

The radar method: An effective line search for piecewise linear concave functions*

C. Beltran-Royo[†]

April 17, 2007

Abstract

The maximization of one-dimensional piecewise linear concave (OPLC) functions arises in the line search associated with the maximization of piecewise linear concave functions (e.g. Kelley cutting plane method). The OPLC line search is usually done by the *next-break-point method*, where one goes from break point to break point up to the optimum. If the number of break points is large this method will be computationally expensive. One can also use some classical derivative-free line search method as for example the *golden section* method. Such methods do not take advantage of the OPLC geometry. As an alternative, we propose an improved version of the so-called *radar* method, which maximizes an OPLC function by maximizing successive outer approximations. **We prove superlinear and finite convergence of the radar method. Furthermore, our computational test shows that the radar method is highly effective independently from the number of break points.**

Keywords: Piecewise linear concave function, line search, radar method, next-break-point method, golden section method, Kelley cutting plane method.

1 Introduction

In this paper we study the convergence and performance of **an improved version of the so-called radar method** [BH05], a procedure intended to maximize a one-dimensional piecewise linear concave (OPLC) function.

OPLC functions are nondifferentiable and arise, for example, in the line search associated with the maximization of a piecewise linear concave (PLC) function by

*This work was partially supported by Logilab, HEC, University of Geneva and the Spanish government, MCYT subsidies TIC2003-05982-C05-05 and MTM2006-14961-C05-05.

[†]cesar.beltran@hec.unige.ch, Logilab, HEC, University of Geneva, Switzerland.

a direction following method. A typical source of PLC maximization is the Kelley cutting plane method [Kel60]. We assume that the OPLC function is explicitly given as the lower envelope of a finite set of lines.

When dealing with a line search problem one can compute an optimal point of the associated unidimensional function (*exact line search*) or an approximate optimal point (*approximate line search*). Most of the time the exact OPLC line search is done by the *next-break-point method*. Taking advantage of the well known geometry of the graph of an OPLC function, one goes from break point to break point of the OPLC function up to the optimum [HP04].

In the case of a large number of break points one may perform an approximate OPLC line search, by a truncated version of the next-break-point method. Sometimes, even a single step of the next-break-point is performed [CC90]. Exact or approximate OPLC line searches could also be done by using classic derivative-free line search methods such as the *golden section method* [BS79].

The relevance of the exact OPLC line search has been shown for example in [NBR00], where a PLC function is optimized by using an approximate subdifferential method combined with an OPLC line search (three versions of the OPLC line search are tested). Roughly speaking, the exact OPLC line search divides by two the total number of (costly) subdifferential iterations combined with an approximate OPLC line search. In [HP04] the usefulness of the exact OPLC line search is also demonstrated for the gradient projection method applied to the graph partitioning problem.

Our starting point is the radar method developed in [BH05]. In that paper an approximate line search is performed by optimizing an approximation to the OPLC function. This first version of the radar method was tested as an approximate line search for the subgradient method and clearly outperformed the Polyak step rule [Sho98]. In this paper, we improve the previous radar method and study the convergence of the improved version. Throughout the remaining of the paper we will refer to this improved version of the radar method when using the term *radar method*. The improved version has been successfully used as the exact line search method for a conjugate Rosen's gradient projection method used in piecewise linear concave optimization [Bel07].

This paper is organized as follows. In section 2 we state the OPLC problem and notation. In section 3 the radar method is introduced. A discussion on the convergence of the radar method is presented in section 4. Finally a numerical test and conclusions can be found in sections 5 and 6, respectively.

2 Statement of the problem

We consider an OPLC function $f(\alpha)$ defined as the lower envelope of the lines $\{r_j(\alpha)\}_{j \in J}$, where $J = \{1, \dots, m\}$ and $r_j(\alpha) = m_j\alpha + n_j$. That is $f(\alpha) =$

$\min_{j \in J} r_j(\alpha)$. Our objective is to solve the *OPLC problem* (see Fig. 1):

$$\alpha^* = \arg \max_{\alpha \in \mathbb{R}} f(\alpha), \quad (1)$$

Definition 1 We say that r_j is an *active line* at $\bar{\alpha}$ if $f(\bar{\alpha}) = r_j(\bar{\alpha})$.

Definition 2 A line r_j is called the *steepest active line* at $\bar{\alpha}$ if it has the greater slope among the active lines at $\bar{\alpha}$.

We recall here the definition of left and right derivatives, $f^-(\bar{\alpha})$ and $f^+(\bar{\alpha})$, respectively:

$$f^-(\bar{\alpha}) := \lim_{x \rightarrow \bar{\alpha}, x < \bar{\alpha}} \frac{f(x) - f(\bar{\alpha})}{x - \bar{\alpha}} \quad \text{and} \quad f^+(\bar{\alpha}) := \lim_{x \rightarrow \bar{\alpha}, x > \bar{\alpha}} \frac{f(x) - f(\bar{\alpha})}{x - \bar{\alpha}}.$$

3 The radar method

We concentrate now in the solution of the OPLC problem (1) by the radar method. Without loss of generality, we assume that α^* is the smallest optimal point in case of multiple optimizers, that f is parameterized such that $f(0) = 0$ and that $\alpha^* \geq 0$. The graph of f in the interval of interest $I = [0, \alpha^*]$ is the union of r line segments that joint the break points $p_i = (a_i, b_i) \in \mathbb{R}^2$, $i = 0, \dots, r$. We have that $p_0 = (0, 0)$ and $p_r = (\alpha^*, f(\alpha^*))$. The interval I is partitioned into

$$I_1 \cup I_2 \cup \dots \cup I_r = [a_0, a_1[\cup [a_1, a_2[\cup \dots \cup [a_{r-1}, a_r].$$

Considering that the line segments form the graph of a concave function, their slopes will be a decreasing sequence: $m_1 > m_2 > \dots > m_r > 0$. The point a_r is characterized as the only point where the left derivative of f is strictly **positive** and the right derivative is **negative or null**, i.e., $f^-(a_r) > 0$ and $f^+(a_r) \leq 0$. This property could be used to solve the global line search: Start at $a_0 = 0$, then iterate from break point to break point up to the first break point with a change of sign in the lateral derivatives (*next-break-point method*).

The number of iterations in the next-break-point method equals the number of break points. To avoid the potentially large number of iterations of the next-break-point method we use the *radar method*. Before of introducing the radar method we need to partition the set of lines that define f . We consider *active lines* in $[0, +\infty[$ indexed by J_a , that is, lines that intersect with the graph of f in more than one point within the interval $[0, +\infty[$. Analogously, we consider *inactive lines* indexed by J_i . We also consider *strictly positive lines* indexed by J^+ , that is, lines with strictly positive slope. Analogously, *negative lines* indexed by J^- (negative lines may have null slope). By combining these two attributes, we partition J into four classes :

$$J = J^+ \cup J^- = J_a^+ \cup J_i^+ \cup J_a^- \cup J_i^-.$$

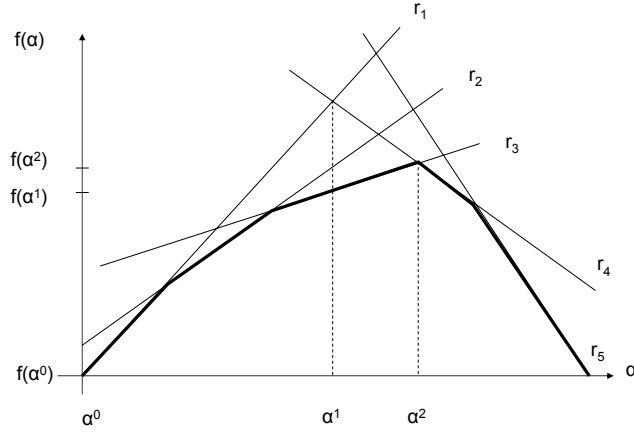


Figure 1: **One of the advantages of the radar method is that, usually, the number of iterations is smaller than the number of break points. In this case with four break points the radar method would take two iterations.**

As we can see in Fig. 1, at iteration k , the radar method approximates $f(\alpha)$ by only two of its defining lines: r_{j_1} for some $j_1 \in J_a^+$ and r_{j_2} for some $j_2 \in J^-$. **To introduce the radar method, we rename r_{j_1} and r_{j_2} by r^k and s^k , respectively.** r^k is the *steepest active line* at α^k and s^k is the *stopping negative line*, i.e., the first negative line that intersects with r^k as α increases. The intersection point of r^k and s^k is $(\alpha^{k+1}, r^k(\alpha^{k+1}))$, which gives the next radar iterate α^{k+1} and an upper bound to the optimal value ($r^k(\alpha^{k+1}) \geq f(\alpha^*)$). **In what follows, the slope and intercept of r^k are denoted by m^k and n^k , respectively. That is**

$$r^k(\alpha) = m^k \alpha + n^k.$$

See [BH05] for more details. The radar method can be summarized as follows:

Radar method

Step 1. *Initialization:* Set $\alpha^0 = 0$ and $k = 0$.

Step 2. *Compute next radar iterate:*

$$\alpha^{k+1} = \min \left\{ \frac{n^k - n_j}{m_j - m^k} \mid j \in J^- \right\},$$

Step 3. *Stopping criterion:* If $r^k(\alpha^{k+1}) = f(\alpha^{k+1})$ then stop since α^{k+1} maximizes $f(\alpha)$. Else, set $k = k + 1$ and go back to Step 1.

Remark 1 a) A single iteration version of the radar method was first used in [BH05], thus obtaining a suboptimal solution of the line search problem (approximate OPLC line search). In this paper, the aim of the radar method is to compute an optimal solution of the line search problem (exact OPLC line search).

b) The already mentioned next-break-point method replaces Step 2 of the radar method by:

$$\alpha^{k+1} = \min \left\{ \beta_j \mid \beta_j = \frac{n^k - n_j}{m_j - m^k}, \beta_j > \alpha^k, j \in J \right\}. \quad (2)$$

4 Convergence issues

4.1 Finite convergence of the radar method

Lemma 1 (Interval reduction) *If $\alpha^k \in [a_i, a_{i+1}[$ then:*

- a) $\alpha^{k+1} \in [a_{i+1}, \alpha^*]$,
- b) $f(\alpha^{k+1}) > f(\alpha)$, for all $\alpha \in [a_i, a_{i+1}[$.
- c) Furthermore, for $k \geq 0$ we have $f(\alpha^{k+1}) > f(\alpha^k)$.

Proof: (a.1) Let us see that $\alpha^{k+1} \geq a_{i+1}$: In the radar method, $p_{i+1} = (a_{i+1}, b_{i+1})$ is the first intersecting point that can be considered as a candidate to determine α^{k+1} , therefore, $\alpha^{k+1} \geq a_{i+1}$.

(a.2) Let us see that $\alpha^{k+1} \leq \alpha^*$: We distinguish two cases depending on whether the stopping line $s^k(\alpha)$ is active at α^* . In the first case we assume that $s^k(\alpha^*) = f(\alpha^*)$. (Proof by contradiction) In this case, if $\alpha^{k+1} > \alpha^*$ then $\delta := \alpha^{k+1} - \alpha^* > 0$. On the one hand, by definition $r^k(\alpha^{k+1}) = s^k(\alpha^{k+1})$. Furthermore, since r^k is strictly increasing, $r^k(\alpha^{k+1} - \delta) < r^k(\alpha^{k+1})$ and, since s^k is non increasing, $s^k(\alpha^{k+1} - \delta) \geq s^k(\alpha^{k+1})$, thus

$$r^k(\alpha^{k+1} - \delta) < r^k(\alpha^{k+1}) = s^k(\alpha^{k+1}) \leq s^k(\alpha^{k+1} - \delta).$$

Equivalently, $r^k(\alpha^*) < s^k(\alpha^*) = f(\alpha^*)$, which contradicts the definition of $f(x)$ as the lower envelope of the lines $\{r_j(\alpha)\}_{j \in J}$.

In the second case we assume that $s^k(\alpha^*) > f(\alpha^*)$. (Proof by contradiction) Suppose that $\alpha^{k+1} > \alpha^*$. We have that $f(\alpha^*) = r_{i_0}(\alpha^*)$ for some $i_0 \in J_-$ (note that r_{i_0} is not s^k since $s^k(\alpha^*) > r_{i_0}(\alpha^*)$). Then we will have that $r^k(\alpha^*) \geq r_{i_0}(\alpha^*)$. Let us see the two possible subcases: In the first subcase, $r^k(\alpha^*) = r_{i_0}(\alpha^*)$, which

contradicts the fact that s^k is the first stopping plane at $\alpha^{k+1} > \alpha^*$. In the second subcase, $r^k(\alpha^*) > r_{i_0}(\alpha^*)$. However, considering that r^k is strictly increasing and that r_{i_0} is non increasing, there must exist $\delta_1 > 0$ such that $r^k(\alpha^* - \delta_1) = r_{i_0}(\alpha^* - \delta_1)$, which also contradicts the fact that s^k is the first stopping plane at $\alpha^{k+1} > \alpha^* - \delta_1$.

(b) $f(\alpha^{k+1}) > f(\alpha)$, for all $\alpha \in [a_i, a_{i+1}[$ since $\alpha^{k+1} \geq a_{i+1}$ and $f(\alpha)$ is strictly increasing in $] - \infty, \alpha^*$.

(c) Since $\alpha^k \in [a_i, a_{i+1}[$. ■

Theorem 1 *The radar method converges in a finite number of iterations when it is used solve the OPLC problem (1). The number of iterations is bounded by the number of break points in the interval $[\alpha^0, \alpha^*$.*

Proof: Let $\{\alpha^k\}$ be the sequence generated by the radar method. For each α^k there exists $I^k = [a_i, a_{i+1}[$ such that $\alpha^k \in I^k$. By Lemma 1 b) and c), each interval I^k can be visited at most once since $f(\alpha) < f(\alpha^{k+1}) < f(\alpha^{k+2}) \dots$, for all $\alpha \in I^k$. Therefore, considering that the number of possible intervals is r , the radar method will optimize f in at most r iterations. ■

4.2 Superlinear convergence of the radar method.

In this section we show that the radar method converges superlinearly with order at least two (see for example [Ber95] for a definition of superlinear convergence). To show the superlinear convergence we first need two technical Lemmas.

Lemma 2 *Given two points in \mathbb{R}^2 , (a, y_1) and (b, y_2) with $a \neq b$, and given two slopes m_1 and m_2 , there exists a unique polynomial of third degree*

$$P(x) = c_0 + c_1x + c_2x^2 + c_3x^3$$

such that $P(a) = y_1$, $P(b) = y_2$, $P'(a) = m_1$ and $P'(b) = m_2$.

Proof: Given that $P'(x) = c_1 + 2c_2x + 3c_3x^2$, it is enough to show that there exist a solution for the following linear system whose variables are c_0, c_1, c_2 and c_3 .

$$\begin{aligned} c_0 + ac_1 + a^2c_2 + a^3c_3 &= y_1, \\ c_1 + 2ac_2 + 3a^2c_3 &= m_1, \\ c_1 + 2bc_2 + 3b^2c_3 &= m_2, \\ c_0 + bc_1 + b^2c_2 + b^3c_3 &= y_2. \end{aligned} \tag{3}$$

In matrix form

$$\begin{pmatrix} 1 & a & a^2 & a^3 \\ 0 & 1 & 2a & 3a^2 \\ 0 & 1 & 2b & 3b^2 \\ 1 & b & b^2 & b^3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ m_1 \\ m_2 \\ y_2 \end{pmatrix}$$

We call A^0 the coefficient matrix. To solve this system is equivalent to solve the systems associated to the following equivalent matrices (with the equivalent right-hand-side vectors). The first equivalent matrix A^1 is equal to A^0 except for the 4th row. We obtain the 4th row by subtracting the 4th row of A^0 from the 1st row of A^0 . In compact notation $A_{4*}^1 := A_{1*}^0 - A_{4*}^0$.

$$A^1 = \begin{pmatrix} 1 & a & a^2 & a^3 \\ 0 & 1 & 2a & 3a^2 \\ 0 & 1 & 2b & 3b^2 \\ 0 & a-b & a^2-b^2 & a^3-b^3 \end{pmatrix}$$

A^2 is obtained from A^1 by the operations: $A_{3*}^2 := A_{2*}^1 - A_{3*}^1$ and $A_{4*}^2 := aA_{2*}^1 - bA_{3*}^1 - A_{4*}^1$.

$$A^2 = \begin{pmatrix} 1 & a & a^2 & a^3 \\ 0 & 1 & 2a & 3a^2 \\ 0 & 0 & 2(a-b) & 3(a^2-b^2) \\ 0 & 0 & a^2-b^2 & 2(a^3-b^3) \end{pmatrix}$$

A^3 is obtained from A^2 by the operation: $A_{4*}^3 := (a+b)A_{3*}^2 - 2A_{4*}^2$.

$$A^3 = \begin{pmatrix} 1 & a & a^2 & a^3 \\ 0 & 1 & 2a & 3a^2 \\ 0 & 0 & 2(a-b) & 3(a^2-b^2) \\ 0 & 0 & 0 & Q(a,b) \end{pmatrix}$$

where $Q(a,b) = -(a-b)^3$ as we will show in the last step of the proof. Since by hypothesis we have $a \neq b$ then $\det A^3 = -2(a-b)^4 \neq 0$. Therefore, the linear system (3) has a unique solution and $P(x)$ exists.

To complete the proof, let us see that $Q(a,b) = -(a-b)^3$. By the definition $A_{4*}^3 := (a+b)A_{3*}^2 - 2A_{4*}^2$ we have that

$$Q(a,b) := 3(a+b)(a^2-b^2) - 4(a^3-b^3).$$

Finally, by considering the two basic algebraic properties

$$a^2 - b^2 = (a+b)(a-b) \quad \text{and} \quad a^3 - b^3 = (a^2 + ab + b^2)(a-b),$$

we can write:

$$\begin{aligned} Q(a,b) &= 3(a+b)^2(a-b) - 4(a^2 + ab + b^2)(a-b) \\ &= [3(a+b)^2 - 4(a^2 + ab + b^2)](a-b) \\ &= -[a^2 - 2ab + b^2](a-b) \\ &= -(a-b)^3. \end{aligned}$$

■

Lemma 3 *Let us consider the set of points $\{(a_i, y_i) \in \mathbb{R}^2\}_{i \in K}$ where $K = \{0, \dots, n_k\}$ and $a_i < a_{i+1}$ for all $i \in K \setminus \{n_k\}$. Let us also consider the set of slopes $\{m_i \in \mathbb{R}\}_{i \in K}$. Then, there exists a smooth function $Q(x)$ such that $Q(a_i) = y_i$ and $Q'(a_i) = m_i$ for all $i \in K$.*

Proof: Let us consider the interval $I = [a_0, a_{n_k}]$ and the covering

$$I_1 \cup I_2 \cup \dots \cup I_{n_k} = [a_0, a_1] \cup [a_1, a_2] \cup \dots \cup [a_{n_k-1}, a_{n_k}].$$

By Lemma 2, for each interval $I_i, i \in K$, there exists a unique third degree polynomial, say $P_i(x)$, such that $P_i(a_{i-1}) = y_{i-1}$, $P_i(a_i) = y_i$, $P'_i(a_{i-1}) = m_{i-1}$ and $P'_i(a_i) = m_i$. To finish the proof it is enough to define

$$Q(x) := \begin{cases} P_1(x) & \text{if } x \in I_1, \\ \vdots & \\ P_{n_k}(x) & \text{if } x \in I_{n_k}. \end{cases}$$

■

Now we define two auxiliary functions, g and h , that will be used to proof the superlinear convergence of the radar method. g will be defined as a simplified version of the main function f , and h will be defined as a smooth version of g . The idea of the proof is simple: The superlinear convergence of Newton's method when solving the equation $h(x) = f(\alpha^*)$, will induce superlinear convergence of the radar method when maximizing g and f .

First, let us define g as the lower envelope of a set of lines. This set contains the strictly positive lines and the horizontal line that passes through the point $(\alpha^*, f(\alpha^*))$. To be more precise

$$g(\alpha) := \min (\{r_j(\alpha) \mid j \in J_a^+\} \cup \{r^*(\alpha)\}),$$

where we have used the constant function $r^*(\alpha) := f(\alpha^*)$ for all α . It is easy to see that

$$g(\alpha) = \begin{cases} f(\alpha) & \text{if } \alpha \leq \alpha^* \\ f(\alpha^*) & \text{if } \alpha > \alpha^*. \end{cases}$$

Furthermore, to ease the notation in what follows, we define

$$y^k := g(\beta^k) = r^k(\beta^k) \quad \text{and} \quad y^* := g(\beta^*) = r^*(\beta^*).$$

Secondly, let us define h as a smooth approximation to the lower envelope of the strictly positive lines. To be more precise, to define h we consider the finite sequence of points generated by the radar method to maximize the auxiliary function g , that we denote by $\{(\beta^k, y^k) \in \mathbb{R}^2\}_{k \in K}$, with $K := \{0, \dots, n_k\}$. We also consider the associated sequence of *steepest* active lines $\{r^k\}_{k \in K}$, with $r^k(\beta) = m^k \beta + n^k$. By Lemma 3, there exist a smooth function, say $h(x)$, such that $h(\beta^k) = y^k$ and $h'(\beta^k) = m^k$ for all $k \in K$.

Proposition 1 *Let us consider $\{x^k \mid k \geq 0\}$, the sequence of iterates generated by Newton's method to solve the equation $h(x) - y^* = 0$. Let us also consider $\{\beta^k\}_{k \in K}$ with $K = \{0, \dots, n_k\}$, the sequence of iterates generated by the radar method to maximize $g(\alpha)$. If $x^0 = \beta^0$ then:*

1. *Newton's sequence and the radar sequence are identical for all $k \in K$.*
2. *Newton's sequence converges at iteration n_k .*
3. *The radar sequence converges superlinearly with order at least two.*

Proof: 1) Let us define $\tilde{h}(x) := h(x) - y^*$. In order to solve the equation $\tilde{h}(x) = 0$, Newton's iterates are computed as

$$x^{k+1} = x^k - \frac{\tilde{h}(x^k)}{\tilde{h}'(x^k)} = x^k - \frac{h(x^k) - y^*}{h'(x^k)},$$

from where

$$x^{k+1} = x^k + \frac{y^* - y^k}{m^k}. \quad (4)$$

On the other hand, in the radar method, the slope m^k can be expressed as

$$m^k = \frac{y^* - y^k}{\beta^{k+1} - \beta^k}$$

from where

$$\beta^{k+1} = \beta^k + \frac{y^* - y^k}{m^k}. \quad (5)$$

By hypothesis we have $x^0 = \beta^0$, therefore, by equations (4-5) and applying induction we will have $x^k = \beta^k$ for all $k \in K$.

2) The optimal iterate for the radar method is β^{n_k} what means

$$g(\beta^0) < g(\beta^1) < \dots < g(\beta^{n_k}) = y^*$$

\Updownarrow

$$y^0 < y^1 < \dots < y^{n_k} = y^*$$

\Updownarrow

$$h(x^0) < h(x^1) < \dots < h(x^{n_k}) = y^*$$

as we wanted to prove.

3) It is well known that when Newton's method converges, as in this case, it converges superlinearly with order at least two [Ber95]. By the first statement of this theorem, the radar method inherits the convergence rate of Newton's method. ■

So far we have considered $\{\beta^k\}_{k \in K}$, the sequence generated by the radar method to maximize the auxiliary function g . Let us consider now the sequence generated

by the radar method to maximize the main function f , that we denote $\{\alpha^k\}_{k \in L}$ with $L := \{1, \dots, n_l\}$. The objective of the following three lemmas is to show that $\beta^k \leq \alpha_k$ for all $k \in K \cap L$. This will allow us to prove the superlinear convergence of the radar method at the end of this section.

Lemma 4 *If $\beta^k = \alpha^k$ then $\beta^{k+1} \leq \alpha^{k+1}$ for all $k \in K \cap L$*

Proof: (Proof by contradiction) If $\alpha^{k+1} < \beta^{k+1}$, then, considering that r^k is strictly increasing, we have

$$r^k(\alpha^{k+1}) < r^k(\beta^{k+1}) = y^*. \quad (6)$$

Now, let us consider the first non positive line s^k that determines α^{k+1} by the equation

$$r^k(\alpha^{k+1}) = s^k(\alpha^{k+1}) \quad (7)$$

From (6) and (7), we have

$$s^k(\alpha^{k+1}) < y^* \quad (8)$$

Furthermore, by definition of $f(\alpha)$

$$s^k(\alpha^*) \geq y^* \quad (9)$$

It turns out that $\alpha^{k+1} < \alpha^*$, because by Lemma 1 $\alpha^{k+1} \leq \alpha^*$ and from (8) and (9), we know that $\alpha^{k+1} \neq \alpha^*$. Putting all together, we have $\alpha^{k+1} < \alpha^*$ and $s^k(\alpha^{k+1}) < y^* \leq s^k(\alpha^*)$, which implies that s^k has strictly positive slope. This contradicts the fact that s^k is a non positive line. ■

We now consider γ^{k+1} , the iterate generated by the radar method applied to maximize $g(\alpha)$ when starting at any arbitrary point $\gamma^k \in [0, \alpha^*]$.

Lemma 5 *If $\beta^k \leq \gamma^k$ then $\beta^{k+1} \leq \gamma^{k+1}$ for all $k \in K$.*

Proof: Let us consider r_β and r_γ , the active lines at β^k and γ^k , respectively. We encounter two cases.

a) First case: r_β and r_γ are the same line. In this case by the definition of the radar method, $\beta^{k+1} = \gamma^{k+1}$.

b) Second case: r_β and r_γ are different lines. If we denote by, m_β and m_γ the slopes of r_β and r_γ , respectively, we will have $m_\beta > m_\gamma$, since $\beta^k < \gamma^k$.

Let us denote by y_β^k and y_γ^k the scalars $r_\beta(\beta^k)$ and $r_\gamma(\gamma^k)$, respectively. By the definition of g , the only possible stopping line is r^* and therefore we have that β^{k+1} , is determined by the equation $r_\beta(\beta^{k+1}) = r^*(\beta^{k+1})$. Since $r^*(\beta^{k+1}) = y^*$, we have $r_\beta(\beta^{k+1}) = y^*$. For this reason the points (β^k, y_β^k) and (β^{k+1}, y^*) are in the graph of the line r_β , and can be used to determine its slope m_β :

$$m_\beta = \frac{r_\beta(\beta^{k+1}) - r_\beta(\beta^k)}{\beta^{k+1} - \beta^k} = \frac{y^* - y_\beta^k}{\beta^{k+1} - \beta^k},$$

from where,

$$\begin{aligned}
\beta^{k+1} &= \beta^k + \frac{y^* - y_\beta^k}{m_\beta} \\
&\pm y_\gamma^k / m_\beta \\
&= \beta^k + \frac{y_\gamma^k - y_\beta^k}{m_\beta} + \frac{y^* - y_\gamma^k}{m_\beta}.
\end{aligned} \tag{10}$$

Let us see now that,

$$\beta^k + \frac{y_\gamma^k - y_\beta^k}{m_\beta} \leq \gamma^k. \tag{11}$$

Otherwise,

$$y_\gamma^k > y_\beta^k + m_\beta(\gamma^k - \beta^k),$$

which contradicts the fact that

$$y_\beta^k + m_\beta(\gamma^k - \beta^k) \geq g(\gamma^k) = y_\gamma^k.$$

From (10) and (11):

$$\begin{aligned}
\beta^{k+1} &\leq \gamma^k + \frac{y^* - y_\gamma^k}{m_\beta} \\
&< \gamma^k + \frac{y^* - y_\gamma^k}{m_\gamma} \quad (\text{since } m_\beta > m_\gamma) \\
&= \gamma^{k+1}.
\end{aligned}$$

■

Lemma 6 *If $\beta^0 = \alpha^0$ then $\beta^k \leq \alpha^k$ for all $k \in K \cap L$.*

Proof: (Proof by induction)

If $\beta^0 = \alpha^0$ then $\beta^1 \leq \alpha^1$ (Lemma 4).

Analogously, if $\gamma^1 := \alpha^1$ then $\gamma^2 \leq \alpha^2$ (Lemma 4).

Given that $\beta^1 \leq \alpha^1 = \gamma^1$ then $\beta^2 \leq \gamma^2$ (Lemma 5).

Thus, $\beta^2 \leq \alpha^2$.

To complete our proof, assume that $\beta^k \leq \alpha^k$ and let us see that $\beta^{k+1} \leq \alpha^{k+1}$.

If $\gamma^k := \alpha^k$ then $\gamma^{k+1} \leq \alpha^{k+1}$ (Lemma 4).

Given that $\beta^k \leq \alpha^k = \gamma^k$ then $\beta^{k+1} \leq \gamma^{k+1}$ (Lemma 5).

Thus, $\beta^{k+1} \leq \alpha^{k+1}$. ■

Theorem 2 *The radar method, used to solve the OPLC problem (1), converges in a finite number of iterations. Furthermore, the convergence is superlinear with order at least two.*

Proof: Without loss of generality, we can assume that $\alpha^0 = \beta^0 = 0$. By Lemma 1, we have that $\alpha^k \in [0, \alpha^*]$ for all $k \in L$. Applying Lemma 6 we can further reduce this interval such that $\alpha^k \in [\beta^k, \alpha^*]$ for all $k \in K \cap L$. The proof of the first statement is completed by considering that $\beta^{n_k} = \alpha^*$, which implies $\alpha^k = \alpha^*$ for some $k \in L$. This also shows that the rate of convergence of $\{\alpha^k\}_{k \in L}$ is at least the rate of convergence of the sequence $\{\beta^k\}_{k \in K}$, which by Proposition 1 is superlinear with order at least two. ■

5 Numerical test

The objective of this example is to illustrate the performance of the radar method applied to maximize an one-dimensional piecewise linear concave (OLPC) function either with a few or thousands of break points. We compare the performances of the radar, next-break-point and golden section methods.

All the programs have been written in Matlab 7.0 [HH00]. A difference between the radar and the golden section method is that in the first we compute an *exact* solution and in the second we compute an *interval of uncertainty* that contains the solution. For this test, we stop the golden section iterations when the length of this interval is less than 10^{-6} . All programs have been run on a PC (Pentium-IV, 3 GHz, with 3 GB of RAM memory) under the Windows XP operating system.

For the studied problem, up to our knowledge, there are no public instance data bases. We have generated two different sets of instances.

5.1 Instances with a small number of break points

We solve ten randomly generated OPLC instances. Each instance is generated by taking a collection of N_l random lines (from 10 to 500,000 lines). Each line r_j is defined by a point (a_j, b_j) and a slope m_j , **with a_j and b_j uniformly distributed in $[0, 1]$ and m_j uniformly distributed in $[-1, 1]$ ($j = 1, \dots, N_l$).**

The test results are displayed in two tables. In Table 1 we report the problem description and the optima obtained. In Table 2 we report the number of iterations and CPU time for the three methods. We observe that the number of break points in the interval $[0, \alpha^*]$ is 1 or 2 (the same as the number of next-break-point iterations). As expected, the golden section method does not take advantage of this small amount of break points and on average performs 38 iterations. The performance of the radar and next-break-point method is very similar. For the cases with a large number of lines, on average, the radar method needs a quarter of the time needed by the golden section method.

Table 1: Problem description.

Instance	Number of lines	Optimum	
		α^*	$f(\alpha^*)$
oplc01	10	0.8957	0.1634
oplc02	50	0.5060	0.0336
oplc03	100	0.6066	-0.2432
oplc04	500	0.4598	-0.2238
oplc05	1000	0.5533	-0.3458
oplc06	5000	0.4968	-0.3658
oplc07	10000	0.4557	-0.4008
oplc08	50000	0.4918	-0.4480
oplc09	100000	0.5009	-0.4536
oplc10	500000	0.4973	-0.4805

Table 2: Influence of the number of break points. NBP stands for ‘next-break-point’ method. CPU Time under 0.01 seconds are not reported. (*) Average CPU time only accounts for cases 06 to 10.

Instance	Iterations			CPU time (sec.)		
	Radar	NBP	Golden	Radar	NBP	Golden
oplc01	3	3	39	-	-	-
oplc02	1	1	40	-	-	-
oplc03	1	1	36	-	-	-
oplc04	1	1	38	-	-	-
oplc05	1	1	39	-	-	-
oplc06	1	1	39	-	-	0.03
oplc07	2	2	39	0.03	-	0.02
oplc08	1	1	39	0.03	-	0.11
oplc09	1	1	38	0.05	0.03	0.23
oplc10	2	2	37	0.30	0.20	1.19
Average (*)	1	1	38	0.08	0.05	0.32

Table 3: Problem description.

Instance	Number			Optimum	
	of lines	Δb	Δm	α^*	$f(\alpha^*)$
oplc11	10	0.5	0.05	56.6661	142.5815
oplc12	100	0.5	0.05	46.6278	142.6647
oplc13	1000	0.5	0.05	52.1655	126.0272
oplc14	10000	0.5	0.05	49.3297	125.3751
oplc15	100000	0.5	0.05	49.9642	125.0443
oplc16	10	0	0	48.0761	125.5347
oplc17	100	0	0	49.6130	125.0459
oplc18	1000	0	0	50.0001	125.0002
oplc19	10000	0	0	50.0037	125.0000
oplc20	100000	0	0	50.0000	125.0000

5.2 Instances with a large number of break points

The objective of this section is to maximize OPLC instances with a number of break points that ranges from small to very large, in contrast with the instances of the first set. We solve ten randomly generated OPLC instances derived from the quadratic function $q(\alpha) = -(1/20)\alpha^2 + 5\alpha$, with $q(0) = q(100) = 0$ and $q(\alpha^*) = q(50) = 125$. Each OPLC instance is generated by taking a random collection of N_l lines from the family of lines that lay above the epigraph of q (from 10 to 100,000 lines). **Given two positive tolerances Δb and Δm** , each line r_j is defined by a point (a_j, b_j) and a slope m_j , with a_j, b_j and m_j uniformly distributed in $[0, 100]$, $[q(a_j), (1 + \Delta b)q(a_j)]$ and $[(1 - \Delta m)q'(a_j), (1 + \Delta m)q'(a_j)]$ respectively ($j = 1, \dots, N_l$). **We give the values of Δb and Δm in Table 3.**

The test results are displayed in two tables. In Table 3 we report the problem description and the optima obtained. Note that when Δb and Δm are 0 (instances oplc16-oplc20 in Table 3), the lines are tangent to the graph of q . In this case the number of break points in the interval $[0, \alpha^*]$ is approximately $N_l/2$ (see column ‘Iterations NBP’ in Table 4 for instances oplc16-oplc20).

In Table 4, we report the performance in terms of number of iterations and CPU time. Obviously one would not use the next-break-point method for the problems with a very large number of break points (e.g. instances oplc19-oplc20). We have used it in order to count the number of break points between 0 and the optimal point. **We observe that** the number of radar iterations is independent from the number of break points. **Furthermore, the very low number of radar iterations is explained by its superlinear rate of convergence.** For the cases with a large number of lines, on average, the radar method need half of the time needed by the golden section method.

Table 4: Influence of the number of break points. NBP stands for ‘next-break-point’ method. CPU Time under 0.01 seconds are not reported. (*) Average CPU time only accounts for cases 14, 15 and 18 to 20.

Instance	Iterations			CPU time (sec.)		
	Radar	NBP	Golden	Radar	NBP	Golden
oplc11	2	2	41	-	-	-
oplc12	3	5	38	-	-	-
oplc13	4	10	32	-	-	-
oplc14	5	23	37	0.02	0.05	0.02
oplc15	6	44	38	0.06	0.75	0.25
oplc16	3	3	39	-	-	-
oplc17	6	51	36	-	-	-
oplc18	9	526	38	-	0.09	-
oplc19	13	5063	38	0.03	6.81	0.05
oplc20	14	49873	35	0.11	849.31	0.23
Average (*)	7	5560	37	0.04	177.56	0.11

6 Conclusions

In this paper we have improved the so-called radar method, a method intended to optimize one-dimensional piecewise linear concave (OPLC) functions, defined as the lower envelope of a finite set of lines. The method is thus well suited for exact OPLC line search, that arises for example in the framework of the Kelley cutting plane method.

From a theoretical point of view, we have established finite and superlinear convergence with order at least two for the radar method. The other advantage of the radar method is that the number of iterations does not depend on the number of break points.

From a practical point of view we have observed that the number of radar iterations is very small and independent of the number of break points, as predicted by the theory. For a large number of break points the performance of the radar method is similar if not better than the golden section method, a representative derivative-free line search method. However, in the case of very few break points, the radar method is clearly superior to this type of methods since the number of radar iterations is bounded by the number of break points.

All in all, the radar method seems the most effective method to maximize an OPLC function. Even if the CPU time per line search is small, typically one performs thousands of these line searches in methods such as the Kelley cutting plane method. Therefore, the total CPU time saving may not be negligible if one uses the radar method. An additional advantage is that the radar method computes an exact

optimizer in contrast with the golden method or similar, which compute an approximated optimizer. In general, exact optimizers are better suited for active set optimization methods.

Acknowledgments: We are thankful to Jean-Philippe Vial, Alain B. Haurie and Daniel S. Zachary for their comments and support at Logilab, HEC, University of Geneva. **We are also thankful to the anonymous referees for their constructive comments and suggestions.**

References

- [Bel07] C. Beltran-Royo. A conjugate Rosen’s gradient projection method with global line search for piecewise linear concave optimization. *European Journal of Operational Research*, in press 2007.
- [Ber95] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, (USA), 1995.
- [BH05] C. Beltran and F. J. Heredia. An effective line search for the subgradient method. *Journal on Optimization Theory and Applications*, 125(1):1–18, 2005.
- [BS79] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming*. John Wiley and Sons, USA, 1979.
- [CC90] A. R. Conn and G. Cornuéjols. A projection method for the uncapacitated facility location problem. *Mathematical programming*, 46:373–398, 1990.
- [HH00] D. J. Higham and N. J. Higham. *MATLAB guide*. SIAM, Philadelphia, Pennsylvania, USA, 2000.
- [HP04] W. W. Hager and S. Park. The gradient projection method with exact line search. *Journal of Global Optimization*, 30(1):103–118, 2004.
- [Kel60] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the SIAM*, 8:703–712, 1960.
- [NBR00] P. Neame, N. Boland, and D. Ralph. An outer approximate subdifferential method for piecewise affine optimization. *Mathematical programming*, Ser. A 87:57–86, 2000.
- [Sho98] N. Z. Shor. *Nondifferentiable optimization and polynomial problems*. Nonconvex optimization and its applications. Kluwer academic publishers, 1998.