

Some variants of projection methods for large nonlinear optimization problems

Paweł Białoń

Abstract — Two ideas of modifying projection methods for the case of smooth nonlinear optimization are presented. Projection methods were originally successfully used in solving large-scale linear feasibility problems. The proposed instantiations of projection methods fall into two groups. One of them is a decomposition approach in which projections onto sets are realized as optimization problems which themselves involve much portions of original problem constraints. There are two subproblems: one build with linear constraints of the original problem and the second one build with original nonlinear constraints. These approaches use special accelerating cuts so that the separation of nonlinear and linear constraints can be effective and some problem sparsity preserved. The second group bases on penalty-shifting/multiplier methods and draws from the observation that unconstrained subproblems obtained there may solve very slowly due to their nonsmooth character. Thus it is proposed to solve them with modified projection methods which inherit from conjugate gradient methods a multi-dimensional subspace in one epoche.

Keywords — *projection methods, penalty shifting method, nonlinear optimization.*

1. Introduction

Projection methods [2, 10] in their classical form serve for solving convex feasibility problems, i.e, the problems of finding a common point of several closed convex sets. They received some success, especially in image reconstruction for medical applications (see references in [2]), where problems of million sizes have been solved favourably for some time. There has been a considerable stream of research on adopting projection methods for optimization problems, which differ from feasibility problems only by the existence of a goal function. However, usually the investigations surround themselves with nondifferentiable optimization [13, 19], whereas the author sees some possibilities to apply them in more “standard” branches of optimization, i.e. nonlinear smooth, possibly large-scale optimization. Despite their success in solving large feasibility problems, projection methods have other features that seem appealing for such a usage. They do not have to involve any complex linear algebra in the case where the sets in feasibility problems are halfspaces (which means that the feasibility problem is linear). On the other hand, the large nonlinear optimization problems are usually composed of a big linear part and much smaller nonlinear part. Other

attractive features of projection methods include a clear indication of how to accelerate them, easily seen from the convergence analysis: the method should make long steps. Several things seem discouraging, for example, the theoretical worst-case convergence for projection methods is not much competitive, but the author proposes some ways of taking advantage of information taken from quadratic models of the nonlinear optimization problem, which should accelerate the solving process.

In Section 2 a brief introduction to projection methods is given. The next two sections show a few-years work of the author on adapting projection methods for nonlinear optimization. Two approaches are presented. One of them is only briefly summarized in Section 3 (it was in more detail presented in [4] and [5]). This is a decomposition approach in which projections are realized via solving two different optimization subproblems with auxiliary solvers. One of the subproblems involves the linear constraints from the initial problem and the second one – the nonlinear ones. Due to the separation of constraints, specialized solvers (pure linear and nonlinear) may be used for the subproblems. The method is designed for problems in which the large size is generated mainly by the size of the linear part. The main effort in designing this method was done to generate special accelerating cuts that preserve the good features of the parts of the initial problem (sparse structure of the linear part and the low dimensionality of the nonlinear part). A very interesting feature of this method is that one of its best behaviors can be expected on the so-called nonlinear multicommodity flow problem, a classical item in telecommunication network design.

Section 4 presents an approach introducing projection methods in solving nonlinear optimization problems via the multiplier/penalty shifting method. The multiplier method produces unconstrained subproblems. Due to the spline character of these subproblems they might be sometimes very hard to solve, and the level of this hardness surprized the author who was trying to tackle them with a conjugate gradient method. The author proposes replacing the conjugate gradient method with a special kind of projection method, in which, however, a quadratic model of the minimized function and elements of the conjugate gradient method are still present and allow to obtain long steps in the projection method. This approach is formulated as a core algorithm and may have various realizations, each of them probably requiring a considerable amount of additional conceptual work.

Finally, in Section 5 some discussion and the author's observations regarding the methods as well as conclusions stemming from the author's experience are given.

2. The idea of projection methods

Projection methods serve to solving the following convex feasibility problem:

$$\text{Find } x \in S \stackrel{\text{def}}{=} \bigcap_{i=1, \dots, m} G_i, \quad (1)$$

where $G_i \subset \mathbb{R}^n$ are closed, convex sets. In practice, G_i are often defined as sets of points allowed by some constraints. Assume that S is nonempty. We start our description with the case of $m = 2$.

For $x \in \mathbb{R}^n$ and a closed convex nonempty $C \subset \mathbb{R}^n$ we shall denote by $P_C x$ the (unique) orthogonal projection of x onto C , $P_C x = \arg \min_{y \in C} \|x - y\|^2$. The projection vector for such a projection is $P_C x - x$.

The idea of searching for the solution consists in performing sequential alternate projections onto G_1 and G_2 ; i.e., given the starting point x^0 , we produce a sequence

$$x^1 = P_{G_1} x^0, x^2 = P_{G_2} x^1, x^3 = P_{G_1} x^2, \text{ etc.} \quad (2)$$

We assume such projections are easily realizable numerically.

In the convergence analysis of projection methods it is important that the projection operator possesses the Fejér contraction property.

Definition. A finite or infinite sequence (x^i) of points in a Hilbert space H has the Fejér contraction property with respect to $C \subset H$ if

$$\|x^i - c\|^2 \geq \|x^{i+1} - c\|^2 + \|x^{i+1} - x^i\|^2 \quad (3)$$

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

Fact. Projecting onto a nonempty closed convex set of points in \mathbb{R}^n has Fejér c. p. with respect to this set.

For a proof of the above fact see calculations on page 228 in [14] with $t_{\min} = t_{\max} = 1$.

After putting $C = S$ we see that with every projection performed in our algorithm (2) we decrease the squared norm from (any but fixed) point $c \in S$ by at least the square of the appropriate step (projection vector) length. It now suffices to assure certain lengths of steps to establish the convergence¹.

In other words, the Fejér contraction property of projections in our algorithm means that we approach each solution point with an acute angle.

¹Which is usually easy and is done with the notion of the problem geometrical property called *regularity* – see [2].

Zigzagging often slows down projection methods: we may approach the solution with an angle less than but close to $\pi/2$, making the distance from a solution decrease very slowly. This happens in an example in Fig. 1; there, moreover, consecutive projection vectors form angles close to π .

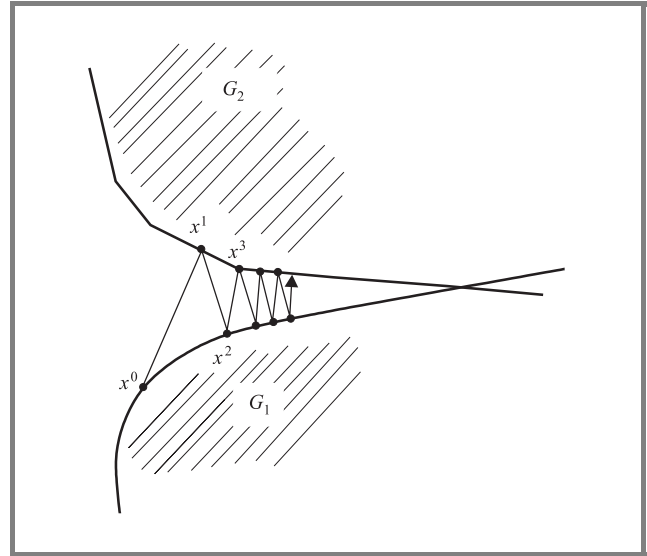


Fig. 1. Zigzagging.

Cuts serve as a standard remedy for zigzagging. Given point a and vector p in a Hilbert space, we define a cut as an inequality $\langle \cdot - a, b \rangle \geq \langle b, b \rangle \geq 0$ with fixed $a, b \in \mathbb{R}^n$; its hyperplane $H(a, b)$ is given as $\{x \in \mathbb{R}^n : \langle x - a, b \rangle = \langle b, b \rangle\}$, its halfspace – as $\{x \in \mathbb{R}^n : \langle x - a, b \rangle \geq \langle b, b \rangle\}$.

Using cuts means replacing (2) with

$$x^1 = P_{G'_1} x^0, x^2 = P_{G'_2} x^1, x^3 = P_{G'_3} x^2, \text{ etc.} \quad (4)$$

where sets $G'_1{}^k$ and $G'_2{}^k$ ($k = 1, 2, 3, \dots$) are G_1 and G_2 narrowed by some cuts, i.e. they were obtained from G_1 and G_2 by intersecting G_1 and G_2 with halfspaces of some cuts.

A geometric cut based on (constructed after) the projection of $x \notin G$ onto close convex G , $G \supset S$ is defined as

$$\langle \cdot - x, P_G x - x \rangle \geq \langle P_G x - x, P_G x - x \rangle.$$

In Fig. 2, unlike in Fig. 1, point x^3 was obtained by projecting x^2 not onto G_2 but onto G_2 narrowed by the geometric cut constructed after projection of x^1 onto G_1 . H is a hyperplane of this cut. We see that the step made is longer and we approach the solution with a smaller angle.

A cut is called *valid* or *proper* if it is satisfied for each point in the solution set S . Validity is necessary to assure that narrowed sets (i.e., $G'_1{}^k$ or $G'_2{}^k$) contain S and thus projection onto them still possesses the Fejér contraction property with respect to S ; moreover, we do not want our method to degenerate by producing empty $G'_1{}^k$ or $G'_2{}^k$. Geometric cuts constructed after a projection of an $x \notin G$ onto nonempty, closed, convex $G \supset S$ can be easily shown to be proper.

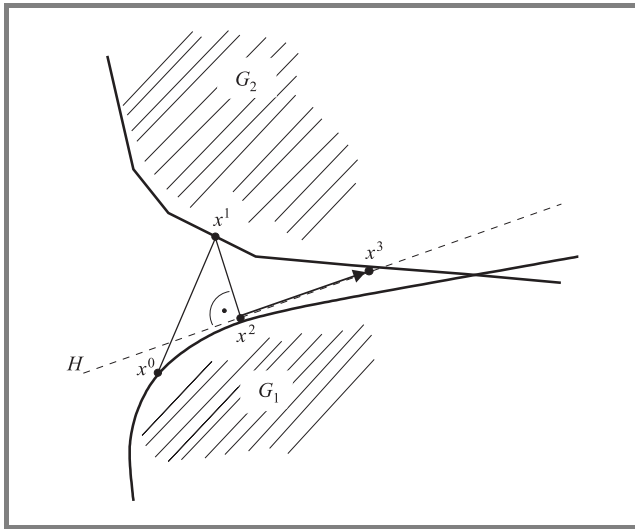


Fig. 2. A geometric cut reduces zigzagging.

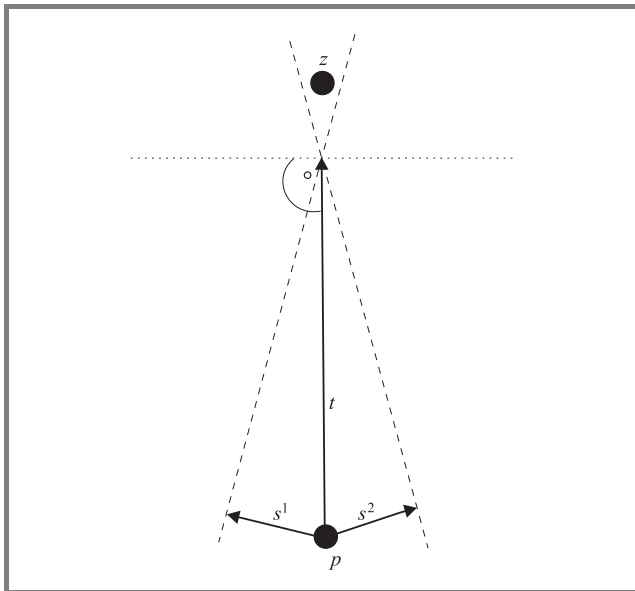


Fig. 3. Obtaining a surrogate cut (dotted line) from two cuts (dashed lines) whose normal vectors form an obtuse cone; $q = 2$.

We may narrow set G_1 or G_2 with only one cut but it may bring a profit in efficiency to narrow them with several cuts simultaneously (i.e., to intersect G_1 or G_2 with the intersection of the halfspaces of several cuts). Various techniques for cuts cumulation are given in [6, 7, 12, 13, 16]. We can:

1. Project on the real intersection of halfspaces of cumulated cuts by solving an optimization problem that reflects the definition of orthogonal projection (it has a quadratic goal function).
2. Construct a valid “surrogate” cut on basis of the cumulated cuts and project on its halfspace, hopefully obtaining a long projection vector (step).

A way of obtaining a surrogate cut from q geometric cuts, say $\langle \cdot - p, t_i \rangle \geq \langle t_i, t_i \rangle$, $i = 1, \dots, q$ was given by Cegielski [6, 7] (and a similar approach – by Kiwiel [12]). Here p is the current iterate point.

Lemma (adopted Remark 7 on Theorem 3 in [6]).
Let $p, z \in \mathbb{R}^n$, $p \neq z$. If

- (a) $\mathcal{S} = \{s^i : i = 1, \dots, q\}$ is a system of linearly independent vectors,
- (b) $\forall_{i \in \{1, \dots, q\}} \langle z - (p + s^i), s^i \rangle \geq 0$,
- (c) coneconv \mathcal{S} obtuse²,
- (d) t is the solution of the following equation system

$$\forall_{i \in \{1, \dots, q\}} \langle (s^i), t - s^i \rangle = 0 \quad (5)$$

then $\langle z - (p + t), t \rangle \geq 0$.

This lemma says that the cut $\langle \cdot - (p + t), t \rangle \geq 0$ is valid on condition that the cumulated cuts are valid³ – see Fig. 3; this is the surrogate cut in Cegielski’s method. The next iterate is obtained from p by adding t , a long vector, to it.

However, the normals s^i of cuts must form an obtuse cone. Cegielski assures it in several ways, the easiest one is to assure $\langle s^i, s^j \rangle \geq 0$ for $i \neq j$ (so s^i form a so-called regular obtuse cone), by a simple rejection of some cuts to be cumulated.

Any convex optimization problem of the form $\min_{x \in \bigcap_{j=1}^m G^j \subset \mathbb{R}^n} f(x)$ may be solved by means of projection methods for feasibility problems after reducing it to a feasibility problem of finding a common point of G^j , $j = 1, \dots, m$ and $\{x \in \mathbb{R}^n : f(x) < Q\}$, parametrized with number Q , which must be experimentally tuned to the optimal value of the initial optimization problem within some scheme, e.g. bisection or level control [11]. Thus we actually need to solve a sequence of feasibility problems with various Q s; usually a detection of infeasibility of a feasibility problem must happen from time to time.

3. Decomposition of large problems into linear and nonlinear parts

The algorithm described in this section is multi-layered, similarly as the group of approaches described in the next section. They both combine some elements typical for smooth optimization with elements of projection methods. Here the projection layer is higher than the layer of standard smooth techniques.

²An acute cone is a cone C such that for $a \in C$, $b \in C$ $\langle a, b \rangle \leq 0$, an obtuse cone is a cone conjugate to some acute cone.

³i.e., they do not cut off any solution point represented here by z .

The solved problem after reducing to a feasibility problem with parameter Q has the following form: find $(x^\top, y^\top)^\top$ that satisfies:

$$\begin{aligned} g(x) &\leq Q \\ A(x^\top, y^\top)^\top &\leq b \\ B(x^\top, y^\top)^\top &= d \\ x^{lo} \leq x \leq x^{up}, y^{lo} &\leq y \leq y^{up}, \end{aligned} \quad (6)$$

where function $g : \mathbb{R}^{m_N} \rightarrow \mathbb{R}^{m_N}$ depends in one coordinate on Q , since this coordinate was made from the goal function of the optimization problem (the coordinate says how much the goal function value exceeds Q). A and B are matrices of appropriate sizes. Functions g_i are continuous quasiconvex, $x^{lo}, x^{up}, y^{lo}, y^{up}$ are constant vector bounds.

The feasibility problem has n_N nonlinear variables⁴, n_L linear variables, m_N nonlinear inequality constraints, m_{LI} linear inequality constraints, m_{LE} linear equality constraints. Let $m = m_N + m_{LI} + m_{LE}$, $n = n_L + n_N$. The better $m_N \ll m$ and $n_N \ll n$, are fulfilled, the more efficient the algorithm will be.

In order to solve the problem (6) we need to see it in the form of (1).

The following sets N and L will play the role of G_1 and G_2 in problem (1):

$$\begin{aligned} N &= \{x \in \mathbb{R}^{n_N} : g(x) \leq 0 \wedge x^{lo} \leq x \leq x^{up}\} \\ L &= \left\{ x \in \mathbb{R}^{n_N} : x^{lo} \leq x \leq x^{up} \wedge \right. \\ &\quad \left. \wedge \exists_{y \in \mathbb{R}^{n_L}} \left(y^{lo} \leq y \leq y^{up} \wedge \right. \right. \\ &\quad \left. \left. \wedge A(x^\top, y^\top)^\top \leq b \wedge B(x^\top, y^\top)^\top = d \right) \right\}. \end{aligned}$$

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 projection algorithm operates in this low-dimensional subspace.

Projections on these sets are realized by solving optimization subproblems with quadratic goal functions, moreover:

1. A projection on N yields a small ($n_N \times m_N$) subproblem with nonlinear constraints.
2. A projection on L yields a large ($n \times m_L$) subproblem but with linear constraints.

The optimization subproblems are solved efficiently with specialized solvers, in the author's experiments nonlinear IAC-DIDASN++ [15] and quadratic HOPDM [1].

A special care is connected with using geometric cuts, but this will be only outlined here (see [4] and [5] for details). An introduction of geometric cuts in general means extending the above subproblems by adding suitable linear inequality constraints to them. Whenever we use such a cut,

⁴A *nonlinear variable* is a problem variable involved in at least one nonlinear function in the model formulation; the remaining variables will be called *linear*.

the inequality must be present in the subproblem realizing the projection.

We can freely construct a geometric cut based on a projection onto N : such a cut introduces at most n_L nonzeros into constraint matrices of the quadratic subproblem, which is not much by comparison with the nonzeros number in constraints from the quadratic optimization subproblem.

Using a cut based on a projection onto L should be avoided: if we wanted to use such a cut in some next projection onto N , we would introduce a bigger relative complication into the nonlinear subproblem. Thus we treat this cut only as "virtual", e.g., we state that such a cut might be constructed and would be proper, but we do not really add it to any collection of cumulated cuts that augments N . Then we cumulate such a "virtual" cut with the latest cut based on the projection onto N , obtaining a surrogate cut according to Lemma in point 2 in Section 2. The surrogate cut, called anti-zigzagging cut (or Z-cut) is used later instead of the "virtual" cut to augment subproblems; however, it augments only the quadratic subproblems so the subproblem complication is not excessive.

The successive Z-cuts can be then cumulated directly, by cumulation of constraints augmenting a subproblem (i.e., in the way described in point 1 in Section 2). Since the cumulation is full, i.e., each successive Z-cut is cumulated, the zigzagging in the method is claimed to decrease in the cited works.

A nice feature of the method is discussed in [5]. The required proportions of sizes are particularly good when we apply the method to the nonlinear multicommodity flow problem [18]. Moreover, the situation becomes better as the number of commodities distinguished in the problem grows.

4. Accelerating the multiplier method with projection methods elements

In this section we shall want to solve the following problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \varphi(x) & \quad (7) \\ \text{s. t.} & \\ g(x) \leq 0 & \quad (8) \end{aligned}$$

with $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, g^i and φ continuous quasiconvex.

The idea of algorithm presented in this section bases on an observed poor behavior of a penalty shifting/multiplier method in the version for inequality constraints ([3], see also [21]) and with the Fletcher-Reeves conjugate gradient method applied to the resulting unconstrained subproblems. The author once solved a problem of the form (7)–(8) with several tens (!) of variables, a quadratic goal function and several tens or about a hundred of quadratic inequality constraints. Even on such a small example, the

solution times sometimes reached a rank of hours. The number of iterations of the multiplier method was small, but the resulting subproblems were solved extremely slow. Probably setting a too high penalty coefficient was not the reason for this behavior, since the coefficient did not exceed the value of 1 even by several ranks of value; neither did the coefficients in the initial problem definition differ from 1 by many ranks. The probable reason for such a behavior of the method was formulated as a nonsmooth (actually spline) character of the unconstrained subproblems, caused by a similar character of the augmented Lagrangian function.

We can find several signs in the literature that seem to support the anxiety about the augmented Lagrangian for inequality constraints and the application of it being spoken. One of them is the existing collection of trials of modifying the augmented Lagrangian in order to eliminate its quadratic-spline character. An example is a construction of a cubic Lagrangian. A second one might be the way of treating nonlinear inequality constraints in the broadly recognized LANCELOT solver [8]. Instead of using the version of the multiplier method for inequality constraints directly, the nonlinear inequality constraints are first transformed to nonlinear equality constraints by an addition of bounded slack variables, similarly to the way it is done in the simplex method. Then the equality constraints are treated with the variant of the multiplier method for equality constraints, in which the Lagrangian is smooth. The constructors of the solver even agree with a possibility of obtaining nonconvex subproblems, with additional slacks and bounds (the bounds on slacks are transferred to the unconstrained subproblems of the multiplier method).

Having in mind the hardness of the subproblems with nonsmooth Lagrangian, the author of this paper decided to solve the subproblems with a variant of projection method.

If we wanted to truly treat the Lagrangian (the goal function of the unconstrained subproblem) as a nondifferentiable function, we would perhaps want to use some variant of the Polyak subgradient method [19] for minimizing a convex goal function ϕ . In an iteration of this method, one makes a projection of the current iterate point on the Q -level set of the linear underestimation of ϕ constructed on the basis of the value and the subgradient of ϕ at the current iterate; Q denotes, as previously, the current estimation of the minimal value of ϕ . But it seems better for the convergence speed if we take an advantage of a quadratic model of the minimized Lagrangian (let us denote it as $f: \mathbb{R}^n \mapsto \mathbb{R}$) which may be, at least locally, good if the initial optimization problem was smooth.

In a Polyak method iterate we actually use the information at one point and we obtain a projection vector of a certain length. In the author's proposition we first make $k \leq n$ steps (an epoche of steps) of the Fletcher-Reeves method, say, while the quadratic approximation of f seems good enough. From there we have an approximate model of the function in a whole subspace Δ of the dimensionality of k . This information allows us to find a point $\tilde{y} \in \Delta$ at which

we can construct a valid cut: $\langle -\tilde{y}, \nabla f(\tilde{y}) \rangle \leq 0$. Then we project the current iterate p onto the halfspace of this cut; since the cut is valid, the projection possesses the Fejér contraction property w.r.t. any solution point. The vector of projection of current iterate p onto the halfspace of this cut is hoped to be much longer than that in the Polyak method, since we can choose \tilde{y} from the whole subspace Δ . As we remember, a big step length usually implies a quicker convergence in projection methods.

We shall manage only to outline the proposition, since it is quite sophisticated and may have many variants. We start with the heart of the proposition, which is calculating \tilde{y} .

Assume the epoche of conjugate gradient method generated points $x^0 \equiv p, x^1, \dots, x^k \in \mathbb{R}^n$, conjugate directions: $d^0, d^1, \dots, d^{k-1} \in \mathbb{R}^n$, gradients of f : $g^0 = \nabla f(x^0), g^1 = \nabla f(x^1), \dots, g^k = \nabla f(x^k)$, real coefficients $\beta^1, \beta^2, \dots, \beta^k$ and the step lengths $\alpha^0, \alpha^1, \dots, \alpha^{k-1}$.

These objects are interrelated with the following dependencies:

$$d^0 = -g^0, \tag{9}$$

$$d^i = -g^i + \beta^i d^{i-1} \quad (i = 1, \dots, k), \tag{10}$$

$$x^{i+1} = x^i + \alpha^i d^i \quad (i = 0, \dots, k-1). \tag{11}$$

For simplicity we assume $x^k = 0$.

We shall now treat f as a quadratic function defined with a symmetric, positive definite matrix.

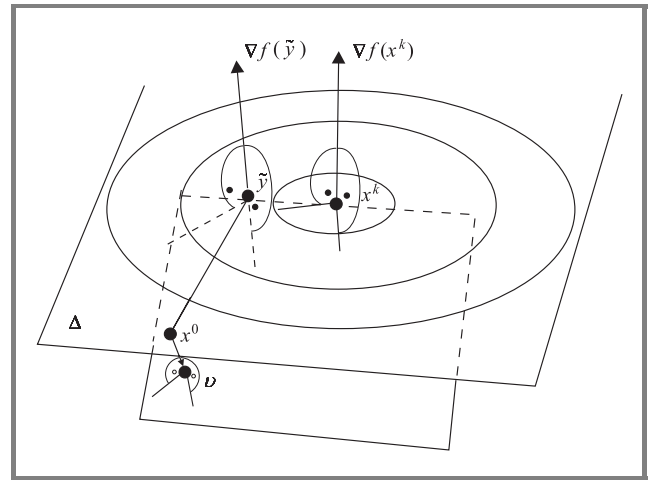


Fig. 4. Choosing point \tilde{y} . The ellipses denote the level sets of f , the smaller plane is the plane of the cut and v is the projection vector that should be as long as possible.

The problem of choosing \tilde{y} is shown in Fig. 4. Vector v is the projection vector and we want to make it as long as possible. In order to make a small exercise try to imagine how this figure change for two choices: $\tilde{y} = x^0$ and $\tilde{y} = x^k$. Observe that x^0 is a poor candidate for \tilde{y} , since for such a choice the final steplen $|v|$ would be equal to 0. Neither $\tilde{y} = x^k$ is a good choice, since from theory of conjugate gradient methods we know that $g^k \perp \Delta$, which against yields $\|v\| = 0$.

The heuristic taken by the author is to search for \tilde{y} among points x satisfying⁵

$$\nabla_{\parallel} f(x) = -\tau(x - x^0), \quad (12)$$

$$x \in \Delta \quad (13)$$

for some positive parameter τ , where $\nabla_{\parallel} f$ denotes the part of ∇f parallel to Δ .

The solution of system (12)–(13) becomes easier when we represent x in the basis of conjugate directions: $x = Ds$, $D \equiv [d^0 d^1 \dots, d^{k-1}]$ and thus reduce them to a search for the best s .

Under such a representation, system (12)–(13) transforms, with some elaborated calculations, to

$$(I + \tau M)s = q \quad (14)$$

with

$$M = \begin{bmatrix} \frac{1}{\alpha^0} & -\frac{\beta^1}{\alpha^1} & & & & \\ \frac{\beta^1 - 1}{\alpha^0} & \frac{\beta^2 + 1}{\alpha^1} & & & & \\ & -\frac{1}{\alpha^1} & \ddots & & & \\ & & & -\frac{\beta^{k-2}}{\alpha^{k-2}} & & \\ & & & \frac{\beta^{k-1} + 1}{\alpha^{k-2}} & -\frac{\beta^{k-1}}{\alpha^{k-1}} & \\ & & & -\frac{1}{\alpha^{k-2}} & \frac{1}{\alpha^{k-1}} & \end{bmatrix} \quad (15)$$

and q being the representation of x^0 in the basis of conjugate directions: $x^0 = Dq$.

Now the search for the best \tilde{y} is reduced to a search for optimal $\tau > 0$. For a candidate τ we find s by solving a quite easy system (14), with a tridiagonal matrix, compute $\tilde{y} = Ds$ and for it (based on real gradient of the minimized function f , or on its quadratic model being considered now), the projection vector. The length of the projection vector seems not to be in general a concave function of τ , but practical experiments showed that one-dimensional optimization in τ may be replaced just with examining several values of τ .

The cuts generated by the algorithm are then cumulated with the Cegielski's method of regular obtuse cone presented in Section 2, but only a limited number of them takes part in the cumulation process in order to keep the linear systems present in the method (one in the cuts cumulation process, second in searching for \tilde{y}) small.

5. Discussion

The method of decomposition from Section 3 has been thoroughly discussed in [4] and [5]. Several modifications of

⁵The demand of exact maximization of $\|v\|$ yields a multidimensional nonconvex optimization problem. Instead, in the heuristics we want to be far from the situation where angle between $(\tilde{y} - x^0)$ and $\nabla_{\parallel} f(\tilde{y})$ equals to $\pi/2$, as it happened with the choice of $\tilde{y} = x^k$.

this method are possible, e.g., a possibility of augmenting set N with geometric cuts instead of set L if we decide so by more precise analysis of particular problem sizes. Other options include subtle changes in the order of cuts cumulation, which may affect the speed of convergence. The method performed quite good on an artificial multicommodity flow problem in the sense of number of iterations in the projection method layer. Thus, the decomposition of the problem into linear and nonlinear parts seems to be done well, but the overall effectiveness of the method depends on the speed of the solvers solving pure (quadratic or nonlinear) subproblems. Applying warm restarts during many runs of the quadratic solver seems to be necessary in order to make this method competitive with commercial solvers on this problem.

Regarding the method of combining projection and the conjugate gradient method from Section 4, one must be aware of a great number of technical details and further decisions that we face when trying to implement it, in particular:

1. A separate treatment of equality constraints in large problems. We constructed our method for purely nonlinear problems. We can formally represent linear constraints as nonlinear, but for large problem it usually becomes essential to treat them separately⁶. Linear constraints can be introduced directly in the form of additional cuts to be cumulated in the conical method.
2. Introducing bounds on variables would somewhat complicate the algorithm; perhaps some elements of projected gradient method would have to be used, may be the bounds would have to be added to the collection of cuts being cumulated.

Speaking about large problems, let us make an important note. Due to inserting linear constraints directly in the projection method steps and due using some conjugate gradient techniques (not variable metric) there is a chance to design the whole algorithm so that any complicated linear algebra, like an implicit inverting sparse matrices, is avoided. Making the cuts based on points \tilde{y} even deeper than in the above descriptions seems to be another important issue. In some circumstances such operations might be essential for an efficient work of the method.

1. Having an approximation Q for the optimal value of the unconstrained subproblem, one can shift any constructed cut (make it deeper) by using techniques known from the Polyak method (apply a linear model of f constructed at point \tilde{y}). However, f must be convex, not only quasiconvex, to make this approach proper.

⁶Remember also that the method does not support nonlinear equality constraints, so linear equality constraints cannot be represented as nonlinear. A propos, the inability of treating nonlinear equality constraints is common to both the propositions in this paper. It seems essential since it stems from the fact that projection methods work only for convex problems. Possibly, one might try some trust region approach to incorporate nonlinear equality constraints.

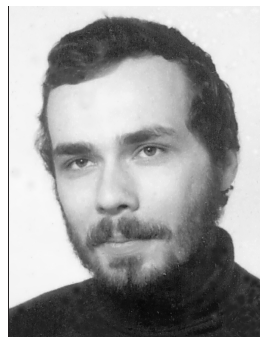
2. Do not use the Lagrangian in all runs of the subproblem solving, but instead its different valid underestimations, which are easy enough to obtain due to the spline character of the Lagrangian. This may introduce some perturbations in the cuts positions (usually deepening) resulting in long steps after the cuts cumulation. A difficulty occurs in such a case: one must stop conjugate gradients run when we go below the optimal value of the subproblem, since otherwise obtained cuts would be invalid.

Many technical decisions, some of them mentioned earlier, certainly must be made in order to make the process of finding τ work properly.

Another concern must be connected with the work of the method of cuts cumulation itself. It might be augmented or tuned for some interesting patterns configuration of cuts, frequently observed during experiments (closeness of the angles between the majority of cut normal pairs to $\pi/2$; very obtuse cones observed for some problems).

References

- [1] A. Altman and J. Gondzio, "Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization", *Optim. Meth. Softw.*, no. 11–12, pp. 275–302, 2000.
- [2] H. Bauschke and J. Borwein, "On projection algorithms for solving convex feasibility problems", *SIAM Rev.*, vol. 38, no. 3, pp. 367–426, 1996.
- [3] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. A Series of Monographs and Textbooks: *Computer Science and Applied Mathematics*. New York, London: Academic Press, 1982.
- [4] P. Białoń, "Large-scale nonlinear projection algorithm using projection methods", *Discuss. Math. Differ. Inclusion., Control Optim.*, vol. 20, no. 2, pp. 171–194, 1999.
- [5] P. Białoń, "A proposition to exploit the partially linear structure of the nonlinear multicommodity flow optimization problem", *J. Telecommun. Inform. Technol.*, no. 3, pp. 49–56, 2002.
- [6] A. Cegielski, "A method of projection onto an acute cone with level control in convex minimization", *Math. Progr.*, vol. 85, pp. 469–490, 1999.
- [7] A. Cegielski, *Metody relaksacyjne w problemach optymalizacji wypukłej*. Monografie, nr 67. Zielona Góra: Wyższa Szkoła Inżynierska, 1993 (in Polish).
- [8] A. Conn, N. Gould, and P. Toint, "Numerical experience with LANCELOT package (Release A) for large-scale nonlinear optimization", *Math. Progr.*, vol. 73, pp. 73–110, 1996.
- [9] R. Dylewski, "Numerical behavior of the method of projection onto an acute cone with level control in convex optimization", *Discuss. Math. Differ. Inclusion., Control Optim.*, vol. 20, pp. 147–158, 1999.
- [10] S. Flåm and J. Zowe, "Relaxed outer projections, weighted averages and convex feasibility", *BIT*, vol. 30, pp. 289–300, 1990.
- [11] S. Kim, A. Hyunsil, and C. Seong-Cheol, "Variable target value subgradient method", *Math. Progr.*, vol. 49, pp. 356–369, 1991.
- [12] K. Kiwiel, "Monotone Gram matrices and deepest surrogate inequalities in accelerated relaxation methods for convex feasibility problems", *Lin. Algebra Appl.*, vol. 215, pp. 27–33, 1997.
- [13] K. Kiwiel, "The efficiency of subgradient projection methods for convex optimization", Part I: "General level methods", *SIAM Control Optim.*, vol. 34, no. 2, pp. 660–676, 1996.
- [14] K. Kiwiel, "Block-iterative surrogate projection methods for convex feasibility problems", *Lin. Algebra Appl.*, vol. 15, pp. 225–259, 1995.
- [15] T. Kręglewski, J. Granat, and A. P. 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. Laxenburg (Austria): International Institute for Applied Systems Analysis, June 1991.
- [16] C. Lemaréchal, A. S. Nemirovskii, and Yu. Nesterov, "New variants of bundle methods", *Math. Progr.*, vol. 69, pp. 111–147, 1995.
- [17] M. Makowski, *LP-DIT data interchange tool for linear programming problems (version 1.20)*. Working Paper, WP-94-36. Laxenburg (Austria): International Institute for Applied Systems Analysis, 1994.
- [18] M. Minoux, "Network synthesis and optimum network design problems: models, solution methods and applications", *Network*, vol. 19, pp. 313–360, 1989.
- [19] T. Polyak, "Minimization of unsmooth functionals", *Zh. Vychisl. Mat. Fiz.*, vol. 9, *Mat. Fiz.*, vol. 9, pp. 14–29, 1969.
- [20] M. Shchepakin, "On a modification of a class of algorithms for mathematical programming", *Zh. Vychisl. Mat., Mat. Fiz.*, vol. 19, pp. 1387–1395, 1979 (in Russian).
- [21] A. Wierzbicki, "A penalty function shifting method in constrained static optimization and its convergence properties", *Archiw. Autom. Telemekh.*, vol. XVI, no. 4, pp. 396–416, 1971.



Paweł M. Białoń was born in Warsaw in 1971. He received his M.Sc. in computer science from Warsaw University of Technology in 1995. Currently he is employed by National Institute of Telecommunications in Warsaw. His research focuses on nonlinear optimization methods and on decision support and is directed towards

the Ph.D. degree. He has participated in several projects applying decision support methods in the telecommunications, agricultural and environmental areas.

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

National Institute of Telecommunications

Szachowa st 1

04-894 Warsaw, Poland