3}$$

Thus, if we define the *partial* max-function $f^{(J)}(x)$ as

$$f^{(J)}(x) \triangleq \max_{j\in J} f_{i_j}(x),$$

it follows from (3) that

$$f^{(J)}(x) \le \ell_y^{(J)}(x), \quad \forall x \in \mathbb{R}^n. \tag{4}$$

Moreover, we observe that

$$f^{(J)}(x) \le f(x), \quad \forall x \in \mathbb{R}^n \tag{5}$$

and

$$I^{(J)} = I \implies f^{(J)}(x) = f(x), \quad \forall x \in \mathbb{R}^n. \tag{6}$$

Summarizing, the *piecewise-affine* function $\ell_y^{(J)}(\cdot)$

- interpolates both the actual function $f(\cdot)$ and the *partial* max-function $f^{(J)}(\cdot)$ at y,
- is an upper approximation of the *partial* max-function,
- is an upper approximation of the actual function whenever $I^{(J)} = I$.

Now, performing the change of variables $x = y + d$, $d \in \mathbb{R}^n$, we come out with the following model function

$$h_y^{(J)}(d) \triangleq \ell_y^{(J)}(y+d) - f(y) = \max_{j\in J}\left\{g^{(j)\top}d - \epsilon_j\right\},$$

where $\epsilon_j \ge 0$, $j \in J$, is defined as

$$\epsilon_j \triangleq f(y) - f_{i_j}(y),$$

and define our bundle as the following set of triplets:

$$\mathcal{B} \triangleq \left\{\left(i_j, g^{(j)}, \epsilon_j, \right), \forall j \in J\right\},$$

where we remind that every pair $(g^{(j)}, \epsilon_j)$ depends on the selected point y (the *stability center* in the sequel).

Focusing now on the model function $h_y^{(J)}(\cdot)$, it follows from (4) that

$$f^{(J)}(y+d) - f(y) \leq h_y^{(J)}(d), \forall d \in \mathbb{R}^n \tag{7}$$

which in turn, combined with (6), implies that

$$I_J = I \implies f(y+d) - f(y) \leq h_y^{(J)}(d), \quad \forall d \in \mathbb{R}^n. \tag{8}$$

Therefore, we consider the model function $h_y^{(J)}(d)$ as an approximation of the difference function $f(y+d) - f(y)$, suitable to calculate the tentative displacement from the stability center y, in order to reduce the objective function value. In particular, we adopt a line-search approach along the direction d_y returned as the unique solution to the following quadratic program

$$\min_{d \in T_X(y)} h_y^{(J)}(d) + \frac{1}{2}\|d\|^2 \tag{9}$$

where d plays the role of a search direction rather than a displacement and, by analogy with bundle methods, a proximity term has been added to the model function $h_y^{(J)}(d)$ for stabilization purposes.

Problem (9) can be put in the following equivalent form

$$\begin{aligned} \min_{d \in \mathbb{R}^n, v \in \mathbb{R}} \; & v + \frac{1}{2}\|d\|^2 \\ & v \geq g^{(j)\top} d - \epsilon_j, \ \forall j \in J \\ & d \in T_X(y) \end{aligned} \tag*{$QP_y^{(J)}$}$$

in the sense that, denoting by (d_y, v_y) the optimal solution of $QP_y^{(J)}$, then there holds

$$v_y = h_y^{(J)}(d_y),$$

where $v_y \leq 0$ due to feasibility of the solution $(d, v) = (0, 0)$.

A useful insight can be achieved by focusing on the dual of $QP_y^{(J)}$:

$$\begin{aligned} \min_{\lambda, \mu} \; & \frac{1}{2}\left\| \sum_{j \in J} \lambda_j g^{(j)} + \sum_{r \in R(y)} \mu_r c^{(r)} \right\|^2 + \sum_{j \in J} \lambda_j \epsilon_j \\ & \sum_{j \in J} \lambda_j = 1, \\ & \lambda_j \geq 0, \forall j \in J, \\ & \mu_r \geq 0, \forall r \in R(y), \end{aligned} \tag*{$DP_y^{(J)}$}$$

where λ and μ denote the vectors obtained by grouping the variables $\lambda_j, j \in J$ and $\mu_r, r \in R(y)$. Assume that (d_y, v_y) and (λ^y, μ^y) are a couple of primal-dual optimal solutions. The following equalities hold:

$$d_y = -\left(\sum_{j\in J} \lambda_j^y g^{(j)} + \sum_{r\in R(y)} \mu_r^y c^{(r)}\right) \tag{10}$$

and

$$v_y = -\|d_y\|^2 - \sum_{j\in J} \lambda_j^y \epsilon_j. \tag{11}$$

We observe, taking into account (10), that if for a given $\theta > 0$ it holds $\epsilon_j \le \theta$, for every $j \in J$, then the occurrence of $d_y = 0$ implies that

$$\sum_{j\in J} \lambda_j^y g^{(j)} + \sum_{r\in R(y)} \mu_r^y c^{(r)} = 0 \tag{12}$$

which will be referred in the sequel as a θ-stationarity condition for point y in the sense defined in [11]. We also note that from (11), taking into account nonnegativity of ϵ_js, it follows that whenever v_y is small in modulus then the norm of d_y is small as well, and, consequently, an approximate θ-stationarity condition is fulfilled.

The latter observations will later help to state the stopping criterion in our algorithm, while the following remark will turn out critical to design the bundle-management rules.

Remark 1. Let $t > 0$ denote the step-size and consider any feasible point $y + td_y$ along the search direction d_y. Observe that from (8) it follows

$$I^{(J)} = I \implies f(y + td_y) - f(y) \le h_y^{(J)}(td_y).$$

In particular the following property holds.

Lemma 1. *If for some $t > 0$ it holds*

$$f(y + td_y) - f(y) > h_y^{(J)}(td_y) \tag{13}$$

then

$$I(y + td_y) \cap I^{(J)} = \emptyset.$$

A ssume by contradiction that there exists an index $i \in I(y + td_y) \cap I^{(J)}$. Taking into account concavity of f_i, from the definition of $h_y^{(J)}(\cdot)$ it follows that

$$f(y + td_y) = f_i(y + td_y) \le f_i(y) + tg^{(i)^\top} d_y \le f_i(y) + h_y^{(J)}(td_y) = f(y) + h_y^{(J)}(td_y),$$

which contradicts the hypothesis.

The above lemma ensures in fact that if the actual function at $y+td_y$ stays above the piecewise-affine function, then no element of the set $I(y+td_y)$ belongs to the set $I^{(J)}$. Such property depends on the concavity of component functions and does not hold for general minmax problems. In particular, satisfaction of condition (13) is the basis of our strategy for bundle enrichment. Observe, finally, that checking (13) requires to design a special kind of line-search, where the actual values of the objective function are compared with those of the piecewise-affine model, unlike standard line-search approaches which involve only a linear model (see [9] where a concave function is used for comparison purpose in the line search).

Before formally introducing our Piecewise-Concave Local Minimization (PCLM) algorithm we emphasize that, in implementing the bundling technique evoked in Remark 1, we adopt a parsimonious approach, trying to keep the bundle as small as possible. In particular, we adopt the set $\mathcal{B}=\{(i_y,g_y,0)\}$ as a standard working bundle, where $g_y\in\partial f_{i_y}(y)$, with $i_y\in I(y)$. Whenever necessary, we enrich the bundle by inserting triplets of the type (i_+,g_+,ϵ_+), defined with respect to feasible points x_+ that are "close" to y. In particular, denoting by i_+ any index in $I(x_+)$, we select g_+ as any subgradient in $\partial f_{i_+}(y)$ and set $\epsilon_+=f(y)-f_{i_+}(y)$.

The formal statement of PCLM is presented in Algorithm 1. PCLM takes as an input any feasible point $x\in X$, and returns an approximate local minimizer $x_*\in X$. The following parameters are to be set: the optimality parameter $\delta>0$, the closeness parameter $\eta>0$ and the step-size reduction parameter $\sigma\in(0,1)$.

Remark 2. The insertion of any index $i_+\in I(x_+)$ in the bundle, as a consequence of Lemma 1, ensures the generation of a useful cut, in the sense that

$$f_{i_+}(y)+g_+^\top(x_+-y)>\ell_y^{(J)}(x_+).$$

Later, in §3, we will focus on finite termination of Algorithm 1 (PCLM). In particular, we will prove that PCLM terminates after finitely many steps at a point satisfying the stopping test at Step 4, showing also that such criterion is a suitable approximate optimality condition. Here, we introduce few explanatory remarks regarding the remaining steps of PCLM. The algorithm generates a sequence of tentative feasible displacements from the current stability center y by means of a *backtracking line-search* along the direction d_y obtained at Step 3. As soon as a sufficient reduction is obtained at Step 10, then a *serious step* can be made and the bundle is entirely renewed, at Step 2, by getting rid of all the past information. If the tentative point does not give sufficient descent, then the closeness test at Step 13 prevents its inclusion in the bundle, by further iterating the line-search at Step 18. As soon as a tentative point, for which the descent test is not satisfied, gets sufficiently close to the stability center, a bundle enrichment, i.e., a *null step*, takes place at

Algorithm 1 PCLM(x)

Input: a feasible starting point $x \in X$, parameters $\delta > 0$, $\eta > 0$, $\sigma \in (0,1)$

Output: an approximate local minimizer $x_* \in X$

1: Set $y = x \in X$ ▷ Initialization
2: Select $i_y \in I(y)$ and $g_y \in \partial f_{i_y}(y)$, set $\mathcal{B} = \{(i_y, g_y, 0)\}$ and $J = \{1\}$ ▷ (Re-)set the bundle
3: Find (d_y, v_y) by solving problem $QP_y^{(J)}$ ▷ Find a descent direction at y
4: **if** $v_y > -\delta$ **then** ▷ Stopping test
5:     set $x_* = y$ and **exit** ▷ Return y as an approximate minimizer
6: **else** ▷ y is not an approximate minimizer
7:     set $t_{\max} = \max\{t > 0 : y + td_y \in X\}$ ▷ Find the maximum step-size along d_y
8:     set $t_0 = \min\{1, t_{\max}\}$ and $k = 0$ ▷ Initialize the line-search
9: **end if**
10: **if** $f(y + t_k d_y) \le f(y) + h_y^{(J)}(t_k d_y)$ **then** ▷ Descent test
11:     set $y = y + t_k d_y$ and **go to 2** ▷ Make a serious step
12: **end if**
13: **if** $t_k \|d_y\| \le \eta$ **then** ▷ Closeness test
14:     set $x_+ = y + t_k d_y$ ▷ Make a null step
15:     select $i_+ \in I(x_+)$, calculate $g_+ \in \partial f_{i_+}(y)$, set $\epsilon_+ = f(y) - f_{i_+}(y)$ ▷ |
16:     set $\mathcal{B} = \mathcal{B} \cup \{(i_+, g_+, \epsilon_+)\}$ and update J and **go to 3** ▷ |
17: **else**
18:     set $t_{k+1} = t_k \sigma$, $k = k+1$, and **go to 10** ▷ Backtrack line-search along d_y
19: **end if**

Steps 14-16. We stress that, unlike standard bundle methods, at Step 2 we completely discard past information, once successful descent is achieved. In this sense the algorithm is parsimonious, as the bundle size is kept as small as possible.

Remark 3. The descent test adopted at Step 10 is substantially different from those usually adopted in line-searches for nonsmooth problems, where most of the times the value of the objective function at the tentative point is tested against a linear function whose slope is a fraction of an estimate of the directional derivative at the stability center. It is easy to verify that the descent test at Step 10 is more restrictive.

3. Termination properties of PCLM

In this section we will denote by $\{y^{(s)}\}_{s\in S}$ the sequence of stability centers generated during the execution of Algorithm 1, and by $(d^{(s)}, v^{(s)})$ the optimal solution of problem $QP^{(J)}_{y^{(s)}}$. Before proving finite termination of Algorithm 1 we state the following Lemma.

LEMMA 2. *Let L be the Lipschitz constant of the objective function $f(\cdot)$ on the feasible region X. Then, the following bound holds:*

$$\|d^{(s)}\| \leq 2L.$$

Proof. First observe that the point $(\bar{d}, \bar{v}) = (0, 0)$ is feasible for $QP^{(J)}_{y^{(s)}}$, since $\epsilon_j \geq 0$, $\forall j \in J$. As a consequence, the optimal solution of $QP^{(J)}_{y^{(s)}}$ is such that

$$v^{(s)} + \frac{1}{2}\|d^{(s)}\|^2 \leq 0.$$

In particular it is

$$g^{\top}_{y^{(s)}} d^{(s)} + \frac{1}{2}\|d^{(s)}\|^2 \leq 0,$$

from which the thesis easily follows. $\square$

We are now ready to present the main result about finite termination of Algorithm 1.

THEOREM 1. *For any feasible starting point $x \in X$ the PCLM algorithm terminates after finitely many steps at a point $y^{(s)} \in X$ satisfying the exit condition at Step 4, i.e., $v^{(s)} > -\delta$.*

Proof. Assume for a contradiction that the algorithm never terminates, hence that infinitely many passages occur through Step 3. Now observe that, due to convexity of the model function and taking into account Lemma 2, whenever the descent test at Step 10 is fulfilled, then the corresponding reduction of the objective function is bounded away from zero, i.e.,

$$f(y^{(s)}) - f(y^{(s)} + t_k d^{(s)}) \geq -h^{(J)}_{y^{(s)}}(t_k d^{(s)}) \geq -t_k v^{(s)} > \frac{\sigma \eta \delta}{2L} > 0.$$

As a consequence, the number of successful line searches at Step 10 cannot be infinite due to the assumptions of compactness of X, and we can assume that a stability-center index $\bar{s}$ exists such that the algorithm loops between Step 3 and Step 16, making an infinite sequence of null steps without ever updating the stability center again. Now observe that, taking into account Lemma 1, every time the new element (i_+, g_+, ϵ_+) is inserted into $\mathcal{B}$ at Step 16, for some $i_+ \in I(x_+)$, it holds that $i_+ \notin I^{(J)}$. Hence, after finitely many insertions of new elements into the bundle, we would have $I^{(J)} = I$. Taking into account (6), the latter implies that

$$f(x_+) = f^{(J)}(x_+) \leq \ell^{(J)}_{y^{(\bar{s})}}(x_+) = f(y^{(\bar{s})}) + h^{(J)}_{y^{(\bar{s})}}(x_+ - y^{(\bar{s})}) \leq f(y^{(\bar{s})}) - \delta \tag{14}$$

which contradicts the fact that a sufficient descent is no longer achievable from $y^{(\bar{s})}$. $\square$

COROLLARY 1. *The PCLM algorithm terminates after finitely many steps at a point $y^{(s)} \in X$ satisfying the approximate θ-stationarity condition*

$$\|\sum_{j\in J}\lambda_j^{(s)}g^{(j)}+\sum_{r\in R(y)}\mu_r^{(s)}c^{(r)}\|<\sqrt{\delta}, \tag{15}$$

where $J\subseteq I_\theta(y^{(s)})$ and $\theta=2L\eta$.

Proof. The stopping condition $v^{(s)}>-\delta$, taking into account (11) and nonnegativity of ϵ_js, implies $\|d^{(s)}\|<\sqrt{\delta}$, that is, from (10),

$$\|\sum_{j\in J}\lambda_j^{(s)}g^{(j)}+\sum_{r\in R(y)}\mu_r^{(s)}c^{(r)}\|<\sqrt{\delta}.$$

Thus all we have to prove is that $J\subseteq I_\theta(y^{(s)})$, i.e., $\epsilon_j\le\theta$ for every $j\in J$. To this aim, we observe that every time the bundle is enriched by a new element (i_+,g_+,ϵ_+), the conditions

$$f_{i_+}(y^{(s)}+t_kd^{(s)})=f(y^{(s)}+t_kd^{(s)})>f(y^{(s)})+h_{y^{(s)}}^{(J)}(t_kd^{(s)})$$

and

$$t_k\|d^{(s)}\|\le\eta$$

hold. On the other hand, concavity of $f_{i_+}(\cdot)$ implies that

$$f_{i_+}(y^{(s)}+t_kd^{(s)})\le f_{i_+}(y^{(s)})+t_kg_+^\top d^{(s)}.$$

Summing up, we have that

$$f(y^{(s)})-f_{i_+}(y^{(s)})\le 2L\eta=\theta$$

from which the thesis easily follows. □

4. Computational experience

In order to assess the performance of the PCLM algorithm, we have prepared two sets of instances, whose objective function is the maximum over finitely many concave quadratic functions, i.e.,

$$f(x)=\max_{i\in I}\left\{\frac{1}{2}x^\top Q_ix+b_i^\top x+c_i\right\} \tag{16}$$

where Q_is are negative definite matrices, b_is are n-dimensional real vectors, and c_is are scalars. The feasible region X is a box in all problems.

The first test-set, next referred to as TS1, contains a group of 7 extra-small scale instances. In particular, TS1 contains three new sample functions in $\mathbb{R}^2$, whose structures are presented in Figures 1, 2, and 3, where x^* denotes the estimate of the global minimizer, and the feasible set is the box defined by l and u, the vectors of lower and upper bounds on the variables, respectively.

The remaining 4 instances are the piecewise-concave reformulation of the box-constrained test-problems available in [19], the problem studied therein being the linearly-constrained maximization of a piecewise-convex function. A summary of the relevant details of TS1 instances is reported in Table 1, where the best known optimal value f^* can also be found. In particular, for problems TS1.1–TS1.3, f^* is the best result obtained adopting the solvers available on the NEOS Server for Optimization [6, 16], while for problems TS1.4–TS1.7, f^* is the value available in [22].

Figure 1 Test problem TS1.1

$$
\begin{aligned}
f_1(x) &= -x_1^2 - x_2^2 + 4x_1\\
f_2(x) &= -x_1^2 - x_2^2 - 4x_1\\
f_3(x) &= -x_1^2 - x_2^2 + 4x_2\\
f_4(x) &= -x_1^2 - x_2^2 - 4x_2\\
f_5(x) &= -x_1^2 - x_2^2 + 12x_1 + 4x_2 - 34\\
f_6(x) &= -x_1^2 - x_2^2 + 12x_1 - 4x_2 - 34\\
f_7(x) &= -x_1^2 - x_2^2 + 8x_1 - 10\\
f_8(x) &= -x_1^2 - x_2^2 + 16x_1 - 58\\
l &= (-1,-1)\\
u &= (7,1)\\
x^* &= (0.0, 0.0)
\end{aligned}
$$

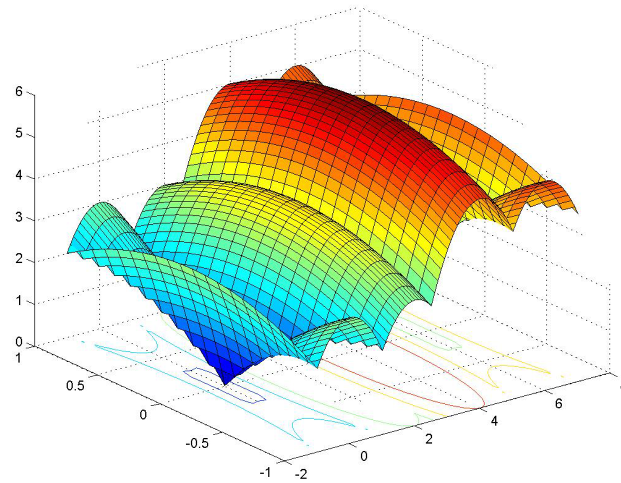
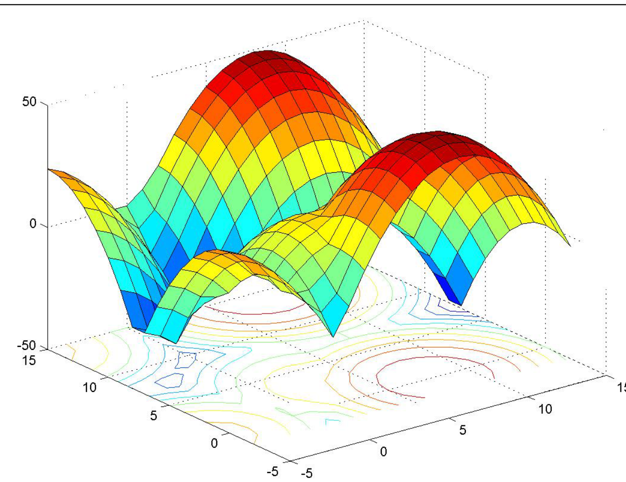


Figure 2 Test problem TS1.2

$$
\begin{aligned}
f_1(x) &= -x_1^2 - x_2^2 + 6x_1 + 6x_2\\
f_2(x) &= -x_1^2 - x_2^2 - 12x_1 - 11\\
f_3(x) &= -x_1^2 - x_2^2 + 16x_1 - 15\\
f_4(x) &= -x_1^2 - x_2^2 + 6x_1 + 36x_2 - 315\\
f_5(x) &= -x_1^2 - x_2^2 - 12x_1 + 30x_2 - 236\\
f_6(x) &= -x_1^2 - x_2^2 + 16x_1 + 30x_2 - 240\\
l &= (-5,-2)\\
u &= (15,15)\\
x^* &= (15.0, 7.5)
\end{aligned}
$$



The second set of instances, next referred to as TS2, is based on the classification proposed in [1, 37], containing four classes of test problems of extra-small (XS, with $n = 20$), small (S, with $n = 50$), medium (M, with $n = 200$), and large (L, with $n = 1000$) size. For each group, we have

Figure 3 **Test problem TS1.3**

$$
\begin{aligned}
f_1(x) &= -9x_1^2 - 25x_2^2 + 225\\
f_2(x) &= -9x_1^2 - 4x_2^2 + 126x_1 - 405\\
f_3(x) &= -9x_1^2 - 4x_2^2 - 180x_1 - 864\\
f_4(x) &= -9x_1^2 - 36x_2^2 + 576x_2 - 1980\\
f_5(x) &= -9x_1^2 - 36x_2^2 - 720x_2 - 3276\\
l &= (-10, -8)\\
u &= (10, 8)\\
x^* &= (-2.78514, -5.29004)
\end{aligned}
$$

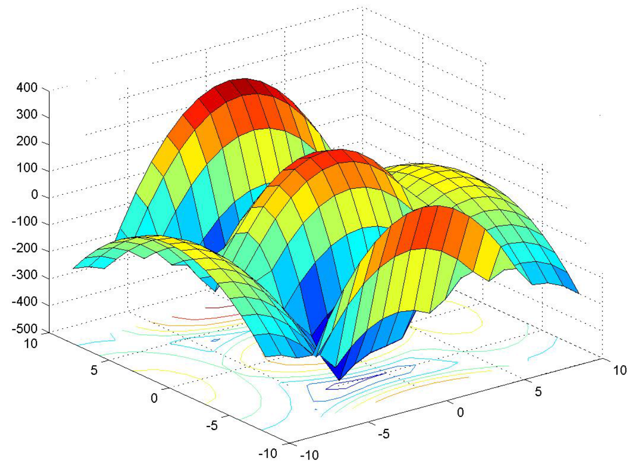


Table 1 **Extra-small test-problems TS1**

Test problem		Size		Best known optimum
ID	Reference	n	m	f^*
1	See Figure 1	2	8	0.000 000
2	See Figure 2	2	6	−56.250 000
3	See Figure 3	2	5	−544.426 392
4	[19, Example 2]	2	5	−1.223 74
5	[19, Example 5]	2	4	−1.223 74
6	[19, Example 8]	3	5	−5.223 69
7	[19, Example 9]	3	5	−17.222 80

adopted $m = 5$ and $m = 10$, and we have considered three different classes of sparsity for randomly generating the negative definite matrix Q (dense, sparse, and diagonal). Algorithm PCLM has been implemented in Java, and the tests were conducted on a 3.50 GHz Intel Core i7 machine. The QP solver of IBM ILOG CPLEX 12.6 [36] has been used to solve the quadratic subprograms. The parameters have been set as follows: $\delta = 0.0001$, $\eta = 0.01$, and $\sigma = 0.7$.

For each test problem of both sets TS1 and TS2, thirty runs of PCLM have been executed, each time adopting a different feasible starting point randomly sampled from a uniform distribution. We have recorded the local minimum and some relevant performance indices returned upon termination of each run, and we present the corresponding statistical results in Tables 2 and 3, where we report the min/avg/max value of

- n_f, the number of function evaluations,
- n_g, the number of gradient evaluations,
- n_f^{it}, the average number of function evaluations per iteration, obtained as the ratio between the number of function evaluations and the number of quadratic problems solved.

Table 2 **PCLM statistical results – TS1**

ID		n_f	n_g	n_f^{it}	n_g^{sc}	*cpu*
	min	17	7	2.3	1.5	0.000
1	avg	45	14	3.2	1.9	0.017
	max	102	26	4.4	2.5	0.094
	min	5	5	1.0	1.1	0.000
2	avg	79	20	3.8	1.6	0.016
	max	145	41	5.8	2.2	0.047
	min	3	3	1.0	1.2	0.000
3	avg	88	34	2.3	1.6	0.020
	max	198	80	3.2	2.6	0.047
	min	3	3	1.0	1.0	0.000
4	avg	64	23	2.5	1.5	0.017
	max	138	42	4.6	2.6	0.031
	min	3	3	1.0	1.0	0.000
5	avg	64	23	2.5	1.5	0.015
	max	138	42	4.6	2.6	0.047
	min	4	4	1.0	1.0	0.000
6	avg	70	26	2.4	1.4	0.017
	max	167	51	3.9	2.0	0.047
	min	4	4	1.0	1.0	0.000
7	avg	66	26	2.3	1.4	0.013
	max	162	59	4.7	2.3	0.031

- n_g^{sc}, the average number of elements in the bundle, obtained as the ratio between the number of gradient evaluations and the number of stability centers touched during the execution.
- *cpu*, the execution time (measured in seconds).

Some considerations can be drawn from the results. We observe first that n_f^{it}, the average number of function evaluations per iteration, is rather small, as it is never larger than 6 on TS1, while in TS2 it gets closer and closer to the ideal value of 1 as the problem size grows. In other words, at almost each iteration the alternative decision about serious or null step is taken without requiring backtrack. Note in fact that n_f and n_g tend to be very similar.

In addition n_g^{sc}, the average number of elements in the bundle, is almost always substantially smaller than m, the number of component functions. This fact represents the improvement our algorithm provides with respect to classic minmax methods (e.g. [31]) based on completely linearized models of the objective function.

To make our computational experience more complete, we have embedded the PCLM algorithm into two global optimization strategies, where PCLM is used as a local-minima generator. In particular, we have first designed a global optimization strategy according to the well known Variable Neighborhood Search (VNS) paradigm, introduced in [41] (see also [32] for many variants, and [7] for convergence properties).

Table 3 **PCLM statistical results – TS2**

ID		n_f	n_g	n_f^{it}	n_g^{sc}	*cpu*	*ID*		n_f	n_g	n_f^{it}	n_g^{sc}	*cpu*
	min	34	34	1.0	1.7	0.000		min	298	298	1.0	1.4	0.234
20.5.de	avg	132	109	1.1	2.5	0.060	200.5.de	avg	1295	1291	1.0	2.5	1.580
	max	533	388	1.6	3.5	0.188		max	4628	4558	1.0	4.1	5.438
	min	31	29	1.0	1.4	0.000		min	380	380	1.0	1.8	0.391
20.5.sp	avg	131	95	1.3	2.0	0.056	200.5.sp	avg	792	792	1.0	2.6	0.978
	max	616	360	1.8	3.0	0.203		max	1702	1700	1.0	3.4	2.203
	min	49	45	1.1	1.1	0.016		min	991	951	1.0	2.3	1.328
20.5.di	avg	179	119	1.4	2.2	0.074	200.5.di	avg	1231	1169	1.1	3.9	1.660
	max	924	544	1.8	2.8	0.297		max	2582	2539	1.1	4.4	3.219
	min	64	55	1.0	1.5	0.031		min	432	432	1.0	1.8	0.578
20.10.de	avg	316	243	1.2	2.9	0.129	200.10.de	avg	3014	2991	1.0	3.0	5.314
	max	1302	879	1.5	4.0	0.484		max	14057	14055	1.0	5.7	21.938
	min	34	33	1.0	1.2	0.000		min	576	576	1.0	2.0	0.922
20.10.sp	avg	198	143	1.3	1.9	0.074	200.10.sp	avg	1640	1640	1.0	3.1	2.989
	max	786	526	1.6	2.8	0.313		max	7040	7040	1.0	4.5	13.344
	min	170	121	1.3	2.5	0.078		min	1671	1644	1.0	4.2	3.500
20.10.di	avg	644	430	1.5	3.2	0.262	200.10.di	avg	2905	2679	1.1	5.6	5.969
	max	1368	816	1.9	4.9	0.484		max	5068	4589	1.2	7.1	10.047
	min	69	69	1.0	1.3	0.016		min	1111	1111	1.0	1.1	11.328
50.5.de	avg	208	196	1.0	2.1	0.117	1000.5.de	avg	1698	1698	1.0	1.5	22.059
	max	770	649	1.2	3.3	0.422		max	3151	3151	1.0	2.6	51.406
	min	52	52	1.0	1.0	0.016		min	1598	1598	1.0	1.5	19.391
50.5.sp	avg	102	101	1.0	1.7	0.055	1000.5.sp	avg	2407	2407	1.0	2.1	36.161
	max	221	209	1.1	2.8	0.156		max	3908	3908	1.0	2.9	65.828
	min	168	143	1.0	2.4	0.063		min	5138	5078	1.0	4.5	85.984
50.5.di	avg	404	322	1.2	3.1	0.205	1000.5.di	avg	5852	5775	1.0	4.7	99.390
	max	1100	687	1.6	3.7	0.438		max	7574	7459	1.0	4.9	127.359
	min	202	202	1.0	2.1	0.078		min	1424	1424	1.0	1.4	28.891
50.10.de	avg	752	663	1.1	3.2	0.386	1000.10.de	avg	3505	3505	1.0	2.8	102.338
	max	2594	2427	1.4	4.1	1.281		max	7891	7891	1.0	5.3	262.672
	min	96	96	1.0	1.7	0.063		min	2070	2070	1.0	1.9	49.500
50.10.sp	avg	515	464	1.1	2.9	0.268	1000.10.sp	avg	4936	4936	1.0	3.4	155.057
	max	1189	1094	1.3	4.4	0.656		max	20507	20507	1.0	5.0	696.625
	min	310	280	1.0	2.0	0.141		min	9926	9859	1.0	8.3	336.938
50.10.di	avg	1210	960	1.3	4.2	0.640	1000.10.di	avg	13144	12961	1.0	8.8	441.005
	max	6123	5936	1.6	5.6	3.266		max	20030	19643	1.0	9.2	672.891

Our VNS implementation works as follows. Assume a local minimum x_*^k, that is an estimate of the global minimizer, be available at the kth iteration of our Piecewise-Concave Variable Neighborhood Search (PCVNS) algorithm, then the aim of PCVNS is to improve x_*^k by generating further tentative points obtained via repeated execution of PCLM. In order to generate new local minimizers we define a set of $\ell_{\max}$ nested box-shaped neighborhoods of x_*^k

$$\mathcal{N}_1(x_*^k) \subset \mathcal{N}_2(x_*^k) \subset \ldots \mathcal{N}_\ell(x_*^k) \subset \ldots \subset \mathcal{N}_{\ell_{\max}}(x_*^k),$$

then we successively run PCLM by randomly selecting the starting point x_ℓ^k inside each neighborhood $\mathcal{N}_\ell(x_*^k)$, beginning from the innermost one, until a better local minimum is found, or no improvement is achieved, even selecting the outermost neighborhood $\mathcal{N}_{\ell_{\max}}(x_*^k)$. We remark that no feasible point should be prevented to become a starting point for PCLM during the iterations of PCVNS, hence we select $\mathcal{N}_{\ell_{\max}}(x_*^k) = X$. For each x_ℓ^k algorithm PCLM will return a new local minimum x_*, that will become the new current estimate x_*^{k+1} only in case sufficient improvement of the objective function value is achieved.

The formal schema of PCVNS is presented in Algorithm 2. PCVNS takes as an input any feasible point $\bar{x} \in X$, the optimality parameter $\delta_G > 0$, the maximum number of neighborhoods $\ell_{\max}$, and returns the best among all the approximate local minimizers generated via PCLM.

Algorithm 2 PCVNS

Input: a feasible starting point $\bar{x} \in X$; parameter $\delta_G > 0$

Output: $x^* \in X$, the best of all local minimizers generated by PCLM

1: Set $x^0 = \bar{x} \in X$ ▷ Initialization
2: Set $x_*^0 = \text{PCLM}(x^0)$ and $k = 0$ ▷ Get the first local minimizer
3: Set $\mathcal{N}_0(x_*^k) = \emptyset$ and $\ell = 1$
4: **if** $\ell > \ell_{\max}$ **then** ▷ Stopping test
5:     set $x^* = x_*^k$ and **exit** ▷ Return x^* as the best local minimizer
6: **else**
7:     generate $\mathcal{N}_\ell(x_*^k)$ ▷ Generate a larger neighborhood of x_*^k
8:     randomly select $x_\ell^k \in \mathcal{N}_\ell(x_*^k) \setminus \mathcal{N}_{\ell-1}(x_*^k)$ ▷ Select a new starting point
9:     set $x_* = \text{PCLM}(x_\ell^k)$ ▷ Get a new local minimizer
10: **end if**
11: **if** $f(x_*) < f(x_*^k) - \delta_G |f(x_*^k)|$ **then** ▷ Sufficient descent
12:     set $x_*^{k+1} = x_*$, $k = k+1$ and **go to 3** ▷ Current solution update
13: **else**
14:     set $\ell = \ell + 1$ and **go to 4**
15: **end if**

Next, we have designed a new global search heuristics that is based on the coordinate search paradigm. Given a local minimum x_*^k as an estimate of the global minimizer, the aim of our Piecewise-Concave Coordinate Direction Search (PCCDS) algorithm is to improve x_*^k by generating further tentative points obtained, also in this case, via repeated execution of PCLM. In particular, denoting by $\mathcal{E} \triangleq \{e_1, -e_1, \ldots, e_n, -e_n\}$ the set of all the signed coordinate directions, we randomly select a subset $E^{(k)} \subseteq \mathcal{E}$, and consider each (signed) coordinate direction $(\pm)e_i \in E^{(k)}$ at x_*^k, as a candidate to provide a set of starting points for PCLM, according to a kind of backtrack along it. In fact, once a direction is selected, and the maximal feasible step-size is calculated, the algorithm backtracks along such a direction to generate starting points for PCLM, until either PCLM returns a better local minimizer than x_*^k, or the direction can be considered as exhausted (i.e., worth no further exploration). The idea of backtracking along coordinate directions is motivated by the attempt to select starting points which are far from x_*^k. Any selection of a step-size gives rise to a starting point for PCLM whose outcome is a new local minimum x_*. Such a point will become the new current estimate x_*^{k+1}, only in case sufficient improvement of the objective function value is achieved. The selected direction is considered exhausted, and the related backtrack is truncated, as soon as PCLM gets to work in the vicinity of the local minimizer x_*^k. Hence, PCCDS terminates at x_*^k as soon as all the directions belonging to $E^{(k)}$ have been explored at x_*^k.

The formal statement of PCCDS is presented in Algorithm 3. PCCDS takes as an input any feasible point $\bar{x} \in X$, and returns the best among all the approximate local minimizers generated via PCLM. The following parameters are to be set: the optimality parameter $\delta_G > 0$, the closeness parameter $\eta_G > 0$, the step-size reduction parameter $\sigma_G \in (0, 1)$.

Some explanatory comments about the relevant steps of Algorithm 3 (PCCDS) are now in order. At Step 3 the algorithm checks whether there are further coordinate directions to explore, aiming to improve the current solution x_*^k. If this is the case, one of such directions $e \in E^{(k)}$ is selected, and a line-search along e is started which first moves as far as possible from x_*^k. The points generated along the search direction are used as starting points for the PCLM algorithm at Step 13, and as soon as the sufficient descent test is satisfied at Step 15, the backtrack is truncated, the current solution is updated, and the set of coordinate directions is newly refilled. Otherwise, if the sufficient descent is never obtained, the backtrack is truncated as soon as either the new starting point of PCLM (see Step 9), or the local minimizer returned by PCLM (see Step 19), gets very close to x_*^k.

Both PCVNS and PCCDS methods have been implemented in Java, and the tests were conducted on the same 3.50 GHz Intel Core i7 machine. The parameters have been set as follows: $\ell_{\max} = 6$, $\delta_G = 0.0001$, $\eta_G = 0.1$, and $\sigma_G = 0.5$.

The experimental plan consists in launching both global algorithms adopting a multi-start procedure by uniform random sampling of 30 feasible starting points. The corresponding statistical results are summarized in Tables 4 (TS1 test problems) and 5-6 (TS2 test problems), where we report the min/avg/max value of

- N_f: the number of function evaluations,
- N_g: the number of gradient evaluations,

Algorithm 3 PCCDS

Input: a feasible starting point $\bar{x} \in X$; parameters $\delta_G > 0$, $\eta_G > 0$, and $\sigma_G \in (0,1)$

Output: $x^* \in X$, the best of all local minimizers generated by PCLM

1: Set $x^0 = \bar{x} \in X$ and $E^{(0)} \subseteq \mathcal{E}$ ▷ Initialization
2: Set $x_*^0 = \mathrm{PCLM}(x^0)$ and $k = 0$ ▷ Get the first local minimizer
3: **if** $E^{(k)} = \emptyset$ **then** ▷ Stopping test
4:     set $x^* = x_*^k$ and **exit** ▷ Return x^* as the best local minimizer
5: **else**
6:     select $e \in E^{(k)}$ and set $E^{(k)} = E^{(k)} \setminus \{e\}$ ▷ Select a search direction
7:     set $r_0 = \max\{r > 0 : x_*^k + re \in X\}$ and $j = 0$ ▷ Initialize line-search
8: **end if**
9: **if** $r_j \leq \eta_G$ **then** ▷ The search direction e is exhausted
10:     **go to 3**
11: **else**
12:     set $x = x_*^k + r_j e$ ▷ Generate a new starting point along e
13:     set $x_* = \mathrm{PCLM}(x)$ ▷ Get a new local minimizer
14: **end if**
15: **if** $f(x_*) < f(x_*^k) - \delta_G |f(x_*^k)|$ **then** ▷ Sufficient descent
16:     set $x_*^{k+1} = x_*$ and $E^{(k+1)} \subseteq \mathcal{E}$ ▷ Current solution update
17:     set $k = k + 1$ and **go to 3**
18: **end if**
19: **if** $\|x_* - x_*^k\| \leq \epsilon_G$ **then** ▷ The search direction e is exhausted
20:     **go to 3**
21: **else**
22:     set $r_{j+1} = \sigma_G r_j$, set $j = j + 1$ and **go to 9** ▷ Step-size reduction
23: **end if**

- N_L: the number of PCLM executions,
- N^*: the number of global minimizer updates,
- *cpu*: the execution time (measured in seconds).

Moreover, in Table 4 we report the min/avg/max value of f^*, the objective function value upon termination, while in Tables 5-6 we report the rate of success w^*, that is the percentage number of times that the algorithm stops at a point whose objective value is within an error of 2% of the global minimum, where we assume the global minimum to be the lowest objective function value returned by either PCVNS or PCCDS over the 60 total runs.

To comment on the results, we observe first that, as far as TS1 problems are concerned, the global minimum is always correctly located by both PCVNS and PCCDS. On the other hand some failures occur in dealing with TS2 problems. In particular, on XS and S problems the rate of success w^* is less than 100% in 4 and 2 cases for PCVNS and PCCDS, respectively. The same phenomenon occurs for M and L problems 2 times in PCVNS, while it is absent in PCCDS. We do not report statistical results of PCVNS on problems 1000.5.di and 1000.10.di since the time limit of 80 hours to complete the 30 runs expired when a significant amount of results were not yet available.

Focusing first on TS1, PCCDS seems to work reasonably better than PCVNS as far N_L, N^* and *cpu* are concerned, and remarkably better with respect to N_f and N_g. This behaviour is substantially confirmed on TS2 problems, where sometimes N_f and N_g differ of an order of magnitude. A possible explanation is in the opposite philosophies adopted by the two approaches in trying to escape from a local minimum. In fact, PCCDS looks initially for a starting point to be as far as possible from the current estimate of the global minimum, while PCVNS starts its search from a point selected in a "small" neighborhood. Thus, in the latter case, it is likely to get more failures in attempting to improve the current estimate of the global minimum. This fact results in an increase of both function and gradient evaluations.

5. Conclusions

We have introduced the PCLM method for the numerical minimization of a piecewise-concave function subject to linear constraints. Such method explicitly exploits concavity of the component functions in order to cope with nonconvexity of the objective function. Furthermore, an approximate line-search strategy, based on a curvilinear model, allows to explore a search direction in order to obtain a descent or an appropriate local improvement of the model. Termination at a point satisfying an approximate stationarity condition has been proved. A thorough computational analysis of the method has been carried out on problems with as many as 1000 variables. We have finally proposed and numerically tested two global search methods embedding PCLM, whose aim is to improve the current estimate of the global minimizer by exploring different parts of the feasible region.

References

[1] A. Bagirov and N. Karmitsa and M. M. Mäkelä, *Introduction to Nonsmooth Optimization: Theory, Practice and Software*, Springer, 2014.

Table 4 Statistical results on TS1

		PCVNS						PCCDS					
ID		f^*	N_f	N_g	N_L	N^*	*cpu*	f^*	N_f	N_g	N_L	N^*	*cpu*
	min	0.000 000	1367	395	23	1	0.156	0.000 000	198	64	7	2	0.016
1	avg	0.000 001	1683	486	29	2	0.344	0.000 000	286	88	11	3	0.056
	max	0.000 014	2192	618	37	3	0.516	0.000 000	363	105	16	4	0.172
	min	−56.250 000	1373	363	23	1	0.172	−56.250 000	291	76	8	1	0.031
2	avg	−56.250 000	2280	572	31	2	0.347	−56.250 000	387	98	9	2	0.066
	max	−56.250 000	3533	859	43	4	0.844	−56.250 000	559	140	12	3	0.172
	min	−544.426 392	1898	672	23	1	0.250	−544.426 392	663	276	11	1	0.094
3	avg	−544.426 391	2693	1002	29	2	0.434	−544.426 391	1066	448	16	2	0.209
	max	−544.426 366	3652	1353	33	3	0.672	−544.426 366	1460	612	22	4	0.328
	min	−1.224 060	1570	494	23	1	0.234	−1.224 060	404	132	8	1	0.063
4	avg	−1.224 043	1688	555	27	2	0.329	−1.224 043	507	168	9	2	0.110
	max	−1.224 025	1875	636	34	4	0.563	−1.224 028	652	208	11	3	0.156
	min	−1.224 060	1570	494	23	1	0.219	−1.224 060	404	132	8	1	0.047
5	avg	−1.224 043	1688	555	27	2	0.342	−1.224 043	507	168	9	2	0.103
	max	−1.224 025	1875	636	34	4	0.703	−1.224 028	652	208	11	3	0.172
	min	−5.224 060	1636	621	23	1	0.250	−5.224 060	489	153	11	1	0.047
6	avg	−5.224 046	1891	698	30	2	0.415	−5.224 047	617	199	12	2	0.119
	max	−5.224 027	2163	791	38	4	0.672	−5.224 030	830	271	17	3	0.172
	min	−17.628 004	1858	635	23	1	0.328	−17.628 004	368	144	9	1	0.063
7	avg	−17.627 975	2154	772	32	2	0.448	−17.451 592	558	197	10	1	0.118
	max	−17.627 950	2464	883	45	4	0.578	−17.250 000	950	302	15	2	0.219

[2] A. Bagirov and A.N. Ganjehlou, *A quasisecant method for minimizing nonsmooth functions*, Optimization methods and software, 25(1), 3–18 2010.

[3] J.W. Bandler and W. Kellerman and K. Madsen, *A Superlinearly Convergent Minimax Algorithm for Microwave Circuit Design*, IEEE Transactions on Microwave Theory and Techniques, 33, 21, 1519–1530, 1985.

[4] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, 2009.

[5] D. Bertsimas, D. B. Brown, and C. Caramanis, *Theory and applications of robust optimization*, SIAM Review, 53, pp. 464–501.

[6] J. Czyzyk and M. P. Mesnier and J. J. Moré , *The NEOS Server*, IEEE Journal on Computational Science and Engineering 5 (1998), pp. 68–75.

[7] J. Brimberg and P. Hansen and N. Mladenovich , *Attraction probabilities in variable neighborhood search*, 4OR-Q J Oper Res, 8, (2010), pp. 181-194.

[8] J. Danskin, *The Theory of Max Min*, Springer, Berlin, 1967.

[9] R. De Leone, M. Gaudioso, and L. Grippo, *Stopping criteria for linesearch methods without derivatives*, Mathematical Programming, 30 (1984), pp. 285–300.

Table 5 Statistical results on TS2, XS and S problems

		PCVNS						PCCDS					
ID		N_f	N_g	N_L	N^*	*cpu*	w^*	N_f	N_g	N_L	N^*	*cpu*	w^*
	min	1565	1336	23	1	0.484		1025	580	24	1	0.203	
20.5.de	avg	2612	2259	34	3	1.011	100	1467	911	25	2	0.393	100
	max	5085	4406	63	8	2.078		3023	2140	28	3	0.938	
	min	2482	1974	23	1	0.734		2024	1155	25	1	0.313	
20.5.sp	avg	2938	2288	28	2	1.041	100	2464	1456	28	2	0.553	100
	max	4208	3032	46	4	1.469		3860	2350	36	5	0.953	
	min	2953	2083	23	1	0.953		2944	1645	31	1	0.641	
20.5.di	avg	5260	3853	47	4	1.877	100	4269	2475	35	3	1.143	100
	max	10036	7225	71	9	3.438		5939	3448	40	5	1.531	
	min	4644	3592	32	4	1.781		3684	2444	36	3	0.891	
20.10.de	avg	8937	7170	67	7	3.606	90	11113	7588	60	8	3.380	100
	max	14060	10505	93	13	5.469		17470	12028	85	12	5.297	
	min	4837	3315	23	1	1.500		5888	3243	24	1	0.953	
20.10.sp	avg	5001	3425	23	1	1.751	100	6052	3353	24	1	1.219	100
	max	5589	3808	23	1	2.234		6640	3736	24	1	1.641	
	min	8821	5981	23	1	3.625		11588	7393	32	1	3.781	
20.10.di	avg	12788	8664	38	3	5.441	100	18634	12697	40	3	6.460	100
	max	19231	13186	65	9	8.000		51610	44644	60	5	20.656	
	min	2880	2835	23	1	1.188		617	491	17	1	0.250	
50.5.de	avg	3903	3842	33	2	1.745	100	794	644	21	2	0.360	100
	max	6423	5980	44	5	2.891		1664	1232	28	3	0.703	
	min	1829	1829	25	2	0.688		340	268	22	1	0.109	
50.5.sp	avg	2683	2664	41	4	1.067	100	967	711	46	3	0.355	100
	max	3730	3719	59	7	1.563		1680	1202	90	5	0.906	
	min	7358	6162	23	1	3.438		8582	6057	21	2	3.406	
50.5.di	avg	14218	12334	54	6	7.268	100	12679	9291	30	3	5.173	100
	max	24663	21706	99	15	12.984		18792	13647	42	5	7.750	
	min	10198	9309	38	3	5.078		1716	1332	20	2	0.672	
50.10.de	avg	20251	18558	68	9	10.017	57	7263	5802	43	6	3.088	70
	max	35394	32454	133	23	17.656		14545	11381	71	11	6.031	
	min	5673	5105	23	1	2.688		2600	1640	20	1	0.859	
50.10.sp	avg	11148	10256	55	7	5.315	90	10486	7172	64	5	3.516	73
	max	24989	23422	105	23	13.000		17740	11647	124	9	5.938	
	min	19810	16524	37	3	10.484		19577	14279	24	2	8.891	
50.10.di	avg	34000	28714	71	9	18.446	93	38863	30553	40	4	18.957	100
	max	56088	47523	139	22	30.547		72900	63023	71	8	38.250	

[10] A. V. Demyanov, A. Fuduli, and G. Miglionico, *A bundle modification strategy for convex minimization*, European Journal of Operational Research, 180, (2007), pp. 38–47.

[11] V. F. Demyanov and V. N. Malozemov, *Introduction to Minimax*, Dover Publications Inc., New York, 1974, (English Translation).

Table 6 Statistical results on TS2, M and L problems

		PCVNS						PCCDS					
ID		N_f	N_g	N_L	N^*	*cpu*	w^*	N_f	N_g	N_L	N^*	*cpu*	w^*
	min	7827	7820	23	1	8.422		1595	1587	21	1	4.156	
200.5.de	avg	13100	13061	33	2	14.480	100	2664	2645	24	2	5.765	100
	max	52611	52222	83	14	66.406		5914	5836	28	3	9.391	
	min	8891	8891	23	1	9.266		1085	1085	19	1	3.750	
200.5.sp	avg	9927	9925	24	1	10.816	53	1533	1533	23	2	4.660	100
	max	20729	20694	46	3	23.672		3028	3026	28	2	6.219	
	min	57210	55437	55	7	77.313		29331	26611	22	2	35.672	
200.5.di	avg	84211	81728	83	14	113.147	100	50837	46141	44	3	62.157	100
	max	189415	183949	191	34	255.125		102446	91378	89	6	126.172	
	min	19613	19478	23	1	34.391		3794	3504	19	1	9.031	
200.10.de	avg	29266	29067	26	2	51.635	57	6440	6121	24	2	13.959	100
	max	54032	53606	46	6	100.375		17241	16950	29	3	29.750	
	min	11240	11240	23	1	18.281		1596	1596	17	1	5.688	
200.10.sp	avg	13715	13715	29	1	22.303	100	2671	2671	23	1	7.817	100
	max	20734	20734	36	2	35.438		8048	8048	26	2	18.344	
	min	57210	55437	55	7	77.313		53810	49026	23	2	103.766	
200.10.di	avg	84211	81728	83	14	113.147	100	92963	85493	42	3	177.269	100
	max	189415	183949	191	34	255.125		147573	139310	69	5	273.703	
	min	25488	25488	23	1	238.469		2154	2154	17	1	411.969	
1000.5.de	avg	29714	29714	27	1	283.531	100	2743	2743	23	1	447.169	100
	max	43124	43124	38	2	413.547		4191	4191	30	2	499.281	
	min	28118	28118	23	1	292.031		2876	2876	17	1	424.406	
1000.5.sp	avg	30518	30518	23	1	339.395	100	3686	3686	23	1	463.044	100
	max	35747	35747	23	1	444.641		5176	5176	30	2	518.234	
	min	-	-	-	-	-		60569	58587	24	2	1369.469	
1000.5.di	avg	-	-	-	-	-	-	100767	96499	39	3	2033.929	100
	max	-	-	-	-	-		147949	141031	58	5	2975.625	
	min	27085	27085	23	1	467.594		3427	3171	17	1	457.578	
1000.10.de	avg	35896	35896	29	1	653.716	100	5510	5254	24	1	530.530	100
	max	52001	52001	38	2	1025.781		9901	9645	30	2	688.922	
	min	33565	33565	23	1	657.375		6130	5988	20	1	548.359	
1000.10.sp	avg	51167	51167	32	2	1073.554	100	9073	8931	24	2	648.642	100
	max	81533	81533	40	2	1889.359		24607	24465	32	2	1153.453	
	min	-	-	-	-	-		178187	172723	25	2	5754.859	
1000.10.di	avg	-	-	-	-	-	-	280762	271623	48	3	9030.660	100
	max	-	-	-	-	-		569082	550040	119	6	17983.875	

[12] W. de Oliveira and C. Sagastizábal, *Level bundle methods for oracles with on-demand accuracy*, Optimization Methods and Software, 29 (2014), pp. 1180–1209.

[13] W. de Oliveira and C. Sagastizábal and C. Lemaréchal, *Convex proximal bundle methods in depth: a unified analysis for inexact oracles*, Mathematical Programming, 148, 1-2 (2014), pp. 241-277.

[14] W. de Oliveira and M. Solodov A doubly stabilized bundle method for nonsmooth convex optimization. Tech. rep. (2013). Available on http://www.optimization-online.org/

[15] G. Di Pillo and L. Grippo and S. Lucidi, *A smooth method for the finite minimax problem*, Mathematical Programming, 60 (1993), pp. 187–214.

[16] E. Dolan, *The NEOS Server 4.0 Administrative Guide*, Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory (2001).

[17] D. Du and F. Hwang, *A proof of the Gilbert-Pollak conjecture on the Steiner ratio*, Algorithmica, 7 (1992), pp. 121–135.

[18] D. Du and P. M. Pardalos, *Minimax and Applications*, Kluwer Academic Publishers, (1995).

[19] D. Fortin and I. Tseveendorj, *Piecewise-convex maximization problems: algorithms and computational experiments*, Journal of Global Optimization, 42 (2002), pp. 61–77.

[20] D. Fortin and I. Tseveendorj, *Global optimization and multi knapsack: a percolation algorithm*, European Journal of Operational Research, 154 (2004), pp. 46–56.

[21] D. Fortin and I. Tseveendorj, *Piecewise convex maximization approach to multiknapsack*, Optimization, 58 (2009), pp. 883–895.

[22] D. Fortin and I. Tseveendorj, *Piecewise convex maximization problems: piece adding technique*, Journal of Optimization Theory and Applications, 148 (2011), pp. 471–487.

[23] A. Frangioni, *Generalized bundle methods*, SIAM Journal on Optimization, 13 (2002), pp. 117–156.

[24] A. Fuduli, M. Gaudioso, and G. Giallombardo, *Minimizing nonconvex nonsmooth functions via cutting planes and proximity control*, SIAM Journal on Optimization, 14 (2004), pp. 743–756.

[25] A. Fuduli, M. Gaudioso, and G. Giallombardo, *A DC piecewise affine model and a bundling technique for nonconvex nonsmooth optimization*, Optimziation methods and software, 19(1), 89–102 (2004).

[26] A. Fuduli, M. Gaudioso, G. Giallombardo, and G. Miglionico, *A partially inexact bundle method for convex semi-infinite minmax problems*, Communications in Nonlinear Science and Numerical Simulation, 21 (2015), pp. 172–180.

[27] M. Gaudioso, G. Giallombardo, and G. Miglionico, *An incremental method for solving convex finite min-max problems*, Mathematics of Operations Research, 31 (2006), pp. 173–187.

[28] M. Gaudioso, G. Giallombardo, and G. Miglionico, *On solving the Lagrangian dual of integer programs via an incremental approach*, Computational Optimization and Applications, 44 (2009), pp. 117–138.

[29] M. Gaudioso, E. Gorgone, and M. F. Monaco, *Piecewise-linear approximation in nonconvex nonsmooth optimization*, Numerische Mathematik, 113 (2009), pp. 73–88.

[30] P. Georgiev, P. Pardalos, and A. Chinchuluun, *Localization of minimax points*, Journal of Global Optimization, 40 (2008), pp. 489–494.

[31] J. Hald and K. Madsen, *Combined LP and Quasi–Newton methods for minimax optimization*, Mathematical Programming, 20 (1981), pp. 49–62.

[32] P. Hansen and N. Mladenovic, *Variable neighborhood search: Principles and applications*, European Journal of Operational Research, 130 (2001), pp. 449–467.

[33] W. Hare and C. Sagastizabal, *Computing proximal points of nonconvex functions*, Mathematical Programming, 116 (2009), pp. 221–258.

[34] W. Hare and C. Sagastizabal, *A redistributed proximal bundle method for nonconvex optimization*, SIAM Journal on Optimization, 20 (2010), pp. 2442–2473.

[35] J. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms Vol. I-II*, Springer-Verlag, 1993.

[36] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[37] N. Karmitsa and A. Bagirov and M. M. Mäkelä *Comparing different nonsmooth minimization methods and software*, Optimization Methods and Software, 27 (2012), pp. 131–153.

[38] K. C. Kiwiel, *Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities*, Mathematical Programming, 69 (1995), pp. 89–109.

[39] K. C. Kiwiel, *A proximal bundle method with approximate subgradient linearizations*, SIAM Journal on Optimization, 16 (2006), pp. 1007–1023.

[40] R. Mifflin, *A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization*, in Nondifferential and Variational techniques in Optimization, D.C. Sorensen and J.B. Wets, eds., Mathematical Programming Study, 17 (1982), pp. 77–90.

[41] N. Mladenovich and P. Hansen, *Variable Neighborhood Search*, Computers and Operations Research, 24, 1, (1997), pp. 1097-1100.

[42] A. Nedic and D. Bertsekas, *The effect of deterministic noise in subgradient methods*, Mathematical Programming, 125 (2010), pp. 75–99.

[43] A. Nedic and D. P. Bertsekas, *Incremental subgradient methods for nondifferentiable optimization*, SIAM Journal on Optimization, 12 (2001), pp. 109–138.

[44] M.R. Osborne and G.A. Watson, *An algorithm for minimax optimization in the nonlinear case*, The Computer Journal, 12 (1969), pp. 63-68.

[45] E. Polak, *On the mathematical foundations of nondifferentiable optimization in engineering design*, SIAM Review, 29 (1987), pp. 21–89.

[46] H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM Journal on Optimization, 2 (1992), pp. 121–152.

[47] A. Takeda, S. Taguchi, and R. Tütüncü, *Adjustable robust optimization models for a nonlinear two-period system*, Journal of Optimization Theory and Applications, 136 (2008), pp. 275–295.

[48] F. Tardella, *Piecewise concavity and discrete approaches to continuous minimax problems*, in Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems, P. Pardalos, ed., Kluwer Academic Publisher, 2000, pp. 511–524.

[49] P. Wolfe, *A method of conjugate subgradients for minimizing nondifferentiable functions*, in Nondifferentiable Optimization, M. Balinski and P. Wolfe, eds., vol. 3 of Mathematical Programming Study, North-Holland, Amsterdam, 1975, pp. 145–173.

[50] W. Zangwill, *The piecewise concave function*, Management Science, 13(11) (1967), pp. 900–912.