Discussiones Mathematicae
Differential Inclusions, Control and Optimization 20 (2000) 147–158

NUMERICAL BEHAVIOR OF THE METHOD OF PROJECTION ONTO AN ACUTE CONE WITH LEVEL CONTROL IN CONVEX MINIMIZATION

Robert Dylewski

Institute of Mathematics, Technical University
ul. Podgórna 50, PL-65–246 Zielona Góra, Poland

e-mail: r.dylewski@im.pz.zgora.pl

Abstract

We present the numerical behavior of a projection method for convex minimization problems which was studied by Cegielski [1]. The method is a modification of the Polyak subgradient projection method [6] and of variable target value subgradient method of Kim, Ahn and Cho [2]. In each iteration of the method an obtuse cone is constructed. The obtuse cone is generated by a linearly independent system of subgradients. The next approximation of a solution is the projection onto a translated acute cone which is dual to the constructed obtuse cone. The target value which estimates the minimal objective value is updated in each iteration. The numerical tests for some tests problems are presented in which the method of Cegielski [1] is compared with the method of Kim, Ahn and Cho [2].

Keywords: convex nondifferentiable minimization, projection method, subgradient method, acute cone, obtuse cone.

1991 Mathematics Subject Classification: 65K05, 90C25.

1 Introduction

1.1 The convex minimization problem

In this paper we consider the convex minimization problem

$$\begin{array}{c} \text{minimize } f(x) \\ \text{subjectto } x \in D, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function (not necessarily differentiable), $D \subset \mathbb{R}^n$ is a convex, compact subset.

We suppose that:

- for any $x \in D$ we can evaluate $f(x)$ and a subgradient $g_f(x)$,
- for any $x \in \mathbb{R}^n$ we can evaluate $P_D(x)$ – the metric projection of x onto D.

1.2 Notation

We use the following notation:

x^j – j-th coordinate of an element $x = (x^1, ..., x^n)^\top \in \mathbb{R}^n$,
x_k – k-th element of a sequence $\{x_k\}$,
$\langle x, y \rangle = x^\top y$ – usual scalar product of x and y in $\mathbb{R}^n$,
$\|x\| = \sqrt{\langle x, x \rangle}$ – Euclidean norm of x,
$P_D(x) = \arg\min_{z \in D} \|z - x\|$ – the metric projection of x onto D,
$S(f, \alpha) = \{x \in \mathbb{R}^n : f(x) \leq \alpha\}$ – the sublevel set of f for a level α,
$f^* = \min_{x \in D} f(x)$ – the minimal value of f on D,
$M = \operatorname{Arg\,min}_{x \in D} f(x)$ – the solution set,
$\partial f(x) = \{g \in \mathbb{R}^n : f(y) - f(x) \geq \langle g, y - x \rangle, y \in \mathbb{R}^n\}$ – the subdifferential of f at x,
$g_f(x)$ – a subgradient of f at x (any element of $\partial f(x)$),
$g_k = g_f(x_k)$,
$f_k(\cdot) = \langle g_k, \cdot - x_k \rangle + f(x_k)$ – a linearization of f at x_k,
L – a Lipschitz constant of f on D,
$\operatorname{diam}(D) = \sup_{x,y \in D} \|x - y\|$ – the diameter of D,
R – an upper approximation of $\operatorname{diam}(D)$,
$f_{L_k} = \max_{i \in L_k} f_i$ for $L_k \subset \{1, 2, ..., k\}$ – a lower approximation of f,
$C^* = \{s \in \mathbb{R}^n : \langle s, x \rangle \leq 0, x \in C\}$ – a cone dual to a given cone C,
$\operatorname{cone} S$ – the cone generated by a subset $S \subset \mathbb{R}^n$,
$\operatorname{Lin} S$ – the linear subspace generated by a subset $S \subset \mathbb{R}^n$.

Furthermore, we identify a matrix A with the system of vectors determined by the columns of A and denote by $\operatorname{cone} A$ the cone generated by the columns of A. A cone C is said to be *acute* if $\langle x, y \rangle \geq 0$ for all $x, y \in C$. A cone C is said to be *obtuse* (in $\operatorname{Lin} C$) if $C^* \cap \operatorname{Lin} C$ is an acute cone.

2 The method of projection onto an acute cone with level control

In this section, we recall a projection method of Cegielski [1], so called the method of projection onto an acute cone with level control. The method has the form

$$x_1 \in D - \text{arbitrary} \tag{2.1}$$
$$x_{k+1} = P_D(x_k + \lambda_k t_k),$$

where $\lambda_k \in (0,2)$,

$$t_k = P_{\bigcap_{i \in L_k} S(f_i, \widetilde{f}_k)}(x_k) - x_k, \tag{2.2}$$

$L_k \subset \{1, 2, ..., k\}$ is such that $k \in L_k$ and

$$\widetilde{f}_k = (1-\nu)\overline{f}_k + \nu \underline{f}_k$$

denotes the current level (an approximation of f^*), where $\nu \in (0,1]$, $\underline{f}_k$ denotes a lower bound of f^* and $\overline{f}_k = \min_{1 \le i \le k} f(x_i)$ is an upper bound of f^*.

In the method

- $\underline{f}_1$ and R are supposed to be known and $\underline{f}_k$ is updated in each iteration,
- L_k is selected by an obtuse cone model and is such that $L_k \subset \{1, 2, ..., k\}$, $k \in L_k$.

More precisely, the sequence x_k is generated by the following iterative scheme which is a special case of [1, Iterative scheme 2.6].

Iterative scheme 2.1. (*The method of projection onto an acute cone with level control*)

Step 0. (*Initialization*)

0.1. Choose: $x_1 \in D$ (*starting point*), $\varepsilon \ge 0$ (*optimality tolerance*), $\lambda \in (0,2)$, (*relaxation parameter*), $\nu \in (0,1)$ (*level parameter*), $R \ge d(x_1, M)$ (*upper bound of the distance of the starting point x_1 to the solution set*), $\underline{f}_1 \in (-\infty, f^*]$ (*initial lower bound of f^**), m-number of saved linearizations.

0.2. Set: $k = 1$ (*iterations counter*), $l = 0$ (*lower bound updates counter*), $\overline{f}_0 = +\infty$(*initial upper bound of f^**), $r_1 = 0$ (*initial distance parameter*), $\overline{x}_1 = x_1$.

Step 1. (*Objective evaluations*)
Evaluate $f(x_k)$ and g_k.

Step 2. (*Upper bound update*)
If $f(x_k) < \overline{f}_{k-1}$ set $\overline{f}_k = f(x_k)$ and $\overline{x}_k = x_k$. Otherwise, set $\overline{f}_k = \overline{f}_{k-1}$ and $\overline{x}_k = \overline{x}_{k-1}$.

Step 3. (*Stopping criterion*)
3.1. If $\overline{f}_k - \underline{f}_k \leq \varepsilon$ then terminate ($\overline{x}_k$ *is an* ε*-optimal solution*).
3.2. If $\|g_k\| R \leq \varepsilon$ then terminate (x_k *is an* ε*-optimal solution*).

Step 4. (*Level update*)
Set $\widetilde{f}_k = (1-\nu)\overline{f}_k + \nu \underline{f}_k$.

Step 5. (*Update of saved linearizations of* f)
Set $J_k \subset \{k-m+1, ..., k\}$ such that $k \in J_k$.

Step 6. (*Obtuse cone model selection*)
6.1. Choose an appropriate subset $L_k \subset J'_k = \{j \in J_k : f_j(x_k) \geq \widetilde{f}_k\}$ such that $k \in L_k$ and such that the system $G_k = [g_j : j \in L_k]$ is linearly independent and generates an obtuse cone.
6.2. If the equality $S_k := \bigcap_{i \in L_k} S(f_i, \widetilde{f}_k) = \emptyset$ is detected then go to Step 10 ($\underline{f}_k$ *is too low*).

Step 7. (*Projection onto an acute cone*)
7.1. Construct $t_k = P_{S_k}(x_k) - x_k$.
7.2. Evaluate $z_k = x_k + \lambda t_k$, $z'_k = P_D(z_k)$ and $q_k = z'_k - z_k$.

Step 8. (*Inconsistency detection*)
8.1. Set $r'_k = r_k + \lambda(2-\lambda)\|t_k\|^2 + \|q_k\|^2$ and $r''_k = r_k + \|t_k\|^2$.
8.2. If $r'_k > R^2 - (R - \|z'_k - x_{k'+1}\|)^2$ or $r''_k > R^2 - (R - \|x_k + t_k - x_{k'+1}\|)^2$, where k' is the last iteration in which Step 10 was executed, then go to Step 10 ($\widetilde{f}_k$ *is too low*).

Step 9. (*Approximation update*)
Set $x_{k+1} = z'_k$, $r_{k+1} = r'_k$, increase k by 1 and go to Step 1.

Step 10. (*Lower bound update*)
10.1. Set $\underline{f}_{k+1} = \widetilde{f}_k$.
10.2. Set $\overline{f}_{k+1} = \overline{f}_k$ and $\overline{x}_{k+1} = \overline{x}_k$.
10.3. Set $r_{k+1} = 0$, $x_{k+1} = \overline{x}_k$, increase k and l by 1 and go to Step 3.

In [1, Section 3], the convergence analysis of the above iterative scheme can be found.

Steps 6 and 7 are most important in Iterative scheme 2.1. Now we recall a construction of a subset $L_k \subset J'_k$ fulfilling the conditions in Step 6.1 (obtuse cone model). This construction is described in details in [1].

Algorithm 2.2. (*Construction of an obtuse cone and of the projection onto an acute cone*)

Step 0. (*Initialization*)
Set $J'_k = \{j \in J_k : f_j(x_k) \geq \widetilde{f}_k\}$, $L = \{k\}$, $G = g_k$ and $C = \|g_k\|$.

Step 1. Set $K = \emptyset$.

Step 2. If $L \cup K = J'_k$ go to Step 5.

Step 3. Choose any $r \in J'_k \backslash (L \cup K)$.

Step 4. (*Obtuse cone detection*)
If $\gamma = (CC^\top)^{-1} G^\top g_r \leq 0$ then
4.1. Set $L := L \cup \{r\}$ and $G := [G, g_r]$.
4.2. Make the update $C := [C, c_r]^\top$ of the Cholesky factorization $CC^\top$ of $G^\top G$. If the Cholesky procedure breaks down, print $\widetilde{f}_k \leq f^*$ and terminate.
4.3. Go to Step 1.
Otherwise set $K := K \cup \{r\}$ and go to Step 2.

Step 5. (*Output data of an obtuse cone*)
5.1. Set $L_k = L$, $G_k = G$ and $C_k = C$.
5.2. Set $t_k = -G_k(C_k C_k^\top)^{-1}(G_k^\top x_k - b_k)$, where b_k is a vector with coordinates $b_k^j = c^j + \widetilde{f}_k$, $j \in L_k$, where $c^j = \langle g_j, x_j \rangle - f(x_j)$.

Step 6. Terminate.

The proof of the fact that the Iterative scheme 2.1 together with the construction of an obtuse cone described in Algorithm 2.2 generates a sequence which converges to a solution of the problem (1.1) can be found in [1].

3. Numerical tests

In this section we present the computation results of the presented method.

3..1 Tests problems

3.1.1. Shor's test problem (Shor) [8]

$f(x) = \max\Big\{ b_i \sum_{j=1}^{5} (x^j - a_{ij})^2 : i = 1, ..., 10 \Big\}$,
$n = 5$, $f^* = 22.60016210$, $\|x_1 - x^*\| = 2.2955$, $f(x_1) = 80$.

3.1.2. Goffin's test problem (Goffin)

$f(x) = n \max\{x^j : j = 1, ..., n\} - \sum_{j=1}^{n} x^j,$

$n = 15,\ f^* = 0,\ \|x_1 - x^*\| = 16.733,\ f(x_1) = 105,$

$n = 50,\ f^* = 0,\ \|x_1 - x^*\| = 102.042,\ f(x_1) = 1225.$

3.1.3. Hilbert's test problem (L1hil)

$f(x) = \sum_{i=1}^{n} \left| \sum_{j=1}^{n} (x^j - 1) / (i + j - 1) \right|,$

$n = 10,\ f^* = 0,\ \|x_1 - x^*\| = 3.162,\ f(x_1) = 13.3754.$

3.1.4. Todd's test problem (Todd) [9]

$f(x) = \max\{0, \delta x^1 + \delta_2 x^2 + 2\delta_3 x^3 : \delta_2, \delta_3 = \pm 1\},$

$n = 3,\ f^* = 0,\ \delta = 0.1,\ f(x_1) = 0.1990.$

3.1.5. Lemaréchal's test problem (Maxquad) [5]

$f(x) = \max\left\{x^\top A_i x - x^\top b_i : i = 1, ..., 5\right\},$

$n = 10,\ f^* = -0.84140833,\ \|x_1 - x^*\| = 3.189,\ f(x_1) = 5337.$

3.1.6. Strongly convex problems (scp)

$f(x) = \max\left\{a_i^\top x + b_i : i = 1, ..., m\right\} + s \sum_{j=1}^{n} (x^j - c_j)^2,$

s − strong convexity constant,

a_i, b_i − randomly generated in the interval $[-1, 1]$,

c_j − randomly generated in the interval $[-2, 2]$,

$m = 10, 20, 50, 100,\ n = 5, 20, 30, 50.$

3..2 Results of numerical tests

Now we present the results of numerical tests for the method of the projection onto an acute cone with level control, described in Section 2, called here the PAC method.

The method was programmed in Fortran 90 (Lahey Fortran 90 v.3.5). All floating point calculations were performed in double precision, allowing the relative accuracy of $2,2 \cdot 10^{-16}$.

In all tests the stopping criterion $\overline{f}_k - \underline{f}_k \leq \varepsilon$ was employed with the absolute optimality tolerance ε, the number of stored subgradients $\#J_k = 100$ and the relaxation parameter $\lambda = 1$.

First we consider the case f^* is known. In this case we set $\underline{f}_1 = f^*$, $\nu = 1$. We compare two methods: the PAC method and the Polyak method, which is a special case of the PAC method – one takes $L_k = \{k\}$ in Iterative scheme 2.1. The results are presented in Table 1.

		PAC method		Polyak method	
Function	ε	$\#f/g$	$f(x_k)$	$\#f/g$	$f(x_k)$
Shor n=5 $f^* = 22.60016210$	10^{-2}	18	22.60899720	1713	22.61012596
	10^{-4}	29	22.60019525	$> 5 * 10^4$	22.60050317
	10^{-6}	39	22.60016287	-	-
	10^{-8}	48	22.60016210	-	-
Goffin n=15 $f^* = 0.0$	10^{-2}	15	0.00000000	597	0.00995232
	10^{-4}	15	0.00000000	1037	0.00009805
	10^{-6}	15	0.00000000	1476	0.00000100
	10^{-8}	15	0.00000000	1916	0.00000001
Goffin n=50 $f^* = 0.0$	10^{-2}	50	0.00000000	7717	0.00999477
	10^{-4}	50	0.00000000	13207	0.00009994
	10^{-6}	50	0.00000000	18696	0.00000100
	10^{-8}	50	0.00000000	24187	0.00000001
L1hil n=10 $f^* = 0.0$	10^{-2}	10	0.00102746	140	0.00984823
	10^{-4}	13	0.00001424	$> 5 * 10^4$	0.00024440
	10^{-6}	17	0.00000011	-	-
	10^{-8}	27	0.00000000	-	-
Todd n=3 $f^* = 0.0$	10^{-2}	5	0.00000000	748	0.00998941
	10^{-4}	5	0.00000000	1899	0.00009988
	10^{-6}	5	0.00000000	3050	0.00000100
	10^{-8}	5	0.00000000	4200	0.00000001
Maxquad n=10 $f^* = -0.84140833$	10^{-2}	23	-0.83204753	684	-0.83147061
	10^{-4}	33	-0.84131302	$> 5 * 10^4$	-0.84127579
	10^{-6}	43	-0.84140758	-	-
	10^{-8}	54	-0.84140833	-	-

Table 1

Now we present the numerical tests for the case f^* is unknown. In this case we set $\nu = 0.5$. We compare two methods: the PAC method and the method of Kim, Ahn and Cho of projection with level control [2] (called here KAC), which is in fact a special case of the PAC method – one takes $L_k = \{k\}$ in Iterative scheme 2.1. The results are presented in Table 2.

			PAC method		KAC method	
Function	R	ε	$\#f/g$	$f(x_k)$	$\#f/g$	$f(x_k)$
Shor n=5 $\underline{f}_1 = 0.0$	100.0	10^{-2}	30	22.60280198	$> 5 * 10^4$	22.63133991
		10^{-4}	43	22.60020768	-	-
		10^{-6}	54	22.60016251	-	-
		10^{-8}	71	22.60016210	-	-
	3.0	10^{-2}	25	22.60280198	28946	22.60188847
		10^{-4}	38	22.60020768	-	-
		10^{-6}	49	22.60016251	-	-
		10^{-8}	63	22.60016210	-	-
Goffin n=15 $\underline{f}_1 = -100.0$	1000.0	10^{-2}	25	0.00014978	$> 5 * 10^4$	0.43445713
		10^{-4}	27	0.00004575	-	-
		10^{-6}	30	0.00000059	-	-
		10^{-8}	34	0.00000000	-	-
	18.0	10^{-2}	25	0.00393333	$> 5 * 10^4$	0.01526670
		10^{-4}	29	0.00006029	-	-
		10^{-6}	34	0.00000034	-	-
		10^{-8}	37	0.00000001	-	-
Goffin n=50 $\underline{f}_1 = -100.0$	1000.0	10^{-2}	64	0.00072351	$> 5 * 10^4$	32.72499431
		10^{-4}	70	0.00007746	-	-
		10^{-6}	77	0.00000009	-	-
		10^{-8}	80	0.00000000	-	-
	105.0	10^{-2}	64	0.00072351	$> 5 * 10^4$	0.43717914
		10^{-4}	70	0.00007746	-	-
		10^{-6}	77	0.00000009	-	-
		10^{-8}	80	0.00000000	-	-
L1hil n=10 $\underline{f}_1 = -100.0$	1000.0	10^{-2}	14	0.00466617	$> 5 * 10^4$	0.11135363
		10^{-4}	42	0.00004219	-	-
		10^{-6}	54	0.00000051	-	-
		10^{-8}	80	0.00000001	-	-
	4.0	10^{-2}	10	0.00563384	$> 5 * 10^4$	0.00481994
		10^{-4}	18	0.00004197	-	-
		10^{-6}	27	0.00000040	-	-
		10^{-8}	59	0.00000000	-	-
Maxquad n=10 $\underline{f}_1 = -10.0$	100.0	10^{-2}	236	-0.84060870	$> 5 * 10^4$	-0.83251823
		10^{-4}	285	-0.84139841	-	-
		10^{-6}	315	-0.84140810	-	-
		10^{-8}	508	-0.84140833	-	-
	4.0	10^{-2}	145	-0.84017311	27370	-0.84079048
		10^{-4}	236	-0.84139640	$> 5 * 10^4$	–0.84079048
		10^{-6}	282	-0.84140805	-	-
		10^{-8}	344	-0.84140833	-	-

Table 2

Finally, we present the numerical tests for several strongly convex problems 3.1.1, 3.1.6. For such problems an upper approximation R of diam(D) can be updated in each iteration. More precisely, we provide:

Step 1.2. Evaluate $R_k = \min\left\{\sqrt{\frac{f(x_k)-\underline{f}_k}{s}}, \frac{\|g_k\|}{2s}\right\}$

and substitute R by R_k in Iterative scheme 2.1, where s is a strong convexity constant (see [2] for details). Similarly, for strongly convex problems $\underline{f}_k$ is additionally updated by adding the following step to Iterative scheme 2.1:

Step 1.3. If $f(x_k) - \frac{\|g_k\|^2}{2s} > \underline{f}_k$, then set $\underline{f}_{k+1} = f(x_k) - \frac{\|g_k\|^2}{2s}$ and go to Step 10.2, (see [2] for details). We call such a modification of Iterative scheme 2.1 the strongly convex (sc) variant of the PAC method.
Two methods are compared for such a modification of Iterative scheme 2.1: the PAC method and the KAC method.

a) The results for the strongly convex problem 3.1.6 with the strong convexity constant $s = 1$ are presented in Table 3.

		PAC method		PAC method,		KAC method	
$m \times n$	ε			sc variant		sc variant	
		$\#l$	$\#f/g$	$\#l$	$\#f/g$	$\#l$	$\#f/g$
10×5	10^{-2}	12	23	7	10	8	38
	10^{-4}	18	33	13	19	14	3068
	10^{-6}	23	41	18	24	-	$> 3 * 10^4$
	10^{-8}	30	44	25	26	-	-
20×20	10^{-2}	12	18	11	16	12	190
	10^{-4}	17	27	16	23	15	12472
	10^{-6}	23	34	21	27	-	$> 3 * 10^4$
	10^{-8}	28	39	28	28	-	-
50×30	10^{-2}	12	26	9	13	9	70
	10^{-4}	18	35	14	21	18	6149
	10^{-6}	24	43	20	28	-	$> 3 * 10^4$
	10^{-8}	30	45	26	29	-	-
100×50	10^{-2}	12	23	15	20	13	268
	10^{-4}	17	31	21	25	18	12328
	10^{-6}	23	38	27	31	-	$> 3 * 10^4$
	10^{-8}	29	41	32	33	-	-

Table 3

b) The results for the Shor's problem 3.1.1 are presented in Table 4 (note, that the Shor's function is strongly convex with a strong convexity constant $s = 1$).

		PAC method		KAC method	
		sc variant		sc variant	
Function	ε	$\#f/g$	$f(x_k)$	$\#f/g$	$f(x_k)$
Shor	10^{-2}	27	22.60280198	6204	22.60184265
n=5	10^{-4}	40	22.60020768	34692	22.60020224
$\underline{f}_1 = 0.0$	10^{-6}	51	22.60016251	$> 5 * 10^4$	22.60019054
	10^{-8}	64	22.6001620	-	-

Table 4

3..3 Conclusions

First we explain why we have compared the method of projection onto an acute cone (the PAC method) with the variable target subgradient method of Kim, Ahn and Cho [2] (the KAC method) and with its variant – the subgradient method of Polyak [6]. As we have previously observed the last two methods are special cases of the PAC method: obtuse cones are one-dimensional in the Polyak and in the KAC method. Therefore, we search in the numerical tests the influence of the possibility of the construction of multi-dimensional obtuse cones in Algorithm 2.2 on the speed of convergence. In both cases we see that the employing of the obtuse cone model accelerates considerably the convergence. We see that the Polyak method and the KAC method did not attain the optimality tolerance $\varepsilon = 10^{-4}$ in a reasonable number of objective evaluations. Even the previously described update of R_k and $\underline{f}_k$ in the strongly convex case does not help the KAC method to attain the optimality tolerance $\varepsilon = 10^{-4}$. However, the PAC method converges better after this additional update of R_k and $\underline{f}_k$. Furthermore, for the PAC method the geometrical convergence in all cases and for all tested functions is observed, as Tables 1 and 2 show.

The knowledge of the minimal objective value f^* does not have so much influence on the convergence, as theoretically expected. We have also tested the PAC method for much less initial values of lower bounds $\underline{f}_1$ of f^* than the presented in Table 2. The number of objective evaluations was only a bit bigger than for those presented in Table 2 values of $\underline{f}_1$. The results presented in Table 2 show also that the estimation R of $\|x_1 - x^*\|$ has only little influence on the convergence. The cause of this surprising phenomenon is the fact that in almost all cases the lower bound $\underline{f}_1$ was updated in Step 10 of Iterative scheme 2.1 by detection that $S_k = \emptyset$ in Step 6.2 (actually, this equality has been detected by the Cholesky procedure in Step 4.2 of Algorithm 2.2). The inconsistency has been detected very rarely in Step 8 of Iterative scheme 2.1, contrary to the KAC method.

Very good results of the PAC method for the Goffin's test function can be explained by a special form of the Goffin's function $f(x) = \max_{i \in I} f_i(x)$, where all gradients $\nabla f_i(x)$ generate an obtuse cone, since $\langle \nabla f_i(x), \nabla f_j(x) \rangle < 0$ for all i, j, $i \neq j$. Therefore, the constructed obtuse cones have big dimension for this test function.

We have not compared the PAC method with methods of "bundle type" [3, 4, 7]. This comparison will be the aim in our next study. An initial analysis shows that in some cases the behavior of the PAC method is comparable with the bundle methods, in other cases the bundle methods converge better than the PAC method does. Nevertheless, the cost of one iteration for the PAC method seems to be smaller than for the bundle methods. These methods employ namely QP-procedure in each iteration. On the other hand, Algorithm 2.2 seems to be relatively cheap. Note, that the complexity of this Algorithm has the most important influence on the cost of one iteration of the PAC method.

Acknowledgement

I would like to thank Professor Andrzej Cegielski for his collaboration and many helpful suggestions during the preparation of the paper.

References

[1] A. Cegielski, *A method of projection onto an acute cone with level control in convex minimization*, Mathematical Programming **85** (1999), 469–490.

[2] S. Kim, H. Ahn and S.-C. Cho, *Variable target value subgradient method*, Mathematical Programming **49** (1991), 359–369.

[3] K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Springer-Verlag, Berlin 1985.

[4] C. Lemaréchal, A.S. Nemirovskii and YU.E. Nesterov, *New variants of bundle methods*, Mathematical Programming **69** (1995), 111–147.

[5] C. Lemaréchal and R. Mifflin, *A Set of Nonsmooth Optimization Test Problems*, in: Nonsmooth Optimization, C. Lemaréchal and R. Mifflin, eds., Pergamon Press, Oxford (1978), 151–165.

[6] B.T. Polyak, *Minimization of unsmooth functionals*, Zh. Vychisl. Mat. i Mat. Fiz. **9** (1969), 509–521 (Russian).

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

[8] N.Z. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, Heidelberg 1985.

[9] M.J. Todd, *Some remarks on the relaxation method for linear inequalities*, Technical Report 419, Cornell University, Cornell, Ithaca 1979.

Received 5 November 1999
Revised 7 March 2000