Minimization of discontinuous cost functions by smoothing

José Mario Martínez*

September 23, 2001

Abstract

The minimization of functions that include discontinuous costs is considered. Sufficient conditions are introduced under which a smoothing scheme converges to the solution of the original problem. Some examples are presented and the numerical behavior of the proposed approach is commented.

Keywords: Optimization, discontinuous cost, algorithms, Augmented Lagrangians, Box-constrained minimization.

AMS: 90C05, 90C20, 90C25

1 Introduction

Many practical problems require the minimization of functions that involve discontinuous costs. Among them, we can cite process models with discontinuous investment costs and fixed charges [15], continuous review (s, q) inventory systems with constant demand and batch arrivals [6], design of flowsheets for systems that satisfy fixed demands of steam, electricity and mechanical power [9], models for expansion of capacity of telecommunications networks [14], capacity and flow assignment problems [7], optimal plastic design [13] and many others.

*Institute of Mathematics, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This work was supported by PRONEX, FAPESP (grant 2001-04597-4), FINEP, CNPq, FAEP-UNICAMP.

The tools for solving those problems include Monte Carlo simulation [16], disjunctive programming approaches [15], integer order strategies [6], simulated annealing [9], decomposition methods [14], integer programming techniques [7], active-set and projected gradient strategies [3].

In this paper we analyze the solution of a large family of problems with discontinuous cost using smoothing of the cost function. We establish sufficient conditions on the approximation that ensure that the smoothed problems really approximate the original problem.

The smoothed problems are always nonconvex and, frequently, have a lot of local minimizers. For this reason, "local" optimization algorithms (which are guaranteed to find only stationary points) are not good enough for solving them and it is necessary to use some globalization tool. Here we use the simple procedure of running the local algorithms with many different random starting points which, in many cases, is quite efficient. For solving the nonlinear programming problems that arise after smoothing we use Augmented Lagrangians [8] and box-constrained minimization [5].

The general problem is presented in Section 2, where we also prove that global minimizers of the smoothed problems converge to global minimizers of the general problem. In Section 3 we define critical points and we show that stationary points of the smoothed problems converge to critical points of the original problem. In Section 4 we present some examples and in Section 5 we suggest lines for future research.

Throughout the paper $\|\cdot\|$ denotes the Euclidean norm.

2 The problem and its approximation

We assume that $f : I\!R^n \to I\!R$ is continuous, $g_i : I\!R^n \to I\!R$ is continuous for all $i = 1, \ldots, m$ and $\Omega \subset I\!R^n$. Our problem is to find $x \in \Omega$ that minimizes a function that can be discontinuous in a specified way. We consider nondecreasing functions $H_i : I\!R \to I\!R$, $i = 1, \ldots, m$, such that H_i is continuous except at breakpoints α_{ij}, $j \in I_i$. The set I_i can be finite or infinite but the set of breakpoints is discrete, in the sense that

$$\text{Infimum } \{|\alpha_{i\ell} - \alpha_{ij}| \mid \ell, j \in I_i,\ \ell \neq j\} > 0. \tag{1}$$

The side limits $\lim_{t \to \alpha_{ij}^+} H_i(t)$, $\lim_{t \to \alpha_{ij}^-} H_i(t)$ exist for all $j \in I_i$ and, in addition,

$$\lim_{t \to \alpha_{ij}^-} H_i(t) = H_i(\alpha_{ij}) < \lim_{t \to \alpha_{ij}^+} H_i(t) \tag{2}$$

for all $j \in I_i$, $i = 1, \dots, m$.

Our problem is

$$\text{Minimize } f(x) + \sum_{i=1}^{m} H_i[g_i(x)] \quad \text{subject to} \quad x \in \Omega. \tag{3}$$

If Ω is compact, since the objective function of (3) is lower-semicontinuous, a global solution of this problem exists. See, for example, [2].

The cost function H_i will be approximated by a family of continuous nondecreasing functions $H_{ik} : I\!R \to I\!R$. We assume that the approximating functions are such that, for all $\mu > 0$,

$$\lim_{k\to\infty} H_{ik}(t) = H_i(t) \text{ uniformly in } I\!R - \bigcup_{j\in I_i} (\alpha_{ij}, \alpha_{ij} + \mu). \tag{4}$$

Observe that (4) implies that

$$\lim_{k\to\infty} H_{ik}(t) = H_i(t) \text{ for all } t \in I\!R.$$

For each k, we define the approximate problem P_k as

$$\text{Minimize } f(x) + \sum_{i=1}^{m} H_{ik}[g_i(x)] \quad \text{subject to} \quad x \in \Omega. \tag{5}$$

Problem (5) has a continuous objective function. It is not difficult to define a smooth family of approximating functions H_{ik}, therefore (5) can be solved using continuous optimization algorithms. In the following theorem, we prove that the solution of (3) can be approximated by the solutions of (5).

Theorem 2.1. *Assume that for all $k = 0, 1, 2, \dots$, x^k is a solution of (5) and that $x^* \in \Omega$ is a cluster point of $\{x^k\}$. Then, x^* is a solution of (3).*

Proof. Taking an appropriate subsequence and relabeling, we can assume that $x^k \to x^*$. Since all the functions g_i are continuous, this implies that

$$g_i(x^k) \to g_i(x^*) \text{ for all } i = 1, \dots, m. \tag{6}$$

Suppose that $i \in \{1, \dots, m\}$ is such that

$$g_i(x^*) \notin \{\alpha_{ij}, j \in I_i\}.$$

Define

$$\mu = \frac{1}{2} \min\{|g_i(x^*) - \alpha_{ij}|, j \in I_i\}.$$

Assumption (1) guarantees that $\mu > 0$. Given $\varepsilon > 0$, by (4), there exists k_0 such that, for all $k \geq k_0$,

$$|H_{ik}(t) - H_i(t)| \leq \varepsilon/2 \text{ for all } t \in I\!R - \bigcup_{j \in I_i} (\alpha_{ij}, \alpha_{ij} + \mu). \tag{7}$$

By the continuity of g_i, there exists k_1 such that

$$g_i(x^k) \in I\!R - \bigcup_{j \in I_i} (\alpha_{ij}, \alpha_{ij} + \mu) \text{ for all } k \geq k_1. \tag{8}$$

Then, by (7),

$$|H_{ik}[g_i(x^k)] - H_i[g_i(x^k)]| \leq \varepsilon/2 \text{ for all } k \geq \max\{k_0, k_1\}. \tag{9}$$

Since $g_i(x^k) \to g_i(x^*)$ and $g_i(x^*) \notin \{\alpha_{ij}, j \in I_i\}$, we have that H_i is continuous at $g_i(x^*)$. Therefore, there exists k_2 such that

$$|H_i[g_i(x^k)] - H_i[g_i(x^*)]| \leq \varepsilon/2 \text{ for all } k \geq k_2. \tag{10}$$

By (9) and (10) we have that, for $k \geq \max\{k_0, k_1, k_2\}$,

$$|H_{ik}[g_i(x^k)] - H_i[g_i(x^*)]| \leq |H_{ik}[g_i(x^k)] - H_i[g_i(x^k)]| + |H_i[g_i(x^k)] - H_i[g_i(x^*)]| \leq \varepsilon.$$

Therefore we proved that, if $g_i(x^*) \notin \{\alpha_{ij}, j \in I_i\}$,

$$\lim_{k \to \infty} H_{ik}[g_i(x^k)] = H_i[g_i(x^*)]. \tag{11}$$

Suppose now that $g_i(x^*) = \alpha_j$ for some $j \in I_i$. The sequence $\{H_{ik}[g_i(x^k)]\}$ can be decomposed in two subsequences, one of them corresponding to the indices k such that $g_i(x^k) \leq \alpha_{ij}$ (left subsequence) and the other corresponding to the indices k such that $g_i(x^k) > \alpha_{ij}$ (right subsequence). Perhaps, one of them is finite. Suppose that the left subsequence is infinite. Given an arbitrary $\varepsilon > 0$, take k_0 as before. As in the previous choice of k_1, there exists k_3 such that for all the indices k of the left subsequence such that $k \geq k_3$ we have that (8) holds. Therefore, for those indices and $k \geq k_3$, we have that

$$|H_{ik}[g_i(x^k)] - H_i[g_i(x^k)]| \leq \varepsilon/2.$$

By the left-continuity of H_i at α_{ij}, there exists k_4 such that for all the indices k of the left subsequence such that $k \geq k_4$,

$$|H_i[g_i(x^k)] - H_i[g_i(x^*)]| \leq \varepsilon/2.$$

Combining the two previous inequalities, we obtain that for all the indices of the left subsequence such that $k \geq \max\{k_0, k_3, k_4\}$,

$$|H_{ik}[g_i(x^k)] - H_i[g_i(x^*)]| \leq \varepsilon.$$

So, for the left subsequence,

$$\lim_{k\to\infty} H_{ik}[g_i(x^k)] = H_i[g_i(x^*)]|. \tag{12}$$

For the right subsequence, since $g_i(x^k) \geq g_i(x^*)$ we have that, since H_{ik} is nondecreasing,

$$H_{ik}[g_i(x^k)] \geq H_{ik}[g_i(x^*)]$$

Therefore,

$$\liminf_{k\to\infty} H_{ik}[g_i(x^k)] \geq \liminf_{k\to\infty} H_{ik}[g_i(x^*)] = \lim_{k\to\infty} H_{ik}[g_i(x^*)] = H_i[g_i(x^*)]. \tag{13}$$

By (12) and (13), we have that, when $g_i(x^*) = \alpha_{ij}$ for some $j \in I_i$,

$$\liminf_{k\to\infty} H_{ik}[g_i(x^k)] \geq H_i[g_i(x^*)]. \tag{14}$$

By (11), (14) and the continuity of f, we have that

$$\liminf_{k\to\infty} f(x^k) + \sum_{i=1}^{m} H_{ik}[g_i(x^k)] \geq f(x^*) + \sum_{i=1}^{m} H_i[g_i(x^*)]. \tag{15}$$

Let us assume now, by contradiction, that x^* is not a solution of problem (3). Therefore, there exists $\bar{x} \in \Omega$ and $c > 0$ such that

$$f(\bar{x}) + \sum_{i=1}^{m} H_i[g_i(\bar{x})] < f(x^*) + \sum_{i=1}^{m} H_i[g_i(x^*)] - c. \tag{16}$$

But, for all $i = 1, \ldots, m$,

$$\lim_{k\to\infty} H_{ik}[g_i(\bar{x})] = H_i[g_i(\bar{x})],$$

so,

$$\lim_{k\to\infty} f(\bar{x}) + \sum_{i=1}^{m} H_{ik}[g_i(\bar{x})] = f(\bar{x}) + \sum_{i=1}^{m} H_i[g_i(\bar{x})].$$

Therefore, by (16), there exists k_5 such that

$$f(\bar{x}) + \sum_{i=1}^{m} H_{ik}[g_i(\bar{x})] < f(x^*) + \sum_{i=1}^{m} H_i[g_i(x^*)] - c/2$$

for all $k \geq k_5$. Thus, by (15), for $k \geq k_5$ we have that

$$f(\bar{x}) + \sum_{i=1}^{m} H_{ik}[g_i(\bar{x})] < \liminf_{k\to\infty} f(x^k) + \sum_{i=1}^{m} H_{ik}[g_i(x^k)] - c/2.$$

Since x^k is a solution of (5), this implies that

$$f(\bar{x}) + \sum_{i=1}^{m} H_{ik}[g_i(\bar{x})] < \liminf_{k\to\infty} f(\bar{x}) + \sum_{i=1}^{m} H_{ik}[g_i(\bar{x})] - c/2$$

for all $k \geq k_5$. This is a contradiction that comes from the wrong assumption that x^* is not a solution of (3). Therefore, the desired result is proved. $\Box$

3 Convergence to critical points

In optimization problems, the set of "critical" points includes the set of local minimizers. For example, in smooth unconstrained optimization, the critical points are the ones where the gradient vanishes and in constrained optimization critical points may be defined as the points that satisfy Fritz-John conditions. Usually, critical points are defined in terms of derivative information obtained at the point. However, in a recent paper [10] it was shown that sharper sets of critical points can be obtained as limits of sequences that satisfy gradient projection conditions. In this section, we define critical points of the problem (3) when Ω is given by

$$c_i(x) \leq 0, \; i = 1, \ldots, s, \;\; h_j(x) = 0, \; j = 1, \ldots, q \tag{17}$$

and the functions f, c_i, h_j admit continuous first derivatives. Let us consider a family of continuously differentiable approximations H_{ik}. Define, for all $x \in I\!R^n$,

$$F(x) = f(x) + \sum_{i=1}^{m} H_i[g_i(x)]. \tag{18}$$

$$F_k(x) = f(x) + \sum_{i=1}^{m} H_{ik}[g_i(x)]. \tag{19}$$

We say that y^k is an ε-Fritz-John point of (5) if there exist multipliers $\mu_{k,1}, \ldots, \mu_{k,s}, \lambda_{k,1}, \ldots, \lambda_{k,q}$ such that

$$\sum_{i=1}^{s} \mu_{k,i} + \sum_{j=1}^{q} |\lambda_{k,j}| > 0, \tag{20}$$

$$\mu_{k,i} \geq 0 \text{ for all } i = 1, \ldots, s, \tag{21}$$

$$c_i(y^k) \leq 0,\ h_j(y^k) = 0 \text{ for all } i = 1, \ldots, s,\ j = 1, \ldots, q, \tag{22}$$

$$\mu_{k,i} c_i(y^k) = 0 \text{ for all } i = 1, \ldots, s \tag{23}$$

and

$$\|\nabla F_k(y^k) + \sum_{i=1}^{s} \mu_{k,i} \nabla c_i(y^k) + \sum_{j=1}^{q} \lambda_{k,j} \nabla h_j(y^k)\| \leq \varepsilon. \tag{24}$$

We say that x^* is a critical point of (3) if there exist $\varepsilon_k \to 0$ and $y^k \to x^*$ such that y^k is a ε_k-Fritz-John point of (5).

Theorem 3.1. *The local minimizers of (3) are critical points.*

Proof. The proof follows the ideas used in Chapter 3 of [2] for proving Fritz-John conditions of nonlinear programming and in [10] for proving a sequential optimality condition for constrained minimization. Assume that x^* is a local minimizer of (3). Therefore, there exists $\gamma > 0$ such that for all $x \in \Omega$ such that $\|x - x^*\| \leq \gamma$, we have that $F(x) \geq F(x^*)$. Let us define, for all $x \in I\!R^n$,

$$\bar{F}(x) = F(x) + \|x - x^*\|^2$$

and

$$\bar{F}_k(x) = F_k(x) + \|x - x^*\|^2.$$

Clearly, x^* is the unique global minimizer of $F(x)$ in the set $\bar{\Omega} \equiv \{x \in \Omega \mid \|x - x^*\| \leq \gamma\}$. Let y^k be a global minimizer of $\bar{F}_k(x)$ subject to $x \in \bar{\Omega}$. By Theorem 2.1, since x^* is unique, we have that

$$\lim_{k\to\infty} y^k = x^*.$$

Therefore, for k large enough, $\|y^k - x^*\| < \gamma$. Since the constraint $\|y^k - x^*\| \leq \gamma$ is not active at y^k, this point satisfy the Fritz-John conditions of

$$\text{Minimize } \bar{F}_k(x) \quad \text{subject to } x \in \Omega.$$

This implies (see [2]) that (20)–(23) hold at y^k. Moreover,

$$\|\nabla F_k(y^k) + 2(y^k - x^*) + \sum_{i=1}^{s} \mu_{k,i} \nabla c_i(y^k) + \sum_{j=1}^{q} \lambda_{k,j} \nabla h_j(y^k)\| = 0. \quad (25)$$

Therefore,

$$\|\nabla F_k(y^k) + \sum_{i=1}^{s} \mu_{k,i} \nabla c_i(y^k) + \sum_{j=1}^{q} \lambda_{k,j} \nabla h_j(y^k)\| = 2\|y^k - x^*\|.$$

So, taking $\varepsilon_k = 2\|y^k - x^*\|$, we complete the proof that x^* is a critical point. □

Most practical algorithms for solving (5) find critical (stationary, Fritz-John) points of the problem. So, it is useful to characterize the limits of those points.

Theorem 3.2. *Assume that, for all $k = 0, 1, 2, \ldots$, y^k is a Fritz-John point of (5). Then, every limit point of $\{y^k\}$ is a critical point of (3).*

Proof. Observe that (20)–(24) hold with $\varepsilon_k = 0$. □

4 Examples

The experiments reported here were run in a SPARCstation Sun Ultra 1, with an UltraSPARC 64 bits processor, 167-MHz clock and 128-MBytes of RAM memory. All the codes considered in this paper were written in double precision Fortran and were compiled with the f77 compiler (SC 1.0 Fortran v1.4) using the optimization option -O. The codes corresponding to the examples below are available in [17]. The main tools for solving the approximate smooth problems are the augmented Lagrangian algorithm described in [8] and the box-constrained optimization algorithm given in [5]. See the codes EASY! and BOX9903 of [17].

4.1 Linear programming with starting cost

We address the problem

$$\text{Minimize } \sum_{i=1}^{n} H_i(x_i) \text{ s.t. } Ax = b,\ x \geq 0, \tag{26}$$

where $x = (x_1, \ldots, x_n)$ and

$$H_i(t) = \delta_i + c_i t \text{ if } t > 0,\ \ 0 \text{ otherwise} \tag{27}$$

for all $i = 1, \ldots, n$. We assume that $\delta_i > 0$, $c_i \geq 0$. In this example $I_i = \{0\}$ for all $i = 1, \ldots, n$. In our experiments we defined $\delta_i = i^2$. The problem belongs to the class defined in the previous section with $m = n$,

$$\Omega = \{x \in I\!R^n \mid Ax = b,\ x \geq 0\}$$

and

$$g_i(x) = x_i, i = 1, \ldots, n.$$

The approximating functions H_{ik} are defined as follows:

$$H_{ik}(t) = 0 \text{ if } t \leq 0;$$

$$H_{ik}(t) = c_i t + \frac{\delta_i k t^2}{1 + kt^2} \text{ if } t > 0.$$

Therefore, the approximating problems are

$$\text{Minimize } \sum_{i=1}^{n} H_{ik}(x_i) \text{ s.t. } Ax = b,\ x \geq 0. \tag{28}$$

We use [8] for solving (28) for increasing values of k and we stop when k is large enough. The solution for some value of k is used as initial approximation for the following approximate problem. We used $k = 10^1, 10^2, \ldots, 10^5$. In order to take proper advantage of the solution of one approximating problem as initial point for the next one, in the approximate problems corresponding to $k = 10^\ell, \ell = 1, \ldots, 4$ we allowed only one outer iteration of the algorithm [8] and we used as penalty parameter $\rho = 10$ times the penalty parameter used in the previous approximate problem. In this way the degree of feasibility conquered at some problem is not lost at the beginning of the following one. The penalty parameter used at the first approximate problem was 10. The final solution was "rounded" setting $x_i = 0$ whenever $|x_i| < 10^{-4}$.

The coefficients of A, as well as those of c and the initial approximation for the first approximate problem were taken randomly between 0 and 10. The independent vector b was computed as $A\tilde{x}$, where $\tilde{x}$ is a random $(0, 10)$ vector. The sparsity pattern of A was also taken randomly. Given a probability p, the entrance (i, j) of A is null with probability $1 - p$. The probability p is called "density" of A. The code (STARTCOS) is available in [17] and is, essentially, an adaptation of EASY! for this problem.

Both (26) and (28) are global optimization problems. There is no guarantee that the Augmented Lagrangian method is going to find a global solution of (28) starting from an arbitrary initial point. Therefore, in order to find practical solutions of (26) we executed the procedure sketched above starting from 20 different initial points. Before giving some statistics concerning these experiments, let us comment some qualitative aspects the numerical results.

(a) The solution that comes from the first call of the Augmented Lagrangian method $k = 10, \rho = 10$, is always close to the final solution $(k = 10^5)$. For example, in a problem with 10 variables and 5 linear equality constraints the sup norm of the difference between those points is always smaller than 1 %.

(b) The final value of the objective function of (28) is always very close to the objective function (26). For example, in the problem commented above, the objective function of (28) at the solution was 257.933275 whereas the objective function of (26) at the same point was 257.933298.

(c) Different final values of the objective function correspond, in general, to different solutions. In the problem mentioned above, the best functional value was attained 5 times, the second best value was attained 14 times and a third value was attained in one run.

In the table below we give the results for 20 runs of a problem with 200 variables and 100 constraints ($A \in I\!R^{100 \times 200}$) with a density of 0.2. We report:

(i) REAL OBJ: Final objective function of (26);

(ii) APPROX OBJ: Final objective function of (28);

(iii) FIRST OUTER: Objective function of (26) at the end of the first Augmented Lagrangian outer iteration;

(iii) CPU: Computer time (seconds).

We observe that this (not very small) problem is harder than the previously commented 5×10 problem, in the sense that the minimum objective function value was obtained only once. Therefore, one cannot be reasonable confident that a global minimizer has been obtained.

Initial point	REAL OBJ	APPROX OBJ	FIRST OUTER	CPU
1	1.13 E6	1.13E6	1.13E6	185
2	1.21E6	1.21E6	1.20E6	200
3	1.12E6	1.12E6	1.12E6	174
4	0.958E6	0.958E6	0.952E6	138
5	0.999E6	0.999E6	0.988E6	197
6	1.03E6	1.03E6	1.02E6	184
7	1.19E6	1.19E6	1.09E6	183
8	1.26E6	1.26E6	0.933E6	189
9	1.03E6	1.03E6	1.03E6	202
10	1.18E6	1.18E6	1.17E6	170
11	1.03E6	1.03E6	1.02E6	211
12	1.18E6	1.18E6	1.18E6	217
13	1.21E6	1.21E6	1.20E6	154
14	1.33E6	1.33E6	1.33E6	203
15	1.05E6	1.05E6	1.05E6	229
16	1.17E6	1.17E6	1.16E6	146
17	1.09E6	1.09E6	1.08E6	201
18	1.19E6	1.19E6	1.20E6	150
19	0.906E6	0.906E6	0.893E6	205
20	0.995E6	0.995E6	0.916E6	143

Table 1. 100 rows, 200 columns.

4.2 Kissing problems

The *Kissing Number Problem* seeks to determine the maximum number $\mathcal{K}_{ndim}$ of nonoverlapping spheres of given radius in $\mathbb{R}^{ndim}$ that can simultaneously touch (kiss) a central sphere of same radius. Equivalently, one wants to maximize the number of points in the unitary sphere such that the distance between any pair of them is greater than or equal to 1.

In Table 2 we give the known values/bounds of $\mathcal{K}_{ndim}$ mentioned in [4].

n	$\mathcal{K}_{ndim}$
1	2
2	6
3	12
4	24–25
5	40–46
6	72–82
7	126–140
8	240
9	306–380
10	500–595
11	582–915
12	840–1416

Table 2. Known values and bounds for $\mathcal{K}_{ndim}$.

For points $P_1, \ldots, P_{npun} \in I\!R^{ndim}$, consider the optimization problem

$$\text{Minimize} \sum_{i=1}^{npun-1} \sum_{j=i+1}^{npun} H[1 - \|P_i/\|P_i\| - P_j/\|P_j\|\|], \tag{29}$$

where

$$H(t) = 0 \text{ if } t \leq 0, \quad H(t) = 1 \text{ if } t > 0. \tag{30}$$

If the objective function of (29) is zero, all the distances between the normalized P_ℓ are not smaller than 1 and, so, the Kissing number is greater than or equal to $npun$. Defining

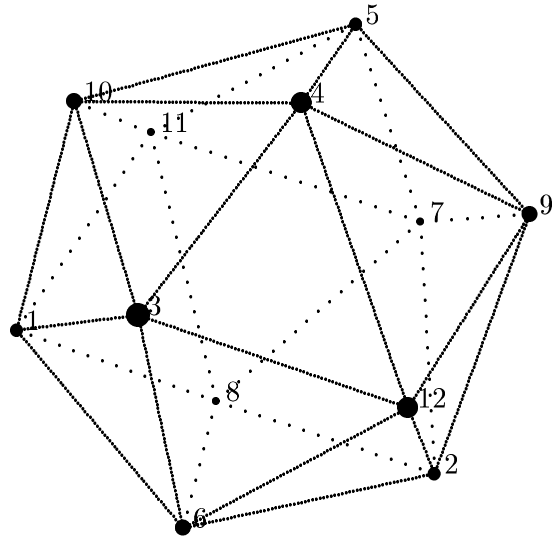
$$H_k(t) = 0 \text{ if } t \leq 0, \quad H_k(t) = kt^2/(1 + kt^2) \text{ if } t > 0, \tag{31}$$

we obtain the following smooth approximation of the discontinuous problem (29)

$$\text{Minimize} \sum_{i=1}^{npun-1} \sum_{j=i+1}^{npun} H_k[1 - \|P_i/\|P_i\| - P_j/\|P_j\|\|]. \tag{32}$$

Again, if the objective function of (32) is null, the kissing number is at least $npun$. The problems (32) are solved using the algorithm [5] taking as initial approximations random points with entries between -5 and 5. The code (KISCOUNT) is available in [17]. It is interesting to analyze the solutions of (29) and (32) compared with the minimax formulations of [8].

For $npun = 12$ and $ndim = 3$ the solution of the minimax reformulation is a regular icosahedron whereas the solution of (32) (with $k = 10$) is a slightly irregular icosahedron with many edges exactly equal to 1. See Picture 1. The code used to produce the Latex-file for this picture (called CIRCLE) is available in [17].



Picture 1: Irregular Icosahedron

4.3 Maximizing number of satisfied constraints

In some practical situations, one wishes to maximize the number of constraints that are satisfied at the solution of a problem. Suppose that the constraints are $g_i(x) \leq 0, i = 1, \ldots, m$. Then, the problem of maximizing the number of satisfied constraints, under restrictions given by Ω, is

$$\text{Minimize} \sum_{i=1}^{m} H[g_i(x)] \text{ subject to } x \in \Omega, \tag{33}$$

where H is given by (30). Defining H_k by (31), the approximate problem is

$$\text{Minimize } \sum_{i=1}^{m} H_k[g_i(x)] \text{ subject to } x \in \Omega. \tag{34}$$

Our first experiment of this class consists in finding the circle of radius 1 that contains the maximum number of given points of a set $\{Q_1, \dots, Q_m\}$. We generated randomly 50 points in the square $[4, 6] \times [4, 6]$. The unknowns are the coordinates of the center C of the circle. Therefore, (33) corresponds to

$$\text{Minimize } \sum_{i=1}^{m} H[\|C - Q_i\|^2 - 1] \tag{35}$$

and the approximate problem is

$$\text{Minimize } \sum_{i=1}^{m} H_k[\|C - Q_i\|^2 - 1]. \tag{36}$$

For solving (36), with $k = 10$, we used the trust-region algorithm described in [5]). The initial point was taken randomly in $[0, 10] \times [0, 10]$.

We used 100 initial points. In all the cases we obtained convergence to $(5.0919933, 5.0573850)$ where 14 constraints are violated. The computer time was negligible: around 0.04 seconds per initial point. The code (BOLAPUN) and the corresponding set of data (BOLAPUN.DAT) are available in [17].

The second experiment of type (33) is a practical application. The professor of a course of 64 students has applied four tests and wants do determine the weights of the corresponding scores in such a way that the number of approved students is the largest possible. The score of each test is between 0 and 10 and the minimum score for final approval is 5. Therefore, calling $P(i, j)$ the score of the student i in the test j and x_1, x_2, x_3, x_4 the weights, the constraints take the form

$$\sum_{j=1}^{4} x_j P(i, j) \geq 5 \text{ for all } i = 1, \dots, 64$$

or $g_i(x) \leq 0$, where

$$g_i(x) = 5 - \sum_{j=1}^{4} x_j P(i, j).$$

In this case, we also have the constraints

$$\sum_{i=1}^{4} x_i = 1, \quad 0.2 \leq x_i \leq 0.8 \; \forall \, i = 1, \ldots, 4.$$

We used the Augmented Lagrangian software [8] for solving the approximate problem. The (real) data for this problem was provided by Márcia Gomes-Ruggiero and is available in the home-page [17] (file MARCIA.DAT). We used 100 different initial points, chosen randomly between 0.2 and 0.8. The best solution was obtained 7 times and corresponded to the weights $0.2, 0.2, 0.4, 0.2$. For these weights only 17 students got a final record smaller than 5. The computer time of the resolution of the problem was less than 0.1 seconds for all the initial points. The code (APROVAR) and the data (MARCIA.DAT) are available in [17]. By the way, if instead of choosing the weights in order to maximize the number of approved students, we do it in order to minimize that number, with an obvious modification of APROVAR, we obtain the weights $0.25, 0.25, 0.25, 0.25$. With these weights, 22 students obtain a final record smaller than 5.

4.4 Minimizing number of unfitted observations

We wish to fit a model $y = F(a, x)$ to a set of experimental observations $(a_i, y_i), i = 1, \ldots, m$. The observations are subject to (perhaps systematic) experimental error and many of them can be completely unreliable. Therefore, instead of using standard minimization criteria (for example, least squares) we prefer to give a tolerance $\varepsilon_i > 0$ for each observation and to choose the parameters x in such a way that the number of observations whose ordinate exceed the predicted one more than ε_i is minimized. Assume that we wish to estimate n parameters, so $x \in I\!R^n$. Our goal is to minimize the number of observations such that

$$|y_i - F(a_i, x)| > \varepsilon_i.$$

Using $H(t)$ given by (30), this is equivalent to

$$\text{Minimize} \sum_{i=1}^{m} H[y_i - F(a_i, x) - \varepsilon_i] + H[F(a_i, x) - \varepsilon_i - y_i]. \tag{37}$$

The organizer of a poll knows that the answer of a determined question should be, approximately, a linear function of the answer to the others but, occasionally, lies must be discarded. To simulate that situation, we generate a random $(0, 10)$ matrix $A \in I\!R^{m\times n}$, $m = 100, n = 8$, $A = (a_{ij})$, we generate weights $w_i \in [0, 1]$ such that $\sum_{i=1}^n w_i = 1$ and we compute $y = Aw$. The entries of A and y are rounded to two decimal places after the dot. Then, 10% of the observations $\hat{y}_i$ are discarded and replaced by 100. We set $\varepsilon_i = 0.1y_i$ for all $i = 1, \ldots, m$. Therefore, our objective is to solve (37) with

$$F(a_i, x) = \sum_{j=1}^{n} a_{ij}x_j \text{ for all } i = 1, \ldots, m.$$

We use the approximation (31) with $k = 10$. As before, the minimization tool is the augmented Lagrangian [8]. The code is available in [17] under the name UNFIT and the generated data are in UNFIT.DAT. In 82 (out of 100) cases the code leaved exactly 10 unfitted points (outliers), which were the ones discarded in the data generation. The computer time was less than 0.1 seconds per (random) initial point. After elimination of the outliers, we proceeded to a least squares estimation using CONVLSQ (see [17]).

4.5 Covering with balls

Given $npun$ points in $I\!R^{ndim}$ we are interested in finding $nball$ balls of a given radius whose union cover the largest possible number of points. Assume that the points are $P_1, \ldots, P_{npun}$ and the required radius is r. Using (30) we wish to find centers $C_1, \ldots, C_{nball}$ that solve the problem

$$\text{Minimize} \sum_{i=1}^{npun} H(\min\{[\|P_i - C_j\|^2 - r^2], j = 1, nball\}). \tag{38}$$

We approximate (38) using (31) with $k = 10$. The code (NBALLS), which is available in [17], uses the algorithm described in [5]. For testing the code we established three centers in $I\!R^3$, $(1, 1, 1)$, $(2, 2, 2)$ and $(4, 4, 4)$ and we generated randomly 100 points in the union of the corresponding unitary balls. See BALL3.DAT in [17]. We ran NBALLS using random initial approximations for the centers. Each coordinate of the initial approximation was between the smaller and the larger coordinate of the data points. The code is prepared to use an arbitrarily large number of initial points but it stops when it finds a set of centers according to which all the data points are covered. In this case, with the first initial point we obtained centers that leave 32 uncovered points. With the second initial approximation, the obtained balls covered all the points. The computer centers were $(1.1, 1, 0.99), (2.12, 2.13, 1.97), (3.97, 3.95, 4.02)$. The total computer time was 0.13 seconds.

5 Final remarks

The classical continuous nonlinear programming problem

$$\text{Minimize } f(x) \text{ s.t. } g(x) \leq 0, \ x \in \Omega, \tag{39}$$

where $f : I\!R^n \to I\!R, g : I\!R^n \to I\!R^m$ are continuous, is (essentially) equivalent to

$$\text{Minimize } f(x) + \sum_{i=1}^{m} MH[g_i(x)] \text{ s.t. } x \in \Omega, \tag{40}$$

where H is defined by (30) and M is a "very large" number. Many penalty methods, including the one based in the quadratic loss function [2], for solving (39) can be interpreted as the application of the smoothing process to (40). The quadratic loss penalty term can be thought as the result of smoothing $H(t)$ by

$$H_k^{quad}(t) = 0 \text{ if } t \leq 0,$$

$$H_k^{quad}(t) = \min\{kt^2, M\} \text{ if } t > 0. \tag{41}$$

The exponential penalty function [1, 2, 11] corresponds to

$$H_k^{exp}(t) = \min\{e^{kt}/k, M\}. \tag{42}$$

The term $\min\{\cdot, M\}$ in (41), (42) and other penalty functions looks awkward but, in fact, corresponds to what is done in computer implementations where a finite maximum penalty term exists, either by practical decision of the programmer or by the computer limitation.

The analogy of the smoothing approach for discontinuous test functions and the penalty approach for nonlinear programming suggests that Lagrangian modifications of the smoothing functions can be introduced with the aim of obtaining good approximations of the solution of (3) for moderate values of k. Instead of using $H_k(t)$, the idea is to use $H_k(t + \lambda)$ in such a way that break-point solutions could be obtained for finite values of k. This will be the subject of future research.

Acknowledgement

The author is indebted to two anonymous referees for useful comments.

References

[1] R. R. Allran and S. E. V. Johnsen. An algorithm for solving nonlinear programming problems subject to nonlinear inequality constraints. *Computer Journal* 13, pp. 171-177 (1970).

[2] D. Bertsekas. *Nonlinear Programming*, 2nd edition. Athena Scientific, Belmont, Massachusetts, 1999.

[3] A. R. Conn and M. Mongeau. Discontinuous piecewise linear optimization. *Mathematical Programming* 90, pp. 315-380 (1998).

[4] J. H. Conway and N. J. C. Sloane. *Sphere packings, lattices and groups.* Springer-Verlag, New York, 1998.

[5] A. Friedlander, J. M. Martínez and S. A. Santos. A new trust region algorithm for bound constrained minimization. *Applied Mathematics and Optimization* 30, pp. 235-266 (1994).

[6] R. Heuts and J. Deklein. An (S-Q) inventory model with stochastic and interrelated lead times. *Naval Research Logistics* 42, pp. 839-859 (1995).

[7] K. Kamimura and H. Nishino. An efficient method for determining economical configurations of elementary packet-switched networks. *IEEE Transactions on Communications* 39, pp. 278-288 (1991).

[8] N. Krejić, J. M. Martínez, M. P. Mello and E. A. Pilotta. Validation of an augmented Lagrangian algorithm with a Gauss-Newton Hessian approximation using a set of hard-spheres problems. *Computational Optimization and Applications* 16, pp. 247-263 (2000).

[9] Maia Loa, L. A. V. de Carvalho and R. Y. Qassim. Synthesis of utility systems by simulated annealing. *Computers and Chemical Engineering* 19, pp. 481-488 (1995).

[10] J. M. Martínez and B. F. Svaiter. A practical optimality condition without constraint qualifications for nonlinear programming. Technical Report, Institute of Mathematics, UNICAMP, Brazil, 1999.

[11] F. H. Murphy. A class of exponential penalty functions. *SIAM Journal on Control* 12, pp. 679-687 (1974).

[12] R. M. Pert, C. E. French and G. W. Isaacs. Optimizing systems when components have discontinuous cost-functions. *Operations Research* 9, pp. 468-478 (1961).

[13] G. Rozvany. Optimal plastic design with discontinuous cost functions. *Journal of Applied Mechanics - Transactions of the ASME* 41, pp. 309-310 (1974).

[14] A. Shulman and R. Vachani. A decomposition algorithm for capacity expansion of local access networks. *IEEE Transactions on Communications* 41, pp. 1063-1073 (1993).

[15] M. Turkay and I. E. Grossmann. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs multiple size regions. *Industrial & Engineering Chemistry Research* 35, pp. 2611-2623 (1996).

[16] J. Walker. Decision support for the single-period inventory problem. *Industrial Management & Data Systems* 100, pp. 61-66 (2000).

[17] `http://www.ime.unicamp.br/~martinez/software.htm`.