Discussiones Mathematicae
Differential Inclusions, Control and Optimization 20 (2000) 171–194

LARGE-SCALE NONLINEAR PROGRAMMING ALGORITHM USING PROJECTION METHODS

Paweł Białoń

National Institute of Telecommunications
Decision Support System Laboratory
ul. Szachowa 1, 04–894 Warszawa, Poland

e-mail: P.Bialon@itl.waw.pl

Abstract

A method for solving large convex optimization problems is presented. Such problems usually contain a big linear part and only a small or medium nonlinear part. The parts are tackled using two specialized (and thus efficient) external solvers: purely nonlinear and large-scale linear with a quadratic goal function. The decomposition uses an alteration of projection methods. The construction of the method is based on the zigzagging phenomenon and yields a non-asymptotic convergence, not dependent on a large dimension of the problem. The method preserves its convergence properties under limitations in complicating sets by geometric cuts. Various aspects and variants of the method are analyzed theoretically and experimentally.

Keywords: nonlinear optimization, large scale optimization, projection methods, zigzagging.

1991 Mathematics Subject Classification: 65M99, 49M45, 49M27, 65-06, 90C06, 90C99.

1 Introduction

The goal of the author was to investigate how projection methods (for feasibility problems) can be utilized in decomposing large scale nonlinear optimization problems. Several observations led to the proposed approach.

First, large nonlinear problems usually contain a big linear part and only a small or average nonlinear part (with the size measured, say, by the number of constraints). Nonlinear functions carry much more information than linear ones. Developing and validating an optimization model with a large nonlinear part would be difficult. An appropriate decomposition of the problem onto both the parts has a potential of accelerating the solution process. The big linear part could be tackled by a specialized, fast solver (actually, allowing for linear constraints and a quadratic goal function).

Second, many practitioners prefer a fast convergence in early stages of an optimization run (quick hitting a neighborhood of the optimal point) over often analyzed asymptotic properties, like superlinear convergence.

Moreover, the author would like to treat the goal and constraint functions uniformly, in order to simplify the algorithm design and analysis.

The last two motivations have determined the use of projection methods for feasibility problems [2, 6]. It is possible to reduce the original optimization problem to a sequence of feasibility problems, i.e. problems of finding a common point of several sets. The minimized goal function $f(x)$ may be replaced with a constraint of the form of $f(x) < Q$, where changing parameter Q, the predicted optimal value of the problem, within an *outer loop* (a bisection scheme or the level control scheme [7]) generates the sequence. The paper discusses mainly the *inner loop*, i.e. solving the feasibility problem. Some remarks and analyzes directed towards the outer loop are also given.

The feasibility problem is solved by making sequential projections onto two sets: one "linear" (defined by linear constraints) and one "nonlinear" (defined by nonlinear constraints). The projections are realized by specialized optimization solvers: quadratic and nonlinear. The use of the former well fits the current optimization "market" tendencies. More and more authors of linear solvers offer nowadays quadratic extensions to their products.

An important feature of the described approach is that we do not have to add accelerating cuts after each projection. The user may resign cutting off either the linear set or the nonlinear set. Both the choices will turn out to have numerical motivations. However, resigning one of the groups of cuts would normally destroy the theoretical profit of cumulating cuts: a decrease in zigzagging. It is shown how to maintain the profit anyway.

The author compares his results to similar propositions of Kiwiel [10, 8, 9] and Cegielski [3, 4]. The common feature of all the methods is a cumulation of cuts, though realized in different ways. The author gives also some general remarks regarding the behavior of an algorithm under a cumulation of cuts.

Usually an optimization problem is not reduced to a sequence of feasibility problems but the subgradient method of Polyak [16] or one of the many derivatives like that of Lemaréchal, Nemirovskii and Nesterov [12] is used. In [12], projecting onto a level set of a linear model of the objective function – which corresponds to the movement in the direction opposite to a subgradient – is replaced with projecting onto its more accurate model, built from cumulated successive cuts. The works [4, 3, 10, 8, 9] propose cut cumulation techniques and almost all of them show how to embed it in a subgradient algorithm. Unfortunately, these methods issue projections onto the admissible set of the original problem, which seemed hardly realizable, even with the used solvers: the set was composed of linear and nonlinear constraints and the projection looked hardly decomposable. The alternative technique of tackling constraints via nondifferentiable penalties would – except of some particular approaches – increase the Lipschitz constants of the goal function and cause a related (by the power of quadratic growth) decrease in the algorithm efficiency.

The article describes the method and its variants, the convergence analysis and the results of experimental investigations of its various aspects, as well as possible extensions. The paper pictures the current preliminary stage of the work on the method.

Notation. We shall operate in the finite-dimensional Hilbert space $\mathbb{R}^n$, whose members are identified with column vectors, with scalar product $\langle x, y\rangle = x^\top y$. The default norm will be the Euclidean norm $\|\cdot\| = \sqrt{\langle\cdot,\cdot\rangle}$. This norm by default generates other objects, e.g., distance of a point from a set. We use the following notation:

$\mathrm{dist}(A, B) = \inf_{a\in A,\, b\in B} \|a-b\|$ – distance between nonempty subsets A and B of $\mathbb{R}^n$,

$\mathrm{dist}(b, A) = \inf_{a\in A} \|a-b\|$ – distance of a point $b \in \mathbb{R}^n$ from a nonempty subset $A \subset \mathbb{R}^n$,

$\overline{ab}$ with $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ – line segment with ends a, b,

$P_A x = \arg\min_{y\in A} \|y - x\|$ – projection of $x \in \mathbb{R}$ onto a closed convex $A \subset \mathbb{R}^n$,

$\mathrm{Lin}\, A$ – linear subspace generated by $A \subset \mathbb{R}^n$,

cone – set $A \subset \mathbb{R}$ such that $x \in A,\ \alpha \geq 0 \Rightarrow \alpha x \in A$ and $x, y \in A \Rightarrow (x + y) \in A$,

$\mathrm{cone}\, A$ – cone generated by $A \subset \mathbb{R}^n$, i.e. the lowest cone containing A,

$C^\star = \{x \in \mathbb{R}^n :\ \forall_{y\in C}\ \langle x, y\rangle \leq 0\}$ – cone dual to a cone $C \subset \mathbb{R}^n$,

acute cone – any cone $C \subset \mathbb{R}^n$ such that $\forall_{x\in C,\, y\in C}\ \langle x, y\rangle \leq 0$,

obtuse cone – any cone $C \subset \mathbb{R}^n$ such that $C^\star \cap \mathrm{Lin}\, C$ is an acute cone.

Recall that the equivalent definition for y being $P_A x$ (A closed, convex) is

$$y \in A \wedge \forall_{z \in A}\ (z-y)^\top (y-x) \geq 0 \tag{1}$$

The inequality (1) in z (with x,y treated as constants) will be called *the cut based on projection $P_A x$ of a point x* or *the cut based on the step from x to y*. For natural i, let $i \bmod \infty = i$. For integer a, b where $a > b$, $\{a, \ldots, b\}$ denotes the empty set.

Acknowledgements. The final target of this work will be to use it in augmenting the IAC-DIDAS-N/DIDAS-N++ [11] decision support system developed in the Institute of Automatic Control at Warsaw University of Technology, and being now reconstructed at the National Institute of Telecommunications. This work was supported by National Institute of Telecommunications, Grant No. 063015. The author is greatful to Prof. Cegielski for his valuable comments, to Dr. Gondzio for making available his linear solver HOPDM (see [1] and the references therein), and to Dr. Granat for the guidance of this work. The author much appreciates the help of Dr. Makowski who gave his LPDIT [14] library of C++ classes for vectors, matrices and linear programming problems and last – but not least – the guidance of Prof. Wierzbicki, as well as his pointing out the question of the structure of large problems and other remarks.

2 Problem formulation

The initial optimization problem is defined as follows

$$\begin{array}{ll}
\min\limits_{x_N} f(x_N) & f : \mathbb{R}^{n_N} \to \mathbb{R} \\
\text{s.t.} & \\
\tilde{g}(x_N) \leq 0 & \tilde{g} : \mathbb{R}^{n_N} \to \mathbb{R}^{m_N - 1} \\
Ax \leq b & A \text{ is a matrix of type } m_{LI} \times n \\
Bx = d & B \text{ is a matrix of type } m_{LE} \times n \\
x_N^{lo} \leq x_N \leq x_N^{up},\ x_L^{lo} \leq x_L \leq x_L^{up}. &
\end{array} \tag{2}$$

Here $x = ({x_N}^\top, {x_L}^\top)^\top$, $x_N \in \mathbb{R}^{n_N}$, $x_L \in \mathbb{R}^{n_L}$, f and $\tilde{g}_i$ continuous, quasiconvex. A resulting feasibility problem $F(Q)$, where Q is a real parameter, consists in finding x satisfying

$$
\begin{aligned}
& g(x_N) \leq 0 \\
& Ax \leq b \\
& Bx = d \\
& x_N^{lo} \leq x_N \leq x_N^{up},\, x_L^{lo} \leq x_L \leq x_L^{up},
\end{aligned} \tag{3}
$$

where $g : \mathbb{R}^{n_N} \to \mathbb{R}^{m_N}$, $g_i(\cdot) = \tilde{g}_i(\cdot)$, $i = 1, \ldots m_N - 1$, $g_{m_N}(\cdot) = f(\cdot) - Q$. The feasibility problem has n_N nonlinear, n_L linear variables, m_N nonlinear inequality, m_{LI} linear inequality, m_{LE} linear equality constraints. Define $m = m_N + m_{LI} + m_{LE}$, $n = n_L + n_N$. It helps the efficiency of the approach when $m_N \ll m$ and also when $n_N \ll n$.

3 The idea of solving the feasibility problem

Define "nonlinear" and "linear" sets

$$
\begin{aligned}
N &= \{x_N \in \mathbb{R}^{n_N} :\ g(x_N) \leq 0 \wedge x_N^{lo} \leq x_N \leq x_N^{up}\} \\
L &= \{x_N \in \mathbb{R}^{n_N} :\ x_N^{lo} \leq x_N \leq x_N^{up} \wedge \exists_{x_L \in \mathbb{R}^{n_L}} (x_L^{lo} \leq x_L \leq x_L^{up} \\
&\qquad \wedge A({x_N}^\top, {x_L}^\top)^\top \leq b \wedge B({x_N}^\top, {x_L}^\top)^\top = d)\}
\end{aligned}
$$

Notice that these are not actually the sets of points allowed by nonlinear and linear constraints but their orthogonal projections on the subspace of nonlinear variables. The author will prefer this subspace in the description of the algorithm. Projecting $y_N \in \mathbb{R}^{n_N}$ onto N can be realized by solving the nonlinear optimization subproblem $\min_{x_N} \frac{1}{2}\|x_N - y_N\|^2$ s.t. $x_N \in N$. Projecting $y_N \in \mathbb{R}^{n_N}$ onto L can be realized by solving the quadratic optimization subproblem $\min_x \frac{1}{2}\|x_N - y_N\|^2$ s.t. $x_N^{lo} \leq x_N \leq x_N^{up} \wedge (x_L^{lo} \leq x_L \leq x_L^{up} \wedge A({x_N}^\top, {x_L}^\top)^\top \leq b \wedge B({x_N}^\top, {x_L}^\top)^\top = d)$. Note that if the solution $({x_N^\star}^\top, {x_L^\star}^\top)^\top$ of the later subproblem satisfies $x_N^\star \in N$ then $({x_N^\star}^\top, {x_L^\star}^\top)^\top$ solves (3). We can use either the whole solution $({x_N^\star}^\top, {x_L^\star}^\top)^\top$ of this subproblem or only the vector $x_N^\star$. The former is appropriate for the users needs (the printout of final solution). The later is more suitable for the description of the algorithm. From now we we will resign from subscripts N in the names of the elements of the subspace $\mathbb{R}^{n_N}$ of nonlinear variables. The possibility of reducing the algorithm analysis a low-dimensional space $\mathbb{R}^{n_N}$ helps the algorithm efficiency, at least as long the upper level of the decomposition is considered.

In the simplest scheme, the k-th iteration of the algorithm ($k = 1, 2, \ldots$) would be described by two evaluations of points in $\mathbb{R}^{n_N}$:

$$\bar{x}^k = P_{L'^k}\breve{x}^{k-1},\ \breve{x}^k = P_{N'^k}\bar{x}^k \tag{4}$$

with $\breve{x}^0$ being the starting point. L'^k and N'^k are subsets of L and N constructed from them as

$$L'^k = \{y \in L : \forall_{j \in J^k}\ (y - \breve{x}^{j-1})^\top(\breve{x}^{j-1} - \bar{x}^{j-1}) \geq 0\} \tag{5}$$

$$N'^k = \{y \in N : \forall_{i \in I^k}\ (y - \bar{x}^i)^\top(\bar{x}^i - \breve{x}^{i-1}) \geq 0\}. \tag{6}$$

Define *geometric cuts* (g-cuts) of type A and B as inequalities

$$(y - \breve{x}^{j-1})^\top(\breve{x}^{j-1} - \bar{x}^{j-1}) \geq 0 \quad \text{(type A g-cuts or A-cuts)} \tag{7}$$

$$(y - \bar{x}^i)^\top(\bar{x}^i - \breve{x}^{i-1}) \geq 0 \quad \text{(type B g-cuts or B-cuts).} \tag{8}$$

Set J^k in (5) equals to $\emptyset$ in case of lack of type A g-cuts, to $\{k\}$ (if $k \geq 2$) or to $\emptyset$ (if $k = 1$) in case of noncumulated type A g-cuts, and to $\{2, \ldots k\}$ in case of (full[1]) cumulation of type A g-cuts. Set I^k in (6) equals to $\emptyset$ in case of lack of type B g-cuts, to $\{k\}$ (if $k \geq 2$) or to $\emptyset$ (if $k = 1$) in case of noncumulated type B g-cuts, and to $\{2, \ldots, k\}$ in case of full cumulation of type B g-cuts. Symbols "$\breve{\ }$" and "$\bar{\ }$" are mnemonic notations for (results of) projections onto the nonlinear and linear sets, respectively, perhaps narrowed by some cuts.

Geometric cuts are a standard way to improve the convergence of projection methods, which could be unacceptably slow without them.

An assumption was made that the user wants to resign either generating g-cuts of type A or of type B. Both the types of cuts have their numerical drawbacks. A-type g-cut can be a quite dense constraint in the definition of L'^k. Such a cut, in principle, introduces up to n_N extra nonzero elements in the matrix defining L'^k. Thus, it may significantly increase the matrix density, especially when n_N is not very small in comparison with n and m; cumulating the cuts magnifies this effect. The second disadvantage of type A cuts is that they may lead to subproblems with degenerated (e.g., nearly "parallel") constraints. This is especially dangerous for linear solvers (the nonlinear solver from the current version of IAC-DIDAS-N++ uses

[1] Later also a periodic cumulation will be considered.

the penalty shifting technique [18] and should tackle such degenerated constraints more easily). However, B-type g-cuts, when cumulated, may lead to an excessive relative growth of the nonlinear subproblem size when m_N is small. The choice of cuts type used should then depend on the properties of the particular problem (2).

The further description assumes the user has chosen to use only A-type cuts. Zigzagging often slows down projection algorithms. In the sequel we will measure zigzagging $Z_{(y^i)}(k,l)$ (with $k < l$) of a finite subsequence $(y^i)_{i=k}^{l}$ of a sequence $(y^i)_{i=0}^{\infty}$ of points in a Hilbert space generated by some algorithm as

$$Z_{(y^i)}(k,l) = \frac{\sum_{i=k}^{l-1} \|y^{i+1} - y^i\|}{\|y^l - y^k\|}. \tag{9}$$

Denote also $Z_{\cdot}(i) = Z_{\cdot}(0,i)$. Using a single (noncumulated) cut cannot in general prevent the algorithm from large zigzagging. An example of this is constructed in two stages. First, we need an infeasible problem for which the algorithm cycles.

Example 1. (cycling) Let us consider the algorithm with only noncumulated type A g-cuts present and any problem for which $n_N = 2$ and

$$N = \{x \in \mathbb{R}^2 : \ \|x - z\|^2 \leq 1\}, \ z = (0, \sqrt{3})^\top$$

$$L = \{x \in \mathbb{R}^2 : \ x_2 \leq 0\}$$

$$\bar{x}^1 = (-1, 0)^\top.$$

Drawing this example it is easy to verify that the algorithm produces the cyclic sequence $\bar{x}^1 = (-1,0)^\top$, $\breve{x}^1 = (-\frac{1}{2}, \frac{\sqrt{3}}{2})^\top$, $\bar{x}^2 = (1,0)^\top$, $\breve{x}^2 = (\frac{1}{2}, \frac{\sqrt{3}}{2})^\top$, $\bar{x}^3 = (-1,0)^\top, \ldots$.

Now we will augment our example by introducing an additional dimension.

Example 2. (zigzagging under noncumulated A-type g-cuts) Now let $n_N = 3$ and the algorithm as above and the problem will be such that

$$N = \{x \in \mathbb{R}^3 : \ (x_1 - z_1)^2 + (x_2 - z_2)^2 \leq 1\}, \ z = (0, \sqrt{3})^\top$$

$$L = \{x \in \mathbb{R}^3 : \ x_2 - \varepsilon x_3 \leq 0\}$$

$$\bar{x}^1 = (-1, 0, 0)^\top.$$

where $\varepsilon > 0$ is small. If ε were equal 0 the produced sequence would be same as above (if we reject the third coordinate of the generated points). If $\varepsilon > 0$ and is small, the sequence is very similar but additionally exhibits a small progress in the direction of the third axis of $\mathbb{R}^3$. This small progress combined with a fair cycling gives large zigzagging.

4 Decreasing zigzagging

Cumulating the geometric cuts in Example 2 would avert an excessive zigzagging. However, in terms of the zigzagging analysis given below there is no such a guarantee for a whole class of problems. This is because only after every second real step (after projecting onto the nonlinear set) a geometric cut is constructed based on the step made and only these cuts are memorized. If the geometric cuts were constructed and memorized with a "full frequency" (after each projection), zigzagging would grow like a square root with the number of steps (projections) done so far.

Theorem 1. *Let a sequence $(x^i)_{i=0}^n$ (where $n \geq 1$) of points in a Hilbert space satisfy the cumulated geometric cuts condition:*

$$\forall_{s,\,1\leq s\leq n-1}\,(x^s - x^{s-1})^\top (x^n - x^s) \geq 0. \tag{10}$$

Then the following assessment for the sequence zigzagging holds:

$$\frac{\sum_{i=0}^{n-1} \|x^{i+1} - x^i\|}{\|x^n - x^0\|} \leq \sqrt{n}.$$

Proof. For $n = 1$ the claim is trivial. In order to show the claim for $n > 1$, we first need to prove that

$$\sum_{i=1}^{n} \|x^i - x^{i-1}\|^2 \leq \|x^n - x^0\|^2. \tag{11}$$

We will show it using induction, i.e. by proving

$$\sum_{i=l}^{n} \|x^i - x^{i-1}\|^2 \leq \|x^n - x^{l-1}\|^2 \tag{12}$$

for l changing from n down to 1.

For $l = n$, (12) is trivial. Now assume (12) is true for $l = k$ $(1 < k \le n)$. We are going to show that it holds for $l = k - 1$. From the theorem assumption (with $s \leftarrow k - 1$)

$$(x^{k-1} - x^{k-2})^\top (x^n - x^{k-1}) \ge 0. \tag{13}$$

Thus $\|x^n - x^{k-1}\|^2 + \|x^{k-1} - x^{k-2}\|^2 \le \|x^n - x^{k-2}\|^2$ and, using (12) (with $l = k$) we obtain

$$\Big(\sum_{i=k}^{n} \|x^i - x^{i-1}\|^2\Big) + \|x^{k-1} - x^{k-2}\| \le \|x^n - x^{k-2}\|^2$$

and then

$$\sum_{i=k-1}^{n} \|x^i - x^{i-1}\|^2 \le \|x^n - x^{k-2}\|^2,$$

which is (12) with $l = k - 1$. The induction is complete and (11) is proven. Certainly

$$\Big(\sum_{i=1}^{n} \|x^i - x^{i-1}\|\Big)^2 = \Big(\sum_{i=1}^{n} \|x^i - x^{i-1}\| \cdot 1\Big)^2.$$

Using the Cauchy-Schwarz inequality we can write

$$\Big(\sum_{i=1}^{n} \|x^i - x^{i-1}\|\Big)^2 \le \Big(\sum_{i=1}^{n} \|x^i - x^{i-1}\|^2\Big) \cdot \Big(\sum_{i=1}^{n} 1^2\Big). \tag{14}$$

Now taking (11) into account we state that

$$\Big(\sum_{i=1}^{n} \|x^i - x^{i-1}\|\Big)^2 \le \|x^n - x^0\|^2 \cdot \Big(\sum_{i=1}^{n} 1^2\Big).$$

Consequently,

$$\frac{\sum_{i=0}^{n-1} \|x^{i+1} - x^i\|}{\|x^n - x^0\|} \le \sqrt{n}$$

∎

Prior to discussing the theorem applications to the convergence analysis and similar results in the literature we will show how to modify the method so it satisfies the assumptions of the theorem.

5 Anti-zigzagging cuts

The author proposes the *anti-zigzagging cut* (*z-cut*), defined as

$$(y - \bar{x}^{i-1})^\top (\bar{x}^{i-1} - \bar{x}^{i-2}) \geq 0 \tag{15}$$

where y is an independent variable. Let us consider the *standard version of the algorithm* in which such cuts are cumulated and together with noncumulated type A g-cuts are used to define the sets L'^k; in (4):

$$L'^k = \{y \in L : \ (y - \breve{x}^{k-1})^\top (\breve{x}^{k-1} - \bar{x}^{k-1}) \geq 0\} \cap \tilde{L}'^k \tag{16}$$

with

$$\tilde{L}'^k = \{y \in L : \ \forall_{i \in \{3,\dots k\}} (y - \bar{x}^{i-1})^\top (\bar{x}^{i-1} - \bar{x}^{i-2}) \geq 0\} \tag{17}$$

(unless $k = 1$ – then $L'^k = L$). Set N'^i remains equal to N.

If we use the sequence $\bar{x}^2$, $\bar{x}^3$, $\bar{x}^4$, ... as sequence $(x^i)_{i=0}^\infty$ in Theorem 1, the assumptions of the theorem will be fulfilled. z-cuts allow to decrease zigzagging and do not complicate set N. However, for stating convergence results it will be necessary to show they are valid, i.e. do not cut off any point from the solution set $S = N \cap L$. Type A or B g-cuts may be proven valid from the alternative definition (1) of projection with N or L (perhaps narrowed by earlier proper cuts) taken as A.

Any newly created z-cut is also valid since it is implied by two valid cuts: the second last type A g-cut we have constructed and a type B g-cut we might (but do not) construct. The implication is understood in the sense of the cut cumulating techniques from the works of Cegielski, Kiwiel and Shchepakin cited in this paper. In the following lemma, validity of several cuts (based on projections of a certain point p onto certain supersets of the solution set) implies validity of a constructed "surrogate cut".

Lemma 1. (adopted from [3, Remark 7 on Theorem 3]) *Let* $p, z \in \mathbb{R}^n$, $p \neq z$. *If*

(i) $\mathcal{S} = \{s^i : \ i = 1, \dots, q\}$ *is a linearly independent system,*

(ii) $\forall_{i \in \{1,\dots,q\}} \ (z - (p + s^i))^\top s^i \geq 0$,

(iii) $\operatorname{cone} \mathcal{S}$ *is obtuse,*

(iv) t *solves the system*

$$\forall_{i \in \{1,\dots,q\}} \ (s^i)^\top (t - s^i) = 0, \tag{18}$$

then $(z - (p + t))^\top t \geq 0$.

Proof. In the referenced remark we replaced inequalities of the form $(a-b)^\top c \ge c^\top c$ with the equivalent $(a-(b+c))^\top c \ge 0$, equality $(s^i)^\top t = (s^i)^\top s^i$ with the equivalent $(s^i)^\top(t-s^i) = 0$, changed the numeration of s-es and put $B = \{z\}$. ■

Often projections of different points, obtained during the algorithm course, are used instead of projections of one point p (projecting a common p is typical for feasibility problems: the projections are made onto sets of points allowed by particular problem constraints). We will not thoroughly discuss modifications of the "surrogating" technique for such a case (see e.g. [4, Section 6]). We confine ourselves to showing a trick to retain the satisfaction of assumption (ii) in Lemma 1 when we project a point $r \ne p$:

Remark 1. If $(s^i)^\top(r-p) \ge 0$ then $(z-(r+s^i))^\top s^i \ge 0$ implies $(z-(p+s^i))^\top s^i \ge 0$.

Proof. Trivial. ■

We give a tool for ensuring obtuseness of cones.

Definition 1. (see [4, Definition 5.4A]) Cone K is a *regular obtuse cone* (in $\operatorname{Lin} K$) if $K = \operatorname{cone}\mathcal{S}$ for some system $\mathcal{S} = \{s^i : \ i \in I \subset \mathbb{Z}\}$ of linearly independent vectors in $\mathbb{R}^n$ satisfying $(s^i)^\top s^j \le 0$ for all $i \ne j$, $i, j \in I$.

Lemma 2. (see [4, Theorem 5.4A]) *A regular obtuse cone $K \subset \mathbb{R}^n$ is obtuse.*

We will also need some establishments regarding angles made by the algorithm.

Lemma 3. (an angle made when returning to a closed convex set by a projection) *Let $A \subset \mathbb{R}^n$ be closed convex, $a \in A$, $b \notin A$, $c = P_A b$. Then $(c-b)^\top(b-a) \le 0$.*

Proof. Assume the contradictory $(c-b)^\top(b-a) > 0$. This inequality together with $(a-c)^\top(c-b) \ge 0$ (which follows from the alternative definition (1) of the projection of b onto A) gives $(c-b)^\top(b-c) \ge 0$ and, consequently, $c = b$, which is false since $c \in A$, $b \notin A$. ■

Lemma 4. (Lemma 3 modified for a projection onto narrowed set A) *Let $A \subset \mathbb{R}^n$ be closed convex, $a \in A$, $b \notin A$, $H = \{y \in \mathbb{R}^n : (y-b)^\top(b-a) \ge 0\}$, $H \cap A$ be nonempty, $c = P_{H\cap A} b$. Then $(c-b)^\top(b-a) = 0$.*

Proof. Assume $(c-b)^\top(b-a) \neq 0$ by contradiction. Now, since $c \in H$, $(c-b)^\top(b-a) > 0$. Thus, using Lemma 3, $c \neq P_A b$. Denote $d = P_A b$. We have $c \neq d$. To complete the contradiction we will show $c \neq P_{A\cap H} b$. Since $P_A b$ is uniquely determined $\|d-b\| < \|c-b\|$. This and the convexity of the distance function $\|\cdot - b\|$ yield

$$\forall_{p\in\overline{cd},\, p\neq c,\, p\neq d}\ \|p-b\| < \|c-b\|. \tag{19}$$

From continuity of function $(\cdot - b)^\top(b-a)$, from $(c-b)^\top(b-a) > 0$ and from $c \neq d$

$$\exists_{p^\star\in\overline{cd},\, p\neq c,\, p\neq d}\ (p^\star - b)^\top(b-a) > 0. \tag{20}$$

Such a $p^\star$ is in A by convexity of A and by $c \in A$, $d \in A$. (20) also says that $p^\star \in H$. Thus $p^\star \in A \cap H$ and $\|p^\star - b\| < \|c-b\|$ (which follows from (19)) yields the contradiction: $c \neq P_{A\cap H} b$. ■

Theorem 2. (validity of z-cuts) *Consider algorithm* (4), *where* L'^k *is defined by* (16), $N'^k = N$, *with* $\breve{x}^0 \in \mathbb{R}^{n_N}$ *being the starting point. Let* $S = N \cap L$ *be nonempty and let* $k \geq 2$. *If* $\bar{x}^{k-1} \notin N$ *and* $\breve{x}^{k-2} \notin L$ (*a solution not found yet*) *then* $S \subset L'^k$.

Proof. We prove the Theorem by induction. The inclusion $S \subset L'^2$ is obvious since $\tilde{L}'^2 = L$. Suppose that $S \subset L'^{k-1}$ for some k, $k \geq 3$. It is enough to prove that

$$\forall_{z\in S}\ (z - \bar{x}^{k-1})^\top(\bar{x}^{k-1} - \bar{x}^{k-2}) \geq 0, \tag{21}$$

which we do in two steps:

1. From the assumption, $\breve{x}^{k-2} \notin \tilde{L}'^{k-1}$. The assumptions of Lemma 4 are fulfilled with $A \leftarrow \tilde{L}'^{k-1}$, $b \leftarrow \breve{x}^{k-2}$, $c \leftarrow \bar{x}^{k-1}$. The lemma yields

$$(\bar{x}^{k-1} - \breve{x}^{k-2})^\top(\breve{x}^{k-2} - \bar{x}^{k-2}) = 0. \tag{22}$$

2. Now apply Lemma 1 with $z \in S$, $p \leftarrow \bar{x}^{k-2}$ $q \leftarrow 2$, $s^1 \leftarrow \breve{x}^{k-2} - \bar{x}^{k-2}$, $s^2 \leftarrow \bar{x}^{k-1} - \breve{x}^{k-2}$ after the following validation of the assumptions fulfillment:
   (i) by (22) and inequalities $\bar{x}^{k-1} \neq \breve{x}^{k-2}$, $\breve{x}^{k-2} \neq \bar{x}^{k-2}$ (obtained from $\bar{x}^{k-1} \notin N$, $\bar{x}^{k-2} \notin N$ and – what follows from $k \geq 3$ – $\breve{x}^{k-2} \in N$).
   (ii) – Case of $i = 1$ – from the alternative definition (1) of projection of $\bar{x}^{k-2}$ onto set N, being a superset of S.

– Case of $i = 2$. We have $(z - \bar{x}^{k-1})^\top(\bar{x}^{k-1} - \breve{x}^{k-2}) \geq 0$ by (1) applied to projection of $\breve{x}^{k-2}$ onto $\{v \in \tilde{L}'^{k-1} : (v - \breve{x}^{k-2})^\top(\breve{x}^{k-2} - \bar{x}^{k-2}) \geq 0\}$, being a superset of S (by the induction assumption and by (1) applied to the projection $\bar{x}^{k-2}$ onto N). Now apply Remark 1 with $p \leftarrow \bar{x}^{k-2}$, $r \leftarrow \breve{x}^{k-2}$, using (22).

(iii) from (22), the linear independence of vectors s^1, s^2 (established in (i)), Definition 1, Lemma 2

(iv) easy to verify using (22). ■

Remark 2. Cumulating both $z - cuts$ and $A - cuts$ does not influence the proof of z-cuts validity if we redefine $\tilde{L}'^k$ as $\{y \in L :\ \forall i \in \{2, \dots, k-1\}$ $(y - \breve{x}^{i-1})^\top(\breve{x}^{i-1} - \bar{x}^{i-2}) \geq 0 \ \wedge\ \forall_{j \in \{3, \dots k\}}(y - \bar{x}^{j-1})^\top(\bar{x}^{j-1} - \bar{x}^{j-2}) \geq 0\}$ so it accounts for all the cuts constructed so far except of the last A-cut.

The reason for introducing z-cuts can be described as "spoiling" influence of projections onto N on the zigzagging reduction by cumulated g-cuts. A similar phenomenon of spoiling effect of some projections was discussed in [3, Section 5]. The remedy applied in this work bases also on Lemma 1, but the retour to a set is done without a narrowing cut.

6 The algorithm for the feasibility problem

The algorithm for problem (3) will be described in terms of the space of nonlinear variables.

The algorithm
Parameters: tolerance $t^N \geq 0$, cuts existence and cumulation options, cumulation period[2] $T \in [3, \infty]$, starting point $\breve{x}^0$. We initiate the iteration counter k to 1.

1. $\bar{x}^k = P_{L'^k}\breve{x}^{k-1}$, where

$$L'^k = \{y \in L : (\forall_{p \in J^k}\ (y - \breve{x}^{p-1})^\top(\breve{x}^{p-1} - \bar{x}^{p-1}) \geq 0) \\ \wedge (\forall_{j \in K^k}\ (y - \bar{x}^{j-1})^\top(\bar{x}^{j-1} - \bar{x}^{j-2}) \geq 0)\},$$

where J^k equals to $\emptyset$ in case of lack of type A g-cuts, to $\{k\}$ (if $k \geq i+2$) or to $\emptyset$ (if $k = i+1$) in case of noncumulated type A g-cuts, and to

[2]The word "period" will refer either to an epoche of the algorithm run or to the length of the epoches, depending on the context.

$\{i+2,\dots k\}$ in case of cumulation of type A g-cuts; K^k equals to $\emptyset$ in case of lack of z-cuts, to $\{k\}$ (if $k \geq i+3$) or to $\emptyset$ (otherwise) in case of noncumulated z-cuts, and to $\{i+3,\dots k\}$ in case of cumulation of z-cuts. Here i is the number of the last iteration in the previous period (or 0 if we are in the first period), i.e $i = k - (k \bmod T)$. If $L'^k = \emptyset$ STOP – report infeasibility.

2. $\breve{x}^k = P_N \bar{x}^k$. If $N = \emptyset$ STOP – report infeasibility. If $\|\breve{x}^k - \bar{x}^k\| \leq t^N$ STOP – return the last solution of the quadratic nonlinear problem as the solution. Otherwise, set $k := k+1$, go to 1.

Remark 3. If $T < \infty$ then Theorem 2 should be used for each period of the algorithm course separately; the iteration numbers in each period should be shifted to start with 1.

Remark 4. The inner procedure was given under the assumption of nonexistence of type B g-cuts and existence of type A g-cuts; the transformation to the reverse assumption (nonexistence of type A g-cuts and existence of type B g-cuts) is trivial; one should, however, remember that the definitions of sets L and N should remain the same and the distance made during projecting on L should be then used in the stopping criterion.

Later we will also consider a modification of the algorithm:

Modification. (early point projection) In step 1 of the algorithm for feasibility problem, if $k \bmod T = 0$, evaluate $\bar{x}^k$ as $\bar{x}^k = P_{L'^k} x^{k-T+1}$.

The term "standard version of the algorithm" denotes the variant with noncumulated A-cuts, cumulated z-cuts, with the Modification not applied.

7 Convergence

In this section we shall by default refer to the standard version of the algorithm described in Section 6, with $t^N = 0$, $T = \infty$.

We shall first define a property of our feasibility problem saying that if we are close to both the sets N and L, we are close to their intersection.

Definition 2. (Bounded regularity, adopted from [2, Definition 5]) A feasible problem (3) or the pair of its sets L, N is *boundedly regular* if for each bounded $G \subset \mathbb{R}^{n_N}$

$$\forall_{\varepsilon>0} \exists_{\delta>0} \forall_{x\in G} \max(\operatorname{dist}(x,L), \operatorname{dist}(x,N)) < \delta \Rightarrow \operatorname{dist}(x, N\cap L) < \varepsilon. \tag{23}$$

Remark 5. For a boundedly regular problem and for each bounded $G \subset \mathbb{R}^{n_N}$ there exist a *regularity function* $\phi_G : \mathbb{R}^+ \setminus \{0\} \to \mathbb{R}^+ \setminus \{0\}$ that satisfies

$$\begin{aligned}&\operatorname{dist}(x, N \cap L) \geq \varepsilon \Rightarrow\\&((x \in N \Rightarrow \operatorname{dist}(x, L) \geq \phi_G(\varepsilon)) \wedge (x \in L \Rightarrow \operatorname{dist}(x, N) \geq \phi_G(\varepsilon)))\end{aligned}$$

for each $x \in G$.

Indeed: Fix G. Take $\phi_G(\varepsilon) = \delta$ with δ satisfying

$$\forall_{x \in G} \max(\operatorname{dist}(x, L), \operatorname{dist}(x, N)) < \delta \Rightarrow \operatorname{dist}(x, N \cap L) < \varepsilon.$$

It is easy to prove that ϕ_G is a regularity function by using the following fact:

$$x \in L \Rightarrow \max(\operatorname{dist}(x, L), \operatorname{dist}(x, N)) = \operatorname{dist}(x, N)$$

and

$$x \in N \Rightarrow \max(\operatorname{dist}(x, L), \operatorname{dist}(x, N)) = \operatorname{dist}(x, L).$$

We will often write ϕ instead of ϕ_G. The explicit definition of G may be suppressed, since often some bounded set containing all the points generated by the algorithm can be easily shown (e.g., L may be taken if the bound constraints on all the variables in the problem are finite). The regularity function allows assessing the distances from N and L but this implies an assessment for lengths of variously defined steps of our algorithms, e.g. $\|\bar{x}^i - \breve{x}^{i-1}\|$ or $\|\bar{x}^i - \bar{x}^{i-1}\|$, even though we project onto L'^k, not onto L:

Remark 6. Consider points $\bar{x}^{i-1}$, $\breve{x}^{i-1}$, $\bar{x}^i$, $\breve{x}^i$ generated by the algorithm from Section 6 with any cutting scheme, with any $T \in [3, \infty]$ and any $t^N \geq 0$ and assume that the Modification was not applied or $i \bmod T \neq 0$. The following statements are true:

a) If $S \neq \emptyset$ then $\|\bar{x}^i - \breve{x}^{i-1}\| \geq \phi(\operatorname{dist}(\breve{x}^{i-1}, L \cap N))$ and $\|\breve{x}^i - \bar{x}^i\| \geq \phi(\operatorname{dist}(\bar{x}^i, N \cap L))$ from $L'^i \subset L$, $N'^i \subset N$ and the definition of orthogonal projection.

b) If an A-cut $(y - \breve{x}^{i-1})^\top(\breve{x}^{i-1} - \bar{x}^{i-1}) \geq 0$ took part in the construction of L'^i then $\|\bar{x}^i - \bar{x}^{i-1}\| \geq \|\bar{x}^i - \breve{x}^{i-1}\|$ from the cosine theorem.

Fact 1. (see [2, Proposition 5.4 (iii)]) As L and N are closed convex subsets of the finite-dimensional space $\mathbb{R}^{n_N}$, bounded regularity holds.

Definition 3. (compare Definition 5.6 in [2]) A feasible problem (3) or the pair L, N of its sets is *boundedly linearly regular* if for each bounded $G \subset \mathbb{R}^{n_N}$ we can give $\kappa(G) > 0$ such that

$$\forall_{x \in G} \mathrm{dist}(x, N \cap L) \leq \kappa(G) \max(\mathrm{dist}(x, L), \mathrm{dist}(x, N)).$$

Remark 7. If κ is such as in Definition 3 then the functions $\phi_G(x) = \kappa(G) \cdot x$ are regularity functions for this problem.

Fact 2. If L, N bounded then $L \cap \mathrm{int} N \neq \emptyset$ implies bounded linear regularity of the pair L, N.

Proof. See Corollary 5.14 in [2]. ■

By an absence of equality constraints[3], $L \cap \mathrm{int} N \neq \emptyset$ is likely whenever the current approximated optimal value Q of the optimization problem is set still too big. For a one-element intersection it might happen that there is no linear ϕ, which the reader can easily check for $n_N = 2$, $L = \{(x_1, x_2)^\top : x_2 \leq 0\}$, $N = \{(x_1, x_2)^\top : x_2 \geq x_1^2\}$.

A more exhaustive discussion of conditions for regularities of the problems is given in [2, Section 5].

Several folklore properties of the projection methods will be quoted in the sequel.

Definition 4. A finite or infinite sequence (x^i) of points in a Hilbert space H has the *Fejér contraction property* w.r.t. $C \subset H$ if

$$\|x^i - c\|^2 \geq \|x^{i+1} - c\|^2 + \|x^{i+1} - x^i\|^2 \tag{24}$$

for each $c \in C$. Similarly, operator $O : H \to H$ has this property if for each $c \in C$ and $x \in H$ $\|x - c\|^2 \geq \|Ox - c\|^2 + \|Ox - x\|^2$.

Fact 3. Projection onto a closed convex set has the F.c.p. w.r. to this set and, consequently, to each of its subsets.

Proof. See calculations in [10] on page 228 with $t_{\min} = t_{\max} = 1$. ■

Remark 8. For sequences or operators, Fejér contraction property w.r.t. C implies *Fejér monotonicity* w.r.t. C: $\|x^i - c\| \geq \|x^{i+1} - c\|$ for all $c \in C$ or, respectively, $\|x - c\| \geq \|Ox - c\|$ for all $c \in C$, $x \in H$.

[3] In case of presense of equality constraints an analysis would involve relative interiors and essential cores of sets – see [2, Section 5].

Remark 9. Sequence (x^i) having Fejér contraction property w.r.t. $S \subset \mathbb{R}^{n_N}$ and with $\|x^{i+1} - x^i\| \geq \kappa \mathrm{dist}(x^i, S)$, exhibits an at least $\sqrt{1-\kappa^2}$-linearly decreasing distance from S.

Indeed: Take $c = P_S x^i$. Now $\|x^i - c\| = \mathrm{dist}(x^i, S)$ and thus the Fejér contraction property yields $\mathrm{dist}^2(x^i, S) \geq \|x^{i+1} - c\|^2 + \|x^{i+1} - x^i\|^2$. Hence $\mathrm{dist}^2(x^i, S) \geq \|x^{i+1} - c\|^2 + \kappa^2 \mathrm{dist}^2(x^i, S)$. This implies $\mathrm{dist}^2(x^i, S) \geq \mathrm{dist}^2(x^{i+1}, S) + \kappa^2 \mathrm{dist}^2(x^i, S)$. Now $\mathrm{dist}(x^{i+1}, S) \leq \sqrt{1-\kappa^2}\mathrm{dist}(x^i, S)$.

Taking Remark 6a) into account, we may prove in this way a linear convergence for our algorithm with any cutting scheme for linearly regular problems: we must take sequence[4] $\bar{x}^1$, $\breve{x}^1$, $\bar{x}^2$, $\breve{x}^2$, ... as (x^i). The result is known, though often inexact projections are considered [2, 13] and thus an additional property of *focusing* for algorithms is introduced in order to assure a linear convergence.

Assume now that for some reason $\bar{x}^i \in A$ for a bounded $A \subset \mathbb{R}^{n_N}$ (as already pointed out, $A = L$ may be often taken).

Consequence 1. (of Theorem 1) Let $\phi = \phi_A$ be a regularity function for the set pair (N, L) and let $S \neq \emptyset$, $T = \infty$, $t^N = 0$. Then the number r of iterations needed by the (standard version of) the algorithm to achieve error ϵ (i.e. first r satisfying $\mathrm{dist}(\bar{x}^r, S) \leq \epsilon$) is not greater than

$$\left(\frac{\mathrm{diam}(A)}{\phi(\epsilon)}\right)^2 + 3. \tag{25}$$

Proof. The algorithm cannot terminate with reporting infeasibility, since neither N nor L'^k generated in step 1 are empty (from $S \neq \emptyset$, $N \supset S$ and Theorem 2). Assume now we reached r greater than (25) and we have not attained error ϵ. Then by the definition of regularity function and by Remark 6a), b) $\|\bar{x}^{i+1} - \bar{x}^i\| \geq \phi(\epsilon)$ for $i \geq 1$. Thus $\|\bar{x}^3 - \bar{x}^r\| \geq \frac{1}{Z_{(\bar{x})^i}(3,r)}(r-3)\phi(\epsilon)$ with $Z_{(\bar{x})^i}$ given by (9). As the z-cut construction starts from the 3rd iteration, we have, using Theorem 1 with a suitable index shifting, $\|\bar{x}^3 - \bar{x}^r\| \geq \frac{1}{\sqrt{r-3}}(r-3)\phi(\epsilon)$, so $\|\bar{x}^3 - \bar{x}^r\| \geq \sqrt{r-3}\phi(\epsilon)$ and, by $r > \left(\frac{\mathrm{diam}(A)}{\phi(\epsilon)}\right)^2 + 3$, $\|\bar{x}^3 - \bar{x}^r\| > \mathrm{diam}(A)$ (nonsense). ∎

In an outer loop of the algorithm, the detection of infeasibility will also take place.

[4]If A-cuts are present, we may take sequence $\bar{x}^1$, $\bar{x}^2$, $\bar{x}^3$, ... and use also Remark 6b).

Consequence 2. (of Theorem 1) In case of infeasibility ($\mathrm{dist}(L,N) = d > 0$) the standard algorithm version with $T = \infty$ and $t^N = 0$ needs at most

$$r = \left(\frac{\mathrm{diam}(A)}{d}\right)^2 + 3 \tag{26}$$

steps to detect infeasibility.

Proof. Assume we have reached a greater r and the infeasibility has not been detected. By Remark 6b), $\|\bar{x}^{i+1} - \bar{x}^i\| \geq d$ for $i \geq 1$. Now we repeat the reasoning from the proof of Consequence 1, using d instead of $\phi(\epsilon)$. ■

Remark 10. It seems reasonable to cumulate both z-cuts and (type A) g-cuts in order to quickly detect infeasibility. As every cut that have ever been used is then cumulated, the problem is – up to the moment of detection of infeasibility – not distinguishable (under the used oracle) from some feasible problem[5]. Therefore any good convergence properties of feasible problems apply and force the algorithm to quickly converge to the imagined solution (which is impossible) or to detect the infeasibility. It is difficult to calculate a regularity function for the "imagined" feasible problem, but the step lengths still can be assessed as minorized by d.

Remark 11. Consider the standard version of the algorithm with $T = \infty$, $t^N = 0$. A fulfillment of the condition $\sum_{j=1}^{i} \|\bar{x}^{j+1} - \bar{x}^j\|^2 > \mathrm{diam}^2(L)$, with i being the current iteration number, also can indicate infeasibility, like in [7], since for any feasible problem $\sum_{j=1}^{i} \|\bar{x}^{j+1} - \bar{x}^j\|^2 \leq \mathrm{dist}^2(\bar{x}^1, S) - \mathrm{dist}^2(x^{i+1}, S)$ holds by Remark 6 a), b) and by iterated Fejér contraction property, causing a contradictory: $\mathrm{diam}^2(L) > \sum_{j=1}^{i} \|\bar{x}^{j+1} - \bar{x}^j\|^2$. This would give a similar (inverse-quadratic in d) infeasibility detection speed estimate as (26). However, infeasibility of a subproblem created using cumulated cuts seems to appear sooner than the satisfaction of the condition from [7], as shown experimentally in [5] for the method from [3].

The Consequence 1 of Theorem 1 is a bit wasteful since it uses ϕ of the final error, a small number, to assess the lengths of all steps. It would seem more economic to determine a relation between the number of steps and a relatively low decrease in error (in some later analysis the consecutive error decreases could be gathered):

[5] All we can say is that $N \cap L$, if nonempty, must be a subset of the current L'^i.

Remark 12. (on Consequence 1 of Theorem 1) Assume additionally the uniqness of solution: $S = \{s\}$. The iterates following the current one x^p are in the ball B with center in s and radius $\mathrm{dist}(x^j, S)$, due to Fejér monotonicity. Then, by a similar argument[6], to decrease the error from $\epsilon_I (= \mathrm{dist}(\bar{x}^p, S))$ to ϵ_{II} we need at most

$$\left(\frac{\mathrm{diam}(B)}{\phi(\epsilon_{II})}\right)^2 = \left(\frac{2\epsilon_I}{\phi(\epsilon_{II})}\right)^2$$

iterations[7], assumed that $p \geq 3$.

Cut cumulation techniques were considered by several authors. Cegielski proposes in [4] a cone method for feasibility problems and for subgradient methods; the application of the later one is described in [3]. Similar approaches, expressed in a more algebraic form were studied in [8], another conical method was also proposed in [17]. Cegielski chooses from several projection vectors a linearly independent system that – like in Lemma 1 – spans an obtuse cone. Projection onto the area allowed by the cuts corresponding to the projections is realized by solving a low-dimensional linear system (18) (or similar) with a Gram matrix. Kiwiel in [10] projects on his surrogate cut by solving an auxiliary quadratic programming problem (note that we also use quadratic programming). Solving small auxiliary problems is perhaps easier than augmenting an existing one (this happens if the solving time grows faster than linearly with the size of problem; though the particular sizes used in both constructions should be examined more thoroughly). We also do not filter out "nonvaluable" cuts. However, since we already use a large-scale solver, the complication of the big problem by additional cuts is not necessarily essential. We can tune this complication by regulating T. Once we agreed to use auxiliary solvers, our algorithm enjoys a bigger implementational simplicity.

Both the constructions of cited authors serve for producing long steps since such cause a big decrease in the error square, by Fejér contraction property. In the described works, the obtained lengths are compared to the lenghts for a nonaccelerated version of the algorithm (c.f. [4, Section 5.2]). Let us perform a similar analysis. Consider the standard version of the algorithm with $T < \infty$, with the Modification. Assume $k \bmod T = 1$ and $(k + l + 1) \bmod T = 0$ and $l < T$ (so k and $k + l + 1$ belong to the same

---

[6] i.e. a contradiction saying that $\|x^p - x^r\| > \mathrm{diam}(B)$, where $r > p + \left(\frac{\mathrm{diam}(B)}{\phi(\epsilon_{II})}\right)^2$.

[7] $\phi = \phi_B$ may be taken.

period). Without z-cuts, the decrease of $\text{dist}^2(\bar{x}^k, S) - \text{dist}^2(\bar{x}^{k+l+1}, S)$ in error square between the k-th and $(k+l+1)$-th iterate would be at least $\sum_{j=1}^{l+1} \|\bar{x}^{k+j} - \bar{x}^{k+j-1}\|^2$ by Fejér contraction property (24) used $l+1$ times. If now the step from x^k to x^{k+l+1} is advantageous or indifferent in the sense of the Fejér- contraction-estimate of decrease in error square, the following must hold:

$$\sum_{j=1}^{l+1} \|\bar{x}^{k+j} - \bar{x}^{k+j-1}\|^2 \leq \|\bar{x}^k - \bar{x}^{k+l+1}\|^2. \tag{27}$$

In our algorithm it holds simply due to (11) when sequence $(\bar{x}^i)_{i=k}^{k+l+1}$ satisfies the cumulated cuts condition. Both cited authors similarly assure a gain from their "long steps", though instead of previous steps $\bar{x}^{k+j} - \bar{x}^{k+j-1}$ $(j \leq l)$, vectors of projections on the sets defining the feasibility problem can also be used. In the conical construction such an assurance is based on the properties of constructed cones (acute/obtuse) and was the reason to introduce such cones.

In singular cases, like $(\bar{x}^k) = (1, 0, 0, \ldots, 0)^\top$, $(\bar{x}^{k+1}) = (0, 1, 0, \ldots, 0)^\top$, $\ldots$, (27) may hold as an equality, giving no gain. This brings a discomfort but similar cases are also possible in the cited propositions. In the conical method, it may happen when the configuration of projection vectors does not allow to choose a subsystem generating an obtuse cone, except of a one-element system or when the projection vectors are perpendicular each to other. In the method of surrogate constraints, such a degenerated case is caused by intersecting of particular hyperplanes (note that in our algorithm the z- cut hyperplanes intersect in $\bar{x}^{k+l+1}$ in the singular case).

The zigzagging bound in Theorem 1, which is of the type of a square function, turns out to be still a bit too large then. The author refers to this question in Section 9

8 Preliminary experiments

The algorithm was implemented in C++; it uses the HOPDM linear-quadratic solver; nonlinear subproblems were solved analytically in experiments below.

Generated *test problems* of the form (3) try to conceive several demands: a faily random generation, sparsity of the linear constraints, a possibility of shape (conditioning) tuning, the simplicity of (analytical) generation, and the existence of only one known solution point. A problem is constructed in several stages, described by complicated formulae; it takes n_N, n, m, $W \in [1, \infty)$, $V \in [1, \infty)$ $X \in [1, \infty)$ as parameters and basically consists of:

1. One nonlinear constraint g_1 resulting in set N being an ellipsoid with the maximum axes lengths ratio of V.
2. m linear inequality constraints:
   (a) $2n - n_N$ ones making only one point feasible. Their hyperplanes are almost orthogonal to the gradient of the nonlinear constraint at the feasible point (in order to make essential the constraints from the next group).
   (b) $m-(2n-n_N)$ ones - their hyperplanes were randomly generated (so as to retain the feasibility of the solution point) and then $1/W$ times dilated in the direction of the gradient of the nonlinear constraint.
3. Bounds.

Such a problem (actually – its sets N and L) is next dilated X times in the direction of the first axis in order to additionally deteriorate its conditioning.

A part of the results is presented in Table 1. The observed quantities are: $Z_p = Z_p(i) = Z_{(\bar{x}_N^j)}(i, i-p)$ – zigzagging of the fragment of p last segments of the trajectory, $\rho = \rho(i) = \|\bar{x}^i - \bar{x}^{i-1}\|$ – the step lengths (in terms of $\mathbb{R}^{n_N}$), $d = d(i)$ – distance of $({x_N^\star}^\top, {x_L^\star}^\top)^\top$ (the solution of the quadratic subproblem whose n_N first coordinates form $\bar{x}^i$) from the solution in terms of whole space $\mathbb{R}^n$.

The experiments are commented in Section 9

9 Conclusions and further work

The first experiments on the algorithm seem rather promising. The algorithm reaches a zigzagging lower than the theoretical $\sqrt{5} \approx 2.236$ while a decrease in step lengths ρ during the 5 zigzagging-measurement steps is low. In other experiments the algorithm seems not very much dependent on tuning T. The size n_N which is most important for the behavior of the sequential projections loop can reach the rank of hundreds. The remaining sizes determine mainly the difficulty of the quadratic optimization subproblem and their limits depend on the properties of the quadratic solver. Some observed accuracy problems are natural in a decomposition scheme. The admissible sets of these subproblems might have been very "flat" (have almost no interior), especially when the regularity is not linear and we are close to the solution. However, the method was intended to quickly converge into a neighborhood of the solution. Then one of the many methods of good

i	no cuts	g nonc			g nonc + z cumm				g cumm + z cumm		
	d_N	d_N	Z_5	ρ	d	d_N	Z_5	ρ	d_N	Z_5	ρ
0	13,31	13,31		68,598	19,01	13,31		68,598	13,31		68,598
1	12,57	12,26		3,030	17,94	12,26		3,030	12,26		3,030
2	12,37	11,52		1,411	16,24	11,51		1,411	11,52		1,411
3	12,26	11,12		1,154	15,33	11,11		1,153	11,10		1,155
4	12,19	11,03		0,682	14,09	10,56		1,007	10,35		1,222
5	12,13	10,20		1,596	13,72	10,28		0,831	9,83		1,008
6	12,08	10,13	1,93	0,722	13,09	9,83	1,64	0,857	9,31	1,57	0,961
7	12,04	9,97	2,45	1,052	12,02	9,24	1,70	0,930	8,29	1,63	1,471
8	12,00	9,91	3,03	0,595	11,86	8,93	1,67	0,650	7,90	1,57	0,811
9	11,97	9,59	2,67	1,149	11,49	8,44	1,71	0,752	7,46	1,60	0,761
10	11,93	9,53	3,92	0,650	10,97	8,08	1,60	0,645	7,12	1,60	0,776
11	11,91	9,39	3,44	0,980	10,08	7,70	1,56	0,629	6,53	1,55	0,922
12	11,88	9,34	4,54	0,554	9,62	7,38	1,59	0,566	6,16	1,60	0,667
13	11,85	9,10	3,33	1,039	9,09	7,01	1,51	0,577	5,82	1,63	0,674
14	11,83	9,05	4,36	0,590	8,82	6,68	1,52	0,554	5,46	1,68	0,703
15	11,80	8,90	3,50	0,964	8,03	6,35	1,50	0,507	5,03	1,64	0,742
16	11,78	8,85	4,80	0,540	7,60	6,03	1,49	0,549	4,69	1,77	0,723
17	11,76	8,72	4,01	0,788	7,07	5,70	1,48	0,475	4,40	1,81	0,661
18	11,74	8,68	4,09	0,505	6,91	5,46	1,57	0,557	4,16	1,83	0,521
19	11,72	8,42	2,91	1,088	6,65	5,13	1,57	0,516	3,73	1,72	0,639
20	11,70	8,36	5,10	0,582	6,45	4,91	1,67	0,425	3,51	1,77	0,428
21	11,69	8,27	4,32	0,710	5,75	4,73	1,71	0,403	3,22	1,82	0,573
22	11,67	8,24	4,42	0,460	5,34	4,40	1,75	0,495	2,87	1,65	0,631
23	11,65	8,09	3,62	0,804	5,04	4,17	1,62	0,380	2,46	1,55	0,545
24	11,64	8,05	3,87	0,509	4,66	3,97	1,68	0,390	2,13	1,57	0,460
25	11,62	7,87	3,16	0,907	4,50	3,75	1,68	0,356	1,90	1,56	0,394
26	11,61	7,83	5,41	0,504	4,31	3,47	1,54	0,421	1,71	1,56	0,391
27	11,59	7,64	3,54	0,894	4,07	3,32	1,62	0,274	1,47	1,47	0,377
28	11,58	7,59	4,17	0,502	3,81	3,03	1,56	0,464	1,35	1,52	0,224
29	11,56	7,49	4,08	0,712	3,49	2,84	1,54	0,292	1,11	1,65	0,383
30	11,55	7,46	4,05	0,447	3,64	2,84	1,50	0,002	0,92	1,65	0,292
31	11,53	7,31	3,51	0,744	3,37	2,79	1,95	0,381	0,90	1,64	0,115
32	11,52	7,28	4,60	0,453	2,46	2,21	1,53	0,804	0,55	1,56	0,473
33	11,51	7,16	3,32	0,812	2,39	2,10	1,57	0,254	0,38	1,51	0,243
34	11,50	7,12	5,59	0,453	2,20	1,97	1,83	0,336	0,31	1,51	0,095
35	11,48	6,93	3,41	0,867	2,21	1,82	1,88	0,277	0,09	1,38	0,239
36	11,47	6,89	4,56	0,475	1,96	1,65	1,65	0,279	0,08	1,29	0,030

Table 1. Algorithm course in initial stages for various cutting schemes. $n_N = 30$, $n = 100$, $m = 1000$, $X = 5$, $V = 15$, $W = 25$, $T = 30$, Modification – not applied.

asymptotic properties, like superlinear convergence, can overtake the optimization. Relaxation of z-cuts by parallelly shifting their hyperplanes often turned out to overcome the numerical problems.

The author is now conducting his works on equipping the algorithm with an outer loop based on the level control [7] with upper bounds on Q obtained by examining $f(\bar{x}^i)$.

Also, the author hopes for improving the theoretical zigzagging limit below the square-root dependency. This would make cut cumulation a stronger tool (in the worst-case sense) than the Fejér monotonicity. Moreover, the inverse-quadratic efficiency (from Consequence 1 by linear regularity) would be surpassed. The impossibility of exceeding such an efficiency, obtained in [15] for subgradient methods in convex minimization is often cited in works in which a cut cumulation is used [3, 9]. However, our problem differs from that of Nemirovskii and Yudin.[8]

The way to realize the improvement in a finite-dimensional case consist in strengthening the assumption (10) in Theorem 1 to the form

$$\forall_{s,\,1\le s\le n-1}\,\forall_{r,s<r\le n}(x^s - x^{s-1})^\top(x^r - x^s) \ge 0, \tag{28}$$

which is certainly fulfilled in the standard algorithm version, since any created z-cut affects the generation of all the subsequent iterates, not only x^n. It is easy to show a sequence in Hilbert space l^2 that fulfills both (10) and (28) and reaches the square-root zigzagging: it is the sequence $x^0 = (1,0,0,\ldots)^\top$, $x^1 = (1,1,0,0,\ldots)^\top$, $x^2 = (1,1,1,0,0,\ldots)^\top, \ldots$. It is interesting that the author has constructed a sequence of points in $\mathbb{R}^2$ reaching the maximal zigzagging that satisfies (10). It seems to be a unique sequence with these two properties and it turns out not to satisfy (28).

References

[1] A. Altman and J. Gondzio, *Regularized Symmetric Indefinite Systems in Interior Point Methods for Linear and Quadratic Optimization*, Optimization Methods and Software **11-12** (2000), 275–302.

[2] H. Bauschke and J. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Review **38** (3) (1996), 367–426.

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

[4] A. Cegielski, Relaxation methods in Convex Optimization Problems, Higher College of Engineering, Series Monographies, No. 67, Zielona Góra, Poland (in Polish).

[8] Unlike them, we can have a quasiconvex goal function. We do not use Lipschiz constants or the normalization assumption of Nemirovskii and Yudin; instead, we use the regularity function. We measure the error differently. Unlike in the proof of the discussed theoretical complexity – we consider a finite-dimensional case.

[5] R. Dylewski, *Numerical behavior of the method of projection onto an acute cone with level control in convex optimization*, Discuss. Math. Differential Inclusions, Control and Optimization **20** (2000), 147–158.

[6] S. Flåm and J. Zowe, *Relaxed outer projections, weighted averages and convex feasibility*, BIT **30** (1990), 289–300.

[7] S. Kim, A. Hyunsil and C. Seong-Cheol, *Variable target value subgradient method*, Math. Progr. **49** (1991), 356–369.

[8] K. Kiwiel, *Monotone Gram matrices and deepest surrogate inequalities in accelerated relaxation methods for convex feasibility problems*, Linear Algebra and its Applications **215** (1997), 27–33.

[9] K. Kiwiel, *The efficiency of subgradient projection methods for convex optimization, Part I: General level methods*, SIAM Control Optim. **34** (2) (1996), 660–676.

[10] K. Kiwiel, *Block-Iterative Surrogate Projection Methods for Convex Feasibility Problems*, Linear Algebra and its Applications **15** (1995), 225–259.

[11] T. Kręglewski, J. Granat and A. Wierzbicki, *IAC-DIDAS-N – A Dynamic Interactive Decision Analysis and Support System for Multicriteria Analysis of Nonlinear Models*, v. 4.0, Collaborative Paper, CP-91-101, International Institute for Applied Systems Analysis, Laxenburg, Austria, June 1991.

[12] C. Lemaréchal, A.S. Nemirovskii and Yu. Nesterov, *New variants of bundle methods*, Math. Progr. **69** (1995), 111–147.

[13] B. Łopuch, Projection methods with an aggregation for convex feasibility problems, Doctoral thesis, Institute of Systems Research, Warsaw 1997 (in Polish).

[14] M. Makowski, LP-DIT data interchange tool for linear programming problems (version 1.20), Working Paper, WP-94-36, International Institute for Applied Systems Analysis, Laxenburg, Austria 1994.

[15] A.S. Nemirovskii and D. Yudin, Optimization Problem Complexity and Method Efficiency, Nauka, Moscow 1979 (in Russian).

[16] T. Polyak, *Minimization of unsmooth functionals*, Zh. Vychiisl. Mat. Fiz. **9** (1969), 14–29 (in Russian).

[17] M. Shchepakin, *On a modification of a class of algorithms for mathematical programming*, Zh. Vychisl. Mat. i Mat. Fiz. **19** (1979), 1387–1395 (in Russian).

[18] A. Wierzbicki, *A Penalty Function Shifting Method in Constrained Static Optimization and its Convergence Properties*, Archiwum Automatyki i Telemechaniki **XVI** (4) (1971), 396–416.

Received 17 November 1999
Revised 15 October 2000