Journal of Global Optimization **42:** 61–77, 2002.

Piecewise-Convex Maximization Problems: Algorithm and Computational Experiments**

DOMINIQUE FORTIN and IDER TSEVENDORJ
[1]*LIM, Centre de Mathématiques et Informatique,*
39 rue Joliot-Curie - F-13453 Marseille Cedex 13, France.
E-mail: fortin@lim.univ-mrs.fr [2]*INRIA, Domaine de Voluceau, Rocquencourt,*
B.P. 105, 78153 Le Chesnay Cedex, France.
E-mail: Ider.Tsevendorj@inria.fr

Abstract. A function $F : R^n \to R$ is called a piecewise convex function if it can be decomposed into $F(x) = \min\{f_j(x) \mid j \in M\}$, where $f_j : R^n \to R$ is convex for all $j \in M = \{1, 2..., m\}$. In this article, we provide an algorithm for solving $\max F(x)$ subject to $x \in D$, which is based on global optimality conditions. We report first computational experiments on small examples and open up some issues to improve the checking of optimality conditions.

AMS subject classifications: 90C46, 90C47

Key words: Optimality conditions, Global search algorithm, Local search algorithm, Nonconvex and nonsmooth problem, Piecewise convex function

Abbreviations: PCMP – Piecewise Convex Maximization Problems

1. Introduction

Let D be a nonempty, compact and convex subset of $\mathbb{R}^n$ and let M be a finite index set. In order to be self contained, we recall some basic definitions and results from [10].

Definition 1. A function $F : R^n \to R$ is called a piecewise convex function if it can be decomposed into :

$$F(x) = \min\{f_j(x) \mid j \in M\}, \tag{1.1}$$

where $f_j : R^n \to R$ is convex for all $j \in M = \{1, 2..., m\}$.

Definition 2. A problem

$$\begin{cases} \text{maximize } \ F(x), \\ \text{subject to } x \in D \end{cases} \tag{PCMP}$$

is called a piecewise convex maximization problem, if $F(\cdot)$ is a piecewise convex function.

** This research was supported by Direction des transports terrestres, Ministère de l'Equipement des Transports et du Logement (projet N 226 75 01 63-41 50)

We will use further notations, $clco(D)$ as the closure of the convex hull of set D and :

$$
\begin{aligned}
I(z) &= \{i \in M \mid f_i(z) = F(z)\}, \\
D_k(z) &= D \cap \{x \mid f_j(x) > F(z) \text{ for all } j \in M \setminus \{k\}\}
\end{aligned}
$$

for respectively, set of active functions at z, and a special subdomain.

Proposition 1. [10] *If $z \in D$ is a global maximum of (PCMP) then for all $k \in I(z)$*

$$\partial f_k(y) \bigcap N(D_k(z), y) \neq \emptyset \quad \textit{for all} \quad y \textit{ s.t. } f_k(y) = F(z). \qquad \text{(gN)}$$

Definition 3. $F(\cdot)$ is said to be **regular** at $z \in D$ if there exists $k \in I(z)$ and $v \in \mathbb{R}^n$ such that $f_k(v) < f_k(z)$.

Theorem 1. [10] *Let $z \in D$ and $F(\cdot)$ be regular at z. Then a sufficient condition for z to be a global maximum for (PCMP) is:*

$$\partial f_k(y) \bigcap N(clco(D_k(z)), y) \neq \emptyset \textit{ for all } y \textit{ s.t. } f_k(y) = F(z) \qquad \text{(gS)}$$

Piecewise-Convex Maximization Problems have many theoretical [2] and practical applications [7], but algorithms and a solution for such problems do not seem to have been extensively studied yet. The well known [5, 6] convex maximization is a special case of (PCMP). The former is a NP-hard problem, therefore the latter is NP-hard as well.

The purpose of this paper is 2-fold:

- to construct an algorithm for finding the global solution to the piecewise convex maximization problem (PCMP) based on the above mentioned necessary and sufficient optimality conditions,
- to fully describe a practical algorithm to handle this problem and to present initial computational experiments.

The present paper is organized as follows. First in Section 2, we suggest an algorithm for finding the global solution to (PCMP). Then Section 3 is devoted to the very important subproblem of local search, where we provide an algorithm and prove its convergence. Section 4 presents some details of a realization of the algorithm described in Section 2. First experiments, reported in Section 5, on small examples in two and three dimensions, turn out to be effective; however, we give some hints on how to handle structural properties of the intersection of level sets to carry the algorithm over to a higher dimensional space, in an efficient way.

[**PCMP**(x)]
Let z **be a local solution to (PCMP) with starting point** x.
Construct $I(z)$ **and choose** $s \in I(z)$.
Approximate $D_s(z)$ **by polytope** $\Phi = \{x \in \mathbb{R}^n \mid Px \leq p\}$, **where** $D_s(z) \subset \Phi \subset D$.
Approximate $\cap_{m \in M}$ **clco**$(D_m(z))$ **by Lebesgue sets intersection graph.**
/* Approximate $\cap_{m \in M}$ **clco**$(D_m(z))$ **from outside */**
foreach constraint
  $y =$ **tangent point on level set** $f_s(y) = F(z)$ **along constraint normal ;**
  $u = \arg\max\{\langle \nabla f_s(y), x\rangle / x \in \Phi\}$**; /* linearized problem */**
  if $\langle \nabla f_s(y), u - y\rangle > 0$ **and** $u \in D_s(z)$
    then $x := u$**; goto 1; /* better point */**
    else if $u \notin D_s(z)$
    then $\Phi := \Phi \bigcap \{x \mid \langle d, x\rangle \leq n, \}$ **/* add cutting plane */**
endfor

2. Global Maximum: a (PCMP) algorithm

A preliminary theoretical algorithm based on the necessary and sufficient condition (gN), (gS) was outlined in [10]. The purpose of this section is to describe a practical algorithm.

We consider, in the following, only quadratic functions to provide analytic solutions for some simple subproblems and to use standard solvers in experiments.

$$f_i(x) = \frac{1}{2}\langle Q_i x, x\rangle + \langle l_i, x\rangle + \gamma_i$$
$$Q_i = [Q_i]^\top,\ Q_i > 0 \text{ (symmetric positive definite)}$$

see Algorithm [**PCMP**(x)]

Remark 1. It is not difficult to see that if $\langle \nabla f_s(y), u - y\rangle > 0$ and $u \in D_s(z)$ then we have an improvement on x ($F(x) \leq F(u)$) since the facts

$$\langle \nabla f_s(y), u - y\rangle > 0 \text{ and } u \in D_s(z)$$

imply

$$f_s(u) - f_s(y) \geq \langle \nabla f_s(y), u - y\rangle > 0,$$
$$f_j(u) > F(z) \text{ for all } j \neq s \text{ and } u \in D.$$

The following sections will develop this raw algorithm into a runnable code, as well as describing the first computational experiments.

3. Local Search: a convergent algorithm

This section is devoted to one of the important and difficult problems in nonconvex optimization; local maximum search is known to be in the NP-hard class in convex maximization [8]. The convex maximization is a particular case of (PCMP) when

$m = 1$. Therefore the local maximum search of (PCMP) clearly belongs to the NP-hard class.

In the literature on global optimization we can often find the phrase *let a local solution be given...* but in practice, an efficient algorithm should handle this assumption since it occurs in an inner loop of the global maximum search. Here, we highlight a practical algorithm to address this important issue.

First, let us enhance our notations with

$$\mathcal{L}_f(\alpha) = \{x \mid f(x) \leq \alpha\},$$
$$J(x) = \{j \in M \mid f_j(x) > F(x)\}$$

for Lebesgue's set of $f(\cdot)$ at α and a Lebesgue related index set. We refine $J(x)$ into $J'(x) = \{j \in M \mid f_j(x) > F(x), \quad \mathcal{L}_{f_j}(F(x)) \neq \emptyset\}$ and accordingly M as $M' = I(x) \cup J'(x)$; in practice, it may happen that $J'(x) \neq J(x)$ at a given x due to rounding errors, requiring refinement in the local search algorithm or else a premature emptiness stopping criterion is wrongly detected.

Let x^k be a feasible point in (PCMP), by definition of $F(\cdot)$ both $x^k \notin \mathcal{L}_{f_j}(F(x^k))$ for all $j \in J(x^k)$ and $I(x^k) \neq \emptyset$ are true.

Remark 2. As an initial feasible point, we choose the maximum over all quadratic minimizations,

$$x^0 = \max_{m \in M}\{\min_x\{f_m(x) \mid x \in D\}\}.$$

In the remainder of the section, for the sake of conciseness we will write I, J instead of $I(x^k)$, $J(x^k)$.

We introduce a set of polytopes P^k related to current point x^k:

$$P^k = \left\{ x \in \mathbb{R}^n \mid \begin{array}{ll} \langle \nabla f_j(x^k), x\rangle \geq \langle \nabla f_j(x^k), v^j\rangle, & j \in J \\ \langle \nabla f_i(x^k), x\rangle \geq \langle \nabla f_i(x^k), x^k\rangle, & i \in I \end{array} \right\}$$

where $v^j = \arg\max\{\langle \nabla f_j(x^k), x\rangle \mid x \in \mathcal{L}_{f_j}(F(x^k))\}, \quad j \in J$ could be analytically solved under the quadratic assumption:

$$\begin{aligned} v &= \arg\max\{\langle d, x\rangle \text{s.t.} \frac{1}{2}\langle Qx, x\rangle + \langle l, x\rangle + \gamma \leq \xi\} \\ &= Q^{-1}\left(d\sqrt{\frac{\langle Q^{-1}l, l\rangle + 2(\xi - \gamma)}{\langle Q^{-1}d, d\rangle}} - l\right) \end{aligned} \tag{5}$$

where d, Q, l, γ are appropriately associated with functions $f_j(\cdot)$ and $\xi = F(x^k)$. (see algorithm [**Local Search**(D, M)])

Remark 3. In fact, theoretically we need examine only one active function i.g. $m \in I$ instead of $m \in M'$; however, we are looking for a point in the complement of Lebesgue's set of $F(\cdot)$, therefore a much symmetric behavior comes from selecting

[**Local Search**(D, M)**]**
Let $x^k \in D$ **be given**
for all $m \in M'$
  $w^m = \arg\max\langle \nabla f_m(x^k), x\rangle$ **s.t.** $x \in D \cap P^k$
endforall
$w^r = \arg\max\{\min_{i \in M} f_i(w^m) \mid m \in M'\}$
if $\| w^r - x^k \| \leq \epsilon$
 then stop /* local solution found */
 else $k = k+1; x^k = w^r$**; goto 2;**

all functions in $M' = I \cup J$ instead. We could notice too, in step 3, that while w^r is extracted from a $M \times M'$ array of values, it is actually found through a one-dimensional loop along all objectives $f_i(\cdot), \quad i \in M$.

Proposition 2. *Under the assumptions mentioned about D and $F(\cdot)$, numerical sequence $\{F(x^k)\}$ is nondecreasing and convergent. Moreover,* $\lim_{k\to\infty} x^k$ *is a stationary point of (PCMP).*

Proof. Since $w^r = \arg\max\{\langle f_r(x^k), x\rangle \mid x \in D \cap P^k\}$, we have $w^r \in D \cap P^k$. By definition of polytope P^k and due to the convexity of the functions, $w^r \in P^k$ implies

$$0 \leq \langle \nabla f_i(x^k), w^r - x^k\rangle \leq f_i(w^r) - f_i(x^k), \quad i \in I,$$
$$0 \leq \langle \nabla f_j(x^k), w^r - v^j\rangle = \lambda\langle \nabla f_j(v^j), w^r - v^j\rangle \leq \lambda(f_j(w^r) - f_j(v^j)), \quad j \in J$$

where $\lambda > 0$ is directly derived from (5). This means $F(x^k) \leq F(x^{k+1})$. The compactness of D and the continuity of $F(\cdot)$ complete the proof of the convergence.

Now, let us prove the stationarity of the accumulation point. Assume that $z = x^k$ is an accumulation point for some large k. Notice, that our function $F(\cdot)$ is directionally differentiable [2]. So, we have to prove

$$\frac{\partial F(z)}{\partial g} \leq 0 \text{ for all } g \in \Gamma(z),$$

where $\Gamma(z)$ is the cone of feasible directions of the set D at the point z.

Assume the opposite. Then there exists a $g^0 \in \Gamma(z)$ such that $\frac{\partial F(z)}{\partial g^0} > 0$. By the definition of the feasible direction (see [2] p. 252) there exists $\lambda > 0$ and a sequence $\{z^s\}$ such that

$$z^s \in D, \quad z^s \neq z, \quad z^s \to z,$$
$$v^s = \frac{z^s - z}{\| z^s - s \|} \to v^0, \quad g^0 = \lambda v^0.$$

Since $F(z^s) = F(z + z^s - z) = F(z + \| z^s - z \| v^s) = F(z + \gamma_s v^s)$ with $\gamma_s = \| z^s - z \| > 0$, then

$$\lim_{s\to\infty} \frac{1}{\gamma_s}(F(z^s) - F(z)) = \lim_{s\to\infty} \frac{1}{\gamma_s}(F(z + \gamma_s v^s) - F(z)) = \frac{\partial F(z)}{\partial v^0}.$$

Hence,

$$F(z^s) = F(z) + \gamma_s \frac{\partial F(z)}{\partial v^0} + o(\gamma_s).$$

On the other hand, we have

$$\frac{\partial F(z)}{\partial v^0} = \frac{1}{\lambda} \lim_{\alpha \to +0} \frac{1}{\alpha \lambda^{-1}} (F(z + \alpha\lambda^{-1}(\lambda v^0)) - F(z)) = \frac{1}{\lambda} \frac{\partial F(z)}{\partial g^0}.$$

So we obtain that $F(z^s) > F(z)$ for sufficiently large s.

For sufficiently large s and k we also have $z^s \in P^k$, that implies $(F(z) =)F(x^{k+1}) \geq F(z^s)$ since $w^m = \arg\max\{\langle \nabla f_m(z), x\rangle \mid P^k \cap D\}$ for all m. This contradiction $F(z) \geq F(z^s) > F(z)$ completes the proof.

Remark 4. In the above algorithm, the stopping criterion comes from the emptiness of polytope P^k; however, in practice, we observed a slow speed of convergence and a tailing off effect close to a local maximum. Therefore, we adopt a multiresolution scheme and early force emptyness of P^k for a given accuracy $\varepsilon = 10^l \times \epsilon$ (a multiple of actual tolerance ϵ); then we refine $\varepsilon = \varepsilon/10$ until $\varepsilon \leq \epsilon$. Under this multiresolution scheme, we get much faster convergence without a tailing off effect in the examples reported below but this practical rule of the thumb requires a thorough sensibility analysis to prove the speed of convergence from coarse to the finest resolution. Under this scheme, polytopes P^k_ε are introduced as inner approximation of the above P^k.

$$P^k_\varepsilon = \left\{ x \in \mathbb{R}^n \mid \begin{array}{ll} \langle \nabla f_j(x^k), x\rangle \geq \langle \nabla f_j(x^k), v^j\rangle + \varepsilon, & j \in J \\ \langle \nabla f_i(x^k), x\rangle \geq \langle \nabla f_i(x^k), x^k\rangle + \varepsilon, & i \in I \end{array} \right\}$$

4. Some Details of the (PCMP) algorithm

4.1. SEPARATION: AN INTERSECTION GRAPH

Given a local solution z, in order to improve the best known solution we have to look for a point in $\mathrm{clco}(D_m(z))$, which is a difficult problem whose construction was bypassed in [10] by recoursing to Caratheodory's existence theorem. In this section, we borrow from linear programming, the well-known equivalence **separation ≡ optimization**, to notice that whenever two Lebesgue's sets of the functions $f_i(\cdot), f_j(\cdot)$ at the level $F(z)$ are disjoint then it's likely to retrieve a $\mathrm{clco}(D_m(z))$ point inbetween. Once more, while from theoretical viewpoint, $\mathrm{clco}(D_m(z))$ is searched for, for only one $m \in M$, a better behavior was observed under isotropic retrieval $\cap_{m\in M}\mathrm{clco}(D_m(z))$. Here, we address an intersection graph construction, namely we build a symmetric graph $\mathcal{G}(z) = (V, E(z))$ having $V = \{\mathcal{L}_{f_i}(F(z)),\quad i \in M\}$ as vertices and $E(z) = \{(f_i, f_j) \mid D \cap \mathcal{L}_{f_i}(F(z)) \cap \mathcal{L}_{f_j}(F(z)) \neq \emptyset\}$ as edges. To decide whether Lebesque's sets $\mathcal{L}_{f_i}(F(z))$ and

[**Intersection edge**(f_i, f_j)**]**
$k = 0$**;**
forever do
$x_i^k = \arg\min\{f_i(x) \mid x \in D \cap P_j^k\}$**;**
$x_j^k = \arg\min\{f_j(x) \mid x \in D \cap P_i^k\}$**;**
if $(f_i(x_i^k) \le F(z)$ **and** $f_j(x_i^k) \le F(z))$ **or** $(f_i(x_j^k) \le F(z)$ **and** $f_j(x_j^k) \le F(z))$
then return edge(f_i, f_j)**;**
if $(f_i(x_i^k) > F(z)$ **and** $f_j(x_i^k) > F(z))$ **or** $(f_i(x_j^k) > F(z)$ **and** $f_j(x_j^k) > F(z))$
then return no edge(f_i, f_j)**;**
enddo

$\mathcal{L}_{f_j}(F(z))$ are separated in D, we turn the separation problem into both optimization problems:

$$\min f_i(x) \text{s.t. } x \in D \cap \mathcal{L}_{f_j}(F(z));$$
$$\min f_j(x) \text{s.t. } x \in D \cap \mathcal{L}_{f_i}(F(z)).$$

Once again, we introduce a set of polytopes that approximate Lebesgue's sets $\mathcal{L}_{f_i}(F(z))$ and $\mathcal{L}_{f_j}(F(z))$ from the outside by piecewise linear hyperplanes at some points on level sets. We have $f_i(\bar{x_i}^k) + \langle \nabla f_i(\bar{x_i}^k), x - \bar{x_i}^k \rangle \le f_i(x) \le F(z)$ using convexity and Lebesgue's set definition; then selecting $\bar{x_i}^k$ on level set, $f_i(\bar{x_i}^k) = F(z)$, we approximate Lebesgue's set as announced: $P_i^k = \cap_{k \ge 0}\{\langle \nabla f_i (\bar{x_i}^k), x - \bar{x_i}^k \rangle \le 0\}$, with initial condition $P_i^0 = \mathbb{R}^n$ being consistent with the algorithm below. We introduce, in the same fashion: $P_j^k = \cap_{k \ge 0}\{\langle \nabla f_j(\bar{x_j}^k), x - \bar{x_j}^k \rangle \le 0\}$ and $P_j^0 = \mathbb{R}^n$. (see algorithm [**Intersection edge**(f_i, f_j)])

The correctness of this algorithm follows $P_i^k \supseteq P_i^{k+1}$ and $P_j^k \supseteq P_j^{k+1}$ so that either condition on edge(f_i, f_j) is fulfilled for some k.

In the sequel, we use "CLCO" as a shortcut of $\cap_{m \in M}\text{clco}(D_m(z))$

4.2. INNER CLCO APPROXIMATION

For all separated objectives in $\mathcal{G}(z)$, we could guess whether a point belongs to $\cap_{m \in M}\text{clco}(D_m(z))$ in the following way; let us consider f_i, f_j be separated in D with $\bar{x_i}, \bar{x_j}$ as points on respective level sets $f_i(\bar{x_i}) = F(z)$, $f_j(\bar{x_j}) = F(z)$ then $x = \frac{1}{2}(\bar{x_i} + \bar{x_j})$ is likely to belong to $\cap_{m \in M}\text{clco}(D_m(z))$ or to be a good starting point to look for a local maximum over all objectives except i and j.

Let M_{ij} denote $M \backslash \{i, j\}$,

$$D_{ij} = D \cap \{\langle \nabla f_j(\bar{x_j}), x - \bar{x_j} \rangle \ge 0, \langle \nabla f_j(\bar{x_j}), x - \bar{x_j} \rangle \ge 0\}.$$

Then solving **Local Search**(D_{ij}, M_{ij}) ends up with either a better point in $D_{ij} \subset \cap_{m \in M}\text{clco}(D_m(z))$ or no better point in D_{ij} which does not mean $\cap_{m \in M}\text{clco}(D_m(z))$ is empty.

4.3. OUTER CLCO APPROXIMATION

In the preliminary algorithm [10] a random generation of points on the level set was suggested. Sometimes, the partitioning of level sets may drastically reduce the number of attempts to improve the current solution through cutting planes; in multiknapsack maximization, for instance, a *geodesic partitioning property* [4] allows us to reduce random generation to some representative cone and yields an efficient algorithm.

In $(PCMP)$ case this appealing technique has to be carried over $\cap_{m\in M}$clco $(D_m(z))$, an open problem; therefore, we try to approximate $\cap_{m\in M}\text{clco}(D_m(z))$ from outside by *guessing* a good point on the level set.

As a first attempt towards this goal, we substitute the pure random point generation by the selection of tangent point to $\mathcal{L}_{f_s}(F(z))$ along each constraint direction and then solve linearized problem at that deterministic point y. Starting with polytope $\Phi = D$, we refine it through cutting planes; let us compute $r = arg\min_j\{f_j(u) \mid j \in M\}$ and index set of **active constraints** in current Φ at u, (where $u = \arg\max\{\langle\nabla f_s(y), x\rangle / x \in \Phi\}$)

$$K(u) = \{l \mid [Pu]_l = p_l\}$$

where P (resp. $P(u)$) denotes the matrix of constraints (resp. active constraints at u) of polytope Φ; under the full dimensionality assumption, $[P(u)]^{-1}$ definitely exists. Let Y be the set (columnwise) of points on the level set $f_r(\cdot)$ intersected by the active cone, namely

$$Y = u \otimes e^{\top} - [P(u)]^{-1}\alpha^r,$$

where $\otimes$ denotes the kronecker product and $\alpha^r \in \mathbb{R}^n_+$ solves the quadratic equations for every column vector y^i of Y

$$f_r(y^i(\alpha)) = F(z).$$

Then vector d, found as the solution of the linear system $Yd = ne$ yields a new cut for polytope Φ, as is well known in the global optimization field; notice that right handside introduces a normalizing factor to avoid tailing off effects since it is usual to observe such an effect in similar algorithms [4, 9].

5. Computational Experiments

We present software experiments on the following small examples. In order to thoroughly test the algorithm, we introduce some variants like discarding useless objectives, extending domain D...

Example 1. (see Figure 1)

$$f_1(x) = x_1^2 + (x_2+4)^2 - 36,$$
$$f_2(x) = (x_1+8)^2 + (x_2-3)^2 - 36,$$
$$f_3(x) = x_1^2 + (x_2-8)^2 - 16,$$
$$f_4(x) = (x_1-8)^2 + (x_2-3)^2 - 53,$$
$$f_5(x) = (x_1-10)^2 + (x_2+10)^2 - 4$$

subject to

$$D = \{x \in \mathbb{R}^2 \mid -4 \le x_1 \le 10, \quad -6 \le x_2 \le 8, \quad x_1 - x_2 \le 10\}.$$

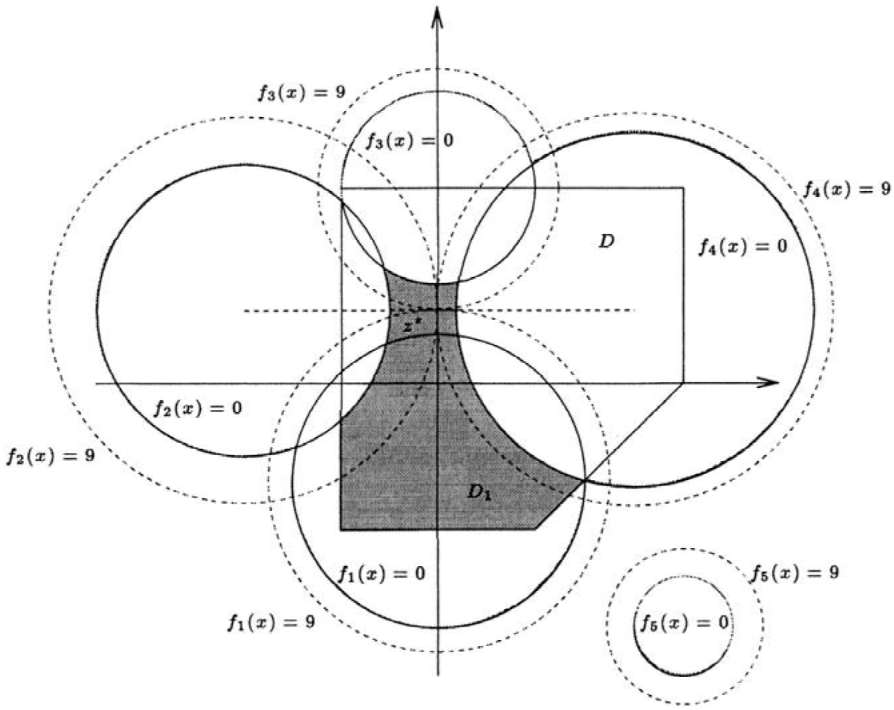

Figure 1. A simple example 1.

Example 2. (see Figure 2)

$$f_1(x) = x_1^2 + (x_2 + 2)^2 - 9,$$
$$f_2(x) = 9(x_1 + 3)^2 + 4x_2^2 - 36,$$
$$f_3(x) = (x_1 + 1)^2 + (x_2 - 4)^2 - 4,$$
$$f_4(x) = \frac{1}{9}(x_1 - 3)^2 + \frac{1}{36}(x_2 - 4)^2 - 1,$$
$$f_5(x) = (x_1 - 5)^2 + (x_2 + 5)^2 - 1$$

subject to box constraint:

$$D = \{x \in \mathbb{R}^2 \mid -2 \le x_1 \le 5, \quad -3 \le x_2 \le 4\}.$$

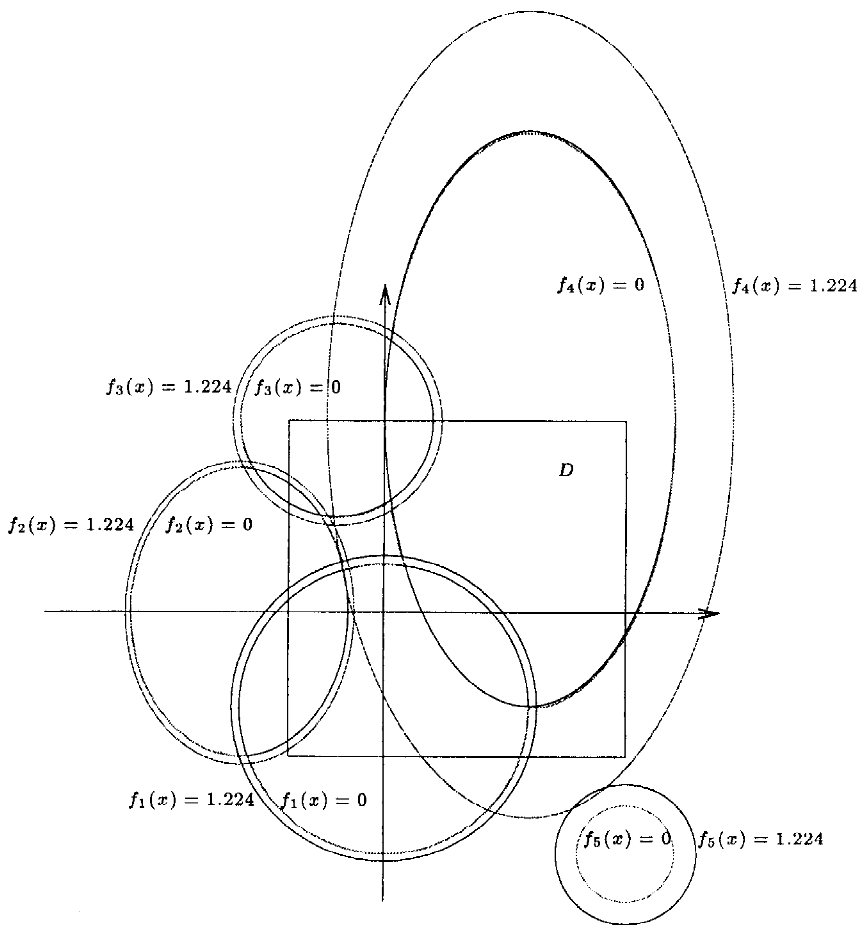

Figure 2. A non-trivial example 2

Variant to fully test intersection graph:

Example 3.

$$
\begin{aligned}
f_1(x) &= x_1^2 + (x_2+4)^2 - 36,\\
f_2(x) &= (x_1+8)^2 + (x_2-3)^2 - 36,\\
f_3(x) &= x_1^2 + (x_2-8)^2 - 16,\\
f_4(x) &= (x_1-8)^2 + (x_2-3)^2 - 53,\\
f_5(x) &= (x_1-10)^2 + (x_2+10)^2 - 4\\
f_6(x) &= (x_1+5)^2 + (x_2+4)^2 - 1
\end{aligned}
$$

subject to

$$D = \{x \in \mathbb{R}^2 \mid -4 \le x_1 \le 10, \quad -6 \le x_2 \le 8, \quad x_1 - x_2 \le 10\}.$$

Variants to handle non-regularity; a problem happens to be non-regular (see Definition 3) as soon as one center of the ellipses lies on the boundary of D while the remaining ellipses are not active at this center. In local maximum search, this leads to a null gradient side effect resulting in an erratic (and slow) trajectory towards an accumulation point. In order to circumvent this bad effect in practice, we escape from a local search and directly apply the inner clco approximation; this considerably speeds up the algorithm since a deep clco point is quickly retrieved.

Example 4.

$$
\begin{aligned}
f_1(x) &= x_1^2 + (x_2+4)^2 - 36,\\
f_2(x) &= (x_1+8)^2 + (x_2-3)^2 - 36,\\
f_3(x) &= x_1^2 + (x_2-8)^2 - 16,\\
f_4(x) &= (x_1-8)^2 + (x_2-3)^2 - 53,
\end{aligned}
$$

subject to

$$D = \{x \in \mathbb{R}^2 \mid -4 \le x_1 \le 10, \quad -6 \le x_2 \le 8, \quad x_1 - x_2 \le 10\}.$$

Example 5.

$$
\begin{aligned}
f_1(x) &= x_1^2 + (x_2+2)^2 - 9,\\
f_2(x) &= 9(x_1+3)^2 + 4x_2^2 - 36,\\
f_3(x) &= (x_1+1)^2 + (x_2-4)^2 - 4,\\
f_4(x) &= \frac{1}{9}(x_1-3)^2 + \frac{1}{36}(x_2-4)^2 - 1,
\end{aligned}
$$

subject to box constraint:

$$D = \{x \in \mathbb{R}^2 \mid -2 \le x_1 \le 5, \quad -3 \le x_2 \le 4\}.$$

Variants lifted to three dimensions. The same examples are lifted to a 3D box and the size of box was chosen as being either symmetric or non-symmetric to measure how the algorithm could escape from very good point lifted from the 2D case whenever the box is large enough.

Example 6.

$$f_1(x) = x_1^2 + (x_2 + 4)^2 + x_3^2 - 36,$$
$$f_2(x) = (x_1 + 8)^2 + (x_2 - 3)^2 + x_3^2 - 36,$$
$$f_3(x) = x_1^2 + (x_2 - 8)^2 + x_3^2 - 16,$$
$$f_4(x) = (x_1 - 8)^2 + (x_2 - 3)^2 + x_3^2 - 53,$$
$$f_5(x) = (x_1 - 10)^2 + (x_2 + 10)^2 + x_3^2 - 4$$

subject to

$$D = \{x \in \mathbb{R}^2 \mid -4 \le x_1 \le 10, \quad -6 \le x_2 \le 8, \quad -2 \le x_3 \le 2, \quad x_1 - x_2 \le 10\}.$$

Example 7.

$$f_1(x) = x_1^2 + (x_2 + 4)^2 + x_3^2 - 36,$$
$$f_2(x) = (x_1 + 8)^2 + (x_2 - 3)^2 + x_3^2 - 36,$$
$$f_3(x) = x_1^2 + (x_2 - 8)^2 + x_3^2 - 16,$$
$$f_4(x) = (x_1 - 8)^2 + (x_2 - 3)^2 + x_3^2 - 53,$$
$$f_5(x) = (x_1 - 10)^2 + (x_2 + 10)^2 + x_3^2 - 4$$

subject to

$$D = \{x \in \mathbb{R}^2 \mid -4 \le x_1 \le 10, \quad -6 \le x_2 \le 8, \quad -2 \le x_3 \le 4, \quad x_1 - x_2 \le 10\}.$$

Example 8.

$$f_1(x) = x_1^2 + (x_2 + 2)^2 + x_3^2 - 9,$$
$$f_2(x) = 9(x_1 + 3)^2 + 4x_2^2 + x_3^2 - 36,$$
$$f_3(x) = (x_1 + 1)^2 + (x_2 - 4)^2 + x_3^2 - 4,$$
$$f_4(x) = \frac{1}{9}(x_1 - 3)^2 + \frac{1}{36}(x_2 - 4)^2 + x_3^2 - 1,$$
$$f_5(x) = (x_1 - 5)^2 + (x_2 + 5)^2 + x_3^2 - 1$$

subject to box constraint:

$$D = \{x \in \mathbb{R}^2 \mid -2 \le x_1 \le 5, \quad -3 \le x_2 \le 4, \quad -2 \le x_3 \le 2\}.$$

Example 9.

$$f_1(x) = x_1^2 + (x_2 + 2)^2 + x_3^2 - 9,$$
$$f_2(x) = 9(x_1 + 3)^2 + 4x_2^2 + x_3^2 - 36,$$
$$f_3(x) = (x_1 + 1)^2 + (x_2 - 4)^2 + x_3^2 - 4,$$
$$f_4(x) = \frac{1}{9}(x_1 - 3)^2 + \frac{1}{36}(x_2 - 4)^2 + x_3^2 - 1,$$
$$f_5(x) = (x_1 - 5)^2 + (x_2 + 5)^2 + x_3^2 - 1$$

subject to box constraint:

$$D = \{x \in \mathbb{R}^2 \mid -2 \leq x_1 \leq 5, \quad -3 \leq x_2 \leq 4, \quad -2 \leq x_3 \leq 4\}.$$

All examples were run under a digital PWS500 Unix Workstation using CPLEX solver and C++. In the tables below, we split the results into 2 parts: one for inner clco approximation only and one for the complete loop through inner along with outer clco approximation.

The meanings for all columns in tables follow:

CP: #hyperplanes in the finest outer CLCO approximation,
best: best known value on corresponding approximation,
fat: how bad outer CLCO approximation could be, namely given the minimum value over each objective f_m considered by itself, i.e. $q = \min\{f_m(y^m) \mid m \in M\}$ where $y^i = \arg\min_y\{f_i(y) \mid y \in \Phi\}$ and the actual best known value $Q = \min_{m \in M}\{f_m(y) \mid y \in \Phi\}$, this ratio (in percentage) is equal to $100 * (Q - q)/ \mid q \mid$; on the other hand, a small value means that outer clco approximation is tight,
time: overall (inner+outer CLCO) user time in seconds,
all remaining columns: number of quadratic minimizations and number of linear programs solved for related step written as #QPs:#LPs.

For outer clco approximation, we test different strategies for refining the current approximation: given $u = \arg\max$ of linearized problem, we select either **worse** objective or **closest** to active objective at u; we also test symmetrization of clco as for the inner approximation and generate a tangent point along all constraints either for **all** or only for the **active** objectives at a local maximum z. Table I reports a worse selection from u and all objectives tangent point generation, Table II reports a worse selection from u and active objectives only and table III reports the closest to active selection (if any) and active objectives only.

As an initial concluding remark, we would say that an inner clco approximation achieves a very good solution with a small amount of effort and that outer clco approximation is not tight and scarcely improves the inner clco result. From the tables, we could see that outer CLCO approximation improves the result only for

Table I. Uuter approximation cutting plane: worse inactive + all objectives

	Inner CLCO				Inner+outer CLCO							
Ex.	Local	Inner	Graph	Best	Best	Local	Inner	Graph	Outer	CP	Fat%	Time
1	5:476	7:96	48:0	11.8676	11.8676	5:476	7:96	48:0	0:720	47	156	3.737
3	6:780	8:64	65:0	11.8671	11.8671	6:780	8:64	65:0	0:1524	49	156	8.045
4	4:340	2:64	36:0	11.8677	11.8677	4:340	2:64	36:0	0:960	51	156	4.735
2	5:64	5:39	47:0	1.2237	1.2237	5:64	5:39	47:0	0:1630	50	12	7.488
5	4:64	2:16	58:0	1.22335	1.22335	4:64	2:16	58:0	0:1216	50	121	5.423
6	5:418	7:93	48:0	15.8672	15.8678	5:624	17:279	96:0	0:7962	921	179	784.958
7	5:418	7:93	48:0	15.8672	15.8672	5:418	7:93	48:0	0:15445	699	169	31064.289
8	5:46	0:18	48:0	5.22369	5.22392	5:57	0:21	69:0	0:8916	654	201	1259.574
9	5:46	0:18	48:0	5.22369	**16.8056**	5:73	0:42	66:0	0:4821	213	425	349.588

Table II. Outer approximation cutting plane: worse inactive + active objectives

	inner CLCO				inner+outer CLCO							
ex.	Local	Inner	Graph	Best	Best	Local	Inner	Graph	Outer	CP	Fat%	Time
1	5:476	7:96	48:0	11.8676	11.8676	5:476	7:96	48:0	0:168	23	155	1.220
3	6:780	8:64	65:0	11.8671	11.8671	6:780	8:64	65:0	0:168	23	155	2.138
4	4:340	2:64	36:0	11.8677	11.8677	4:340	2:64	36:0	0:168	23	155	1.052
2	5:64	5:39	47:0	1.2237	1.2237	5:64	5:39	47:0	0:1012	51	122	3.602
5	4:64	2:16	58:0	1.22335	1.22335	4:64	2:16	58:0	0:1012	51	122	3.636
6	5:418	7:93	48:0	15.8672	15.8672	5:418	7:93	48:0	0:1272	246	176	69.856
7	5:418	7:93	48:0	15.8672	15.8672	5:418	7:93	48:0	0:394	155	160	8.868
8	5:46	0:18	48:0	5.22369	5.22369	5:46	0:18	48:0	0:386	182	197	13.581
9	5:46	0:18	48:0	5.22369	**16.8056**	5:73	0:42	66:0	0:2671	236	365	272.398

the 3D case at a heavy LPs price. In fact, outer CLCO is involved in proving actual emptiness of CLCO whenever inner CLCO fails to return a point inside. In above experiments, outer CLCO exhibits a zigzagging behavior when we get closer to an extreme point of CLCO (overdetermined active cone and singular matrix in retrieving next cutting plane, see Figure 3); in order to prevent this heavy time consumption as late as possible, in the next section, we deal with the question of how to improve further inner CLCO approximation.

Table III. Outer approximation cutting plane: closest to active + active objectives

	inner CLCO				inner+outer CLCO							
ex.	Local	Inner	Graph	Best	Best	Local	Inner	Graph	Outer	CP	Fat%	Time
1	5:476	7:96	48:0	11.8676	11.8676	5:476	7:96	48:0	0:964	74	159	5.709
3	6:780	8:64	65:0	11.8671	11.8671	6:780	8:64	65:0	0:1444	79	156	9.065
4	4:340	2:64	36:0	11.8677	11.8677	4:340	2:64	36:0	0:964	74	159	5.476
2	5:64	5:39	47:0	1.2237	1.2237	5:64	5:39	47:0	0:860	65	120	4.849
5	4:64	2:16	58:0	1.22335	1.22335	4:64	2:16	58:0	0:868	66	123	12.001
6	5:418	7:93	48:0	15.8672	15.8672	5:418	7:93	48:0	0:1194	517	178	252.815
7	5:418	7:93	48:0	15.8672	15.8672	5:418	7:93	48:0	0:88	51	142	1.173
8	5:46	0:18	48:0	5.22369	5.22369	5:46	0:18	48:0	0:4688	328	204	1159.106
9	5:46	0:18	48:0	5.22369	**16.8056**	5:73	0:42	66:0	0:813	245	371	60.163

5.1. TOWARDS AN INCLUSION-EXCLUSION ALGORITHM

Despite the poor outer clco approximation scheme, we suspect that the intersection graph endows nonconvex $\cap_{m\in M}(D_m(z))$ with a very rich structure. We only exploited nonadjacency between two vertices (separation between two objectives) to the inner approximate clco. In this section, we provide a few arguments to open up a possible further study of intersection graph since experiments for 3D cases have shown that it could be difficult to escape from a rather *good* clco situation. In figure 3 we depicted the situation when we end up with an empty inner approximation (upto tolerance) while the outer approximation is far from actual clco extreme points. In the picture on the left, we assume that inner approximation of $\cap_{m\in\{i,j,k\}}\text{clco}(D_m(z))$ has to be augmented with a nonconvex part coming from the outer clco approximation and in the right-hand picture we are concerned with an actual situation in the practical algorithm. So the next step would be to deal with the following (PCMP): let $F_{ij}(x) = \min_{m\in M_{ij}}\{f_m(x)\}$ for nonadjacent f_i, f_j in intersection graph

$$\begin{aligned}
&\text{maximize } F_{ij}(x)\\
&\text{subject to } x \in D\\
&\langle \nabla f_i(\bar{x}_i), x - \bar{x}_i\rangle \geq 0 \qquad (6)\\
&\langle \nabla f_j(\bar{x}_j), x - \bar{x}_j\rangle \geq 0 \qquad (7)
\end{aligned}$$

where as above $M_{ij} = M\backslash\{i, j\}$ and constraints (6) and (7) include some convex part of clco related to the closest points (apart) $\bar{x}_i, \bar{x}_j$ on level sets $f_i(x) = f_j(x) = F(z)$. It could be extended to adjacent f_i, f_j as well, provided the closest points $\bar{x}_i, \bar{x}_j$ in $\mathcal{L}_{f_i}(F(z)) \cap \mathcal{L}_{f_i}(F(z))$ are *close enough* to a clco extreme point. In other words, patching a convex inner clco approximation with nonconvex parts from an outer clco approximation is a challenging issue around the famous inclusion-exclusion principle on property ψ on sets A, B, C:

$$\psi(A\cup B\cup C) = \\ \psi(A) + \psi(B) + \psi(C) - \psi(A\cap B) - \psi(A\cap C) - \psi(B\cap C) + \psi(A\cap B\cap C)$$

In practical experiments we deal with only two sets and outline above how to deal with three sets; as for its general setting with n sets, it remains far beyond the scope of these prospective concluding remarks.

Another minor remark concerns the intersection graph $\mathcal{G}(z)$ itself. It is worthwhile concentrating on connected components in turn since each corresponding clco is a candidate for improvement; however, no clear understanding of their relative *degree of improvement* is available. In the same way, modular decomposition and the maximum clique of $\mathcal{G}(z)$ could help in choosing a *direction* of improvement without any clear understanding of such an impact.

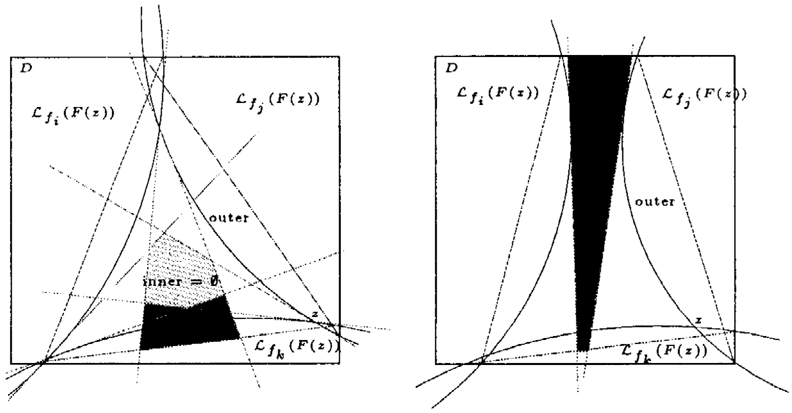

Figure 3. An inclusion–exclusion clco approximation.

6. Concluding remarks

In this article, we presented a practical algorithm to solve piecewise convex maximization problems. It has proved to be efficient for finding an optimal solution in $\mathbb{R}^2$ and at the same time it suggested how difficult it could be to escape from a very good local maximum. We introduced the intersection graph between objectives and noticed how it influences the direction of the search. It compares favorably with *standard* techniques from the global optimization that amount to outer approximating the region of interest through hyperplanes. It opens up a field which could be studied more thoroughly:

- the structure of this graph (connected components, the maximum clique and its relationship with Helly's Theorem [3],...)
- its connection to the inclusion-exclusion principle applied to convex and nonconvex parts of the crucial parts in the solution space,
- its relationships with finding the real solutions of a system of nonlinear equations [1].

To our knowledge, any further improvement will have an impact on solving nonconvex optimization problems.

7. Acknowledgements

We would like to thank two anonymous referees for their valuable suggestions.

References

1. Bulatov, V. P. (2000) Numerical method for finding all real roots of systems of nonlinear equations. *Zh. Vychisl. Mat. Mat. Fiz.*, 40(3): 348–355.

2. Dem'yanov, V.F. and Vasil'ev, L.V. (1985) *Nondifferentiable Optimization.* Optimization Software, Inc., Publications Division, New York.
3. Eggleston, H.G. (1966) *Convexity.* Cambridge University Press.
4. Fortin, D. and Tsevendorj, I. (2000) Global optimization and multiknapsack: a percolation algorithm. *Le rapport de recherche de l'INRIA, France* 3912: 19.
5. Horst, R., Pardalos, P.M. and Van Thoai, N. (1995) *Introduction to Global Optimization.* Kluwer Academic.
6. Horst, R. and Tuy, H. (1990). *Global Optimization*. Springer, Berlin.
7. Kim, D. and Pardalos, P. M. (2000) A dynamic domain contraction algorithm for nonconvex piece-wise linear network flow problems. *J. Global Optim.*, 17: 225–234.
8. Pardalos, P. M. and Schnitger, G. M. (1988) Checking local optimality in constrained quadratic problems is NP-hard. *Oper. Res. Letters*, 7: 33–35.
9. Strekalovsky, A. S. and Tsevendorj, I. (1998). Testing the R-strategy for a reverse convex problem. *J. Global Optim.*, 13(1): 61–74.
10. Tsevendorj, I. (2001) Piece wise-convex maximization problems : Global optimality conditions. *J. Global Optim.*, 21: 1–14.