

PROXIMAL BUNDLE METHOD FOR CONTACT SHAPE OPTIMIZATION PROBLEM

Nikola PLIVOVA, Petr BEREMLIJSKI

Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science,
VSB–Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic

nikola.plivova@vsb.cz, petr.beremlijski@vsb.cz

DOI: 10.15598/aeee.v15i2.2320

Abstract. *From the mathematical point of view, the contact shape optimization is a problem of nonlinear optimization with a specific structure, which can be exploited in its solution. In this paper, we show how to overcome the difficulties related to the nonsmooth cost function by using the proximal bundle methods. We describe all steps of the solution, including linearization, construction of a descent direction, line search, stopping criterion, etc. To illustrate the performance of the presented algorithm, we solve a shape optimization problem associated with the discretized two-dimensional contact problem with Coulomb’s friction.*

Keywords

Clarke calculus, nonsmooth optimization, proximal bundle method, shape optimization.

1. Introduction

Shape optimization problems arise naturally in mechanical engineering whenever the design requirements include an optimal performance of a machine or a gadget comprising several bodies in mutual contact. From the mathematical point of view, these problems can be characterized by a locally Lipschitz continuous cost function, which is differentiable in most but not all points. Moreover, its gradient is not defined explicitly even in the point, at which it exists. Shape optimization problems have the following form:

$$\begin{cases} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \subset \mathbb{R}^n. \end{cases} \quad (1)$$

The solution of such problems can be obtained by a suitable iterative algorithm – its typical structure reads as follows.

Algorithm 1 Basic iterative algorithm.

Step 0: (Initialization)

Find a feasible starting point $x_1 \in \Omega$ and set $k = 1$.

Step 1: (Stopping criterion)

If x_k is "close enough" to the required solution then STOP.

Step 2: (Direction finding)

Find a feasible descent direction $d_k \in \mathbb{R}^n$:

$f(x_k + td_k) < f(x_k)$ and $x_k + td_k \in \Omega$ for some $t > 0$.

Step 3: (Line search)

Find a step size $t_k > 0$ such that:

$t_k \approx \arg \min_{t>0} \{f(x_k + td_k)\}$ and $x_k + td_k \in \Omega$.

Step 4: (Updating)

Set $x_{k+1} = x_k + t_k d_k, k = k + 1$ and go on to Step 1.

The hardest difficulty is the direction searching in Step 2 since the cost function f mentioned in the above section is not differentiable but only locally Lipschitz continuous. This implies that to minimize the function f , we can choose an algorithm from the following two classes: derivative-free methods (like genetic algorithms) and methods that use the subgradient information (like subgradient or bundle methods). Since the subgradient information is available for our problem, we have chosen the latter class of algorithms. In this paper, the proximal bundle method (see [4]) is presented. This method needs the function value $f(x)$ and one (arbitrary) Clarke subgradient of f at x in every step of the iteration process.

Key terms and theorems are introduced in the first part of this paper and then some important sections present direction finding, line search or subgradient aggregation in detail. To show the functionality of the presented algorithm, we solve a shape optimization problem with the discretized two-dimensional contact problem with Coulomb’s friction in the last part.

2. Proximal Bundle Method

2.1. Nonsmooth Analysis - Clarke Calculus

Before we come to the derivation of the aforementioned method, it is necessary to be familiar with the necessary theory. Since we often use the term generalized gradient and subgradients, it is suitable to define them.

Definition 1. A function $f: \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be Lipschitz continuous on Ω if there exists some constant $L = L(\Omega) > 0$ such that

$$|f(x) - f(y)| \leq L \|x - y\|, \quad \forall x, y \in \Omega. \quad (2)$$

Definition 2. A function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be Lipschitz continuous at $x \in \mathbb{R}^n$ if there exists a neighbourhood U of x and a constant $L = L(U) > 0$ such that

$$|f(x) - f(y)| \leq L \|x - y\|, \quad \forall y \in U. \quad (3)$$

Definition 3. A function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be locally Lipschitz continuous in \mathbb{R}^n if this function f is Lipschitz continuous at $x \in \mathbb{R}^n$ for every $x \in \mathbb{R}^n$.

Definition 4. Let $\Omega \subset \mathbb{R}^n$. Then $\text{conv}(\Omega)$ denotes the convex hull of the set Ω , which is defined by:

$$\text{conv}(\Omega) = \left\{ \sum_{i=1}^n \lambda_i x_i \mid n \in \mathbb{N}, \lambda \in \mathbb{R}^n, x_1, \dots, x_n \in \Omega, \lambda_i \geq 0, \forall i, \sum_{i=1}^n \lambda_i = 1 \right\}. \quad (4)$$

Remark 1. The set $\text{conv}(\Omega)$ is the set of all convex combinations of the points in Ω .

Definition 5. Let the objective function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be locally Lipschitz continuous (in \mathbb{R}^n). The generalized gradient of the objective function f at $x \in \mathbb{R}^n$ is the set

$$\partial f(x) = \text{conv} \left\{ g \in \mathbb{R}^n \mid g = \lim_{i \rightarrow \infty} \nabla f(x_i), \right. \\ \left. x_i \rightarrow x, x_i \notin \Omega_f \right\}, \quad (5)$$

where $\Omega_f = \{x \in \mathbb{R}^n, f \text{ is not differentiable in } x\}$.

Each element $g \in \partial f(x)$ is called a subgradient of the objective function f at x .

Definition 6. Let f be locally Lipschitz continuous at $x \in \mathbb{R}^n$. Then the generalized (Clarke) directional derivative of f at the point x in the direction v is defined by:

$$f^\circ(x; v) = \max \{ g^T v \mid g \in \partial f(x) \}, \quad (6)$$

for all $v \in \mathbb{R}^n$.

We illustrate the previous definition Def. 5 in the following example.

Example 1 (Generalized gradient). Find the generalized gradient of a given function f at $x = 0$, where

$$f(x) = |x - 1| + |x| + |x + 1|. \quad (7)$$

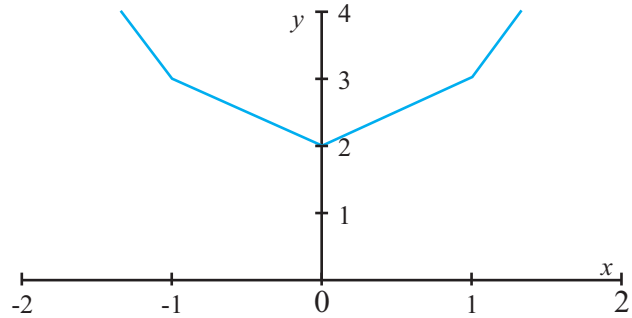


Fig. 1: Graph of function f .

This function (see its graph in Fig. 1) can be rewritten in the form

$$f(x) = \begin{cases} -3x & \text{if } x \in (-\infty, -1), \\ -x + 2 & \text{if } x \in (-1, 0), \\ x + 2 & \text{if } x \in (0, 1), \\ 3x & \text{if } x \in (1, +\infty). \end{cases} \quad (8)$$

As the function f is not smooth at $x = 0$, we have to use one-sided limits of the derivatives of the function f to compute the generalized gradient (see Def. 5). Let us start with the interval $(0, 1)$. For each $x \in (0, 1)$ the derivative is $\nabla f(x) = 1$ and therefore $g = \lim_{i \rightarrow \infty} \nabla f(x_i) = 1$ applies for $x_i \rightarrow 0^+$. We analogously have $g = \lim_{i \rightarrow \infty} \nabla f(x_i)$ for $x_i \rightarrow 0^-$. The generalized gradient of f at x is then:

$$\left. \begin{array}{l} x_i \rightarrow 0^+ : g = 1 \\ x_i \rightarrow 0^- : g = -1 \end{array} \right\} \partial f(0) = \text{conv} \{-1, 1\}, \quad (9)$$

and its graph can be seen in Fig. 2.

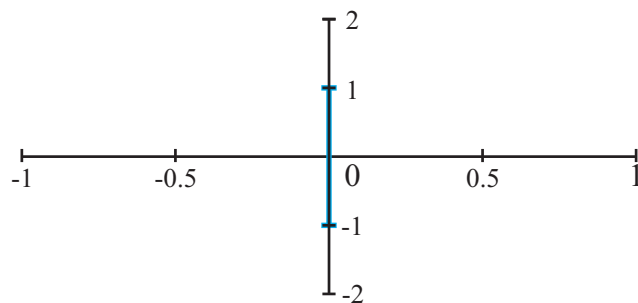


Fig. 2: Graph of the generalized gradient of f at $x = 0$.

In a similar way, we can compute the generalized gradient of the function f for all $x \in \mathbb{R}$ (see its graph in Fig. 3).

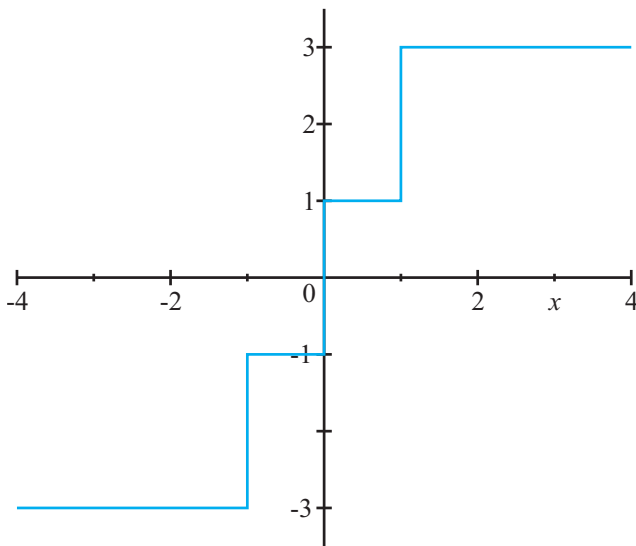


Fig. 3: Graph of the general gradient of f .

2.2. Optimality Conditions

Here are some necessary conditions for a constrained and for an unconstrained optimization problem. We should note that these conditions are only basic ones.

1) Optimality Conditions for Unconstrained Problems

Theorem 1. Let f be Lipschitz continuous at x and let f attain its local minimum at x , then

- $0 \in \partial f(x)$,
- $f^\circ(x; v) \geq 0$ for all $v \in \mathbb{R}^n$.

Consider the first part $0 \in \partial f(x)$ and note that in the Exm. 1 above, the function f attains its local minimum at $x = 0$. The generalized gradient of f at $x = 0$ is the convex hull of $\{-1, 1\}$, which is the interval $\langle -1, 1 \rangle$, and it is no surprise that $0 \in \langle -1, 1 \rangle$.

Theorem 2. Let f be a convex function, then the following conditions are equivalent:

- f attains its global minimum at x ,
- $0 \in \partial f(x)$,
- $f^\circ(x; v) \geq 0$ for all $v \in \mathbb{R}^n$.

2) Optimality Conditions for Constrained Problems

Tangent and normal cones are closely related to the optimality conditions for constrained problems, thus it is important to define them.

Definition 7. Let Ω be a nonempty subset of \mathbb{R}^n . The set Ω is called convex if

$$(1 - \lambda)x + \lambda y \in \Omega, \tag{10}$$

whenever $x \in \Omega, y \in \Omega$ and $\lambda \in \langle 0, 1 \rangle$.

Definition 8. The tangent cone of a convex set Ω at $x \in \Omega$ is defined by:

$$T_\Omega(x) = \{ y \in \mathbb{R}^n \mid \text{there exist } t_i \rightarrow 0^+ \text{ and } y_i \rightarrow y \text{ so that } x + t_i y_i \in \Omega \}. \tag{11}$$

Definition 9. The normal cone of the convex set Ω at the point $x \in \Omega$ is given by the formula:

$$N_\Omega(x) = \{ z \in \mathbb{R}^n \mid y^T z \leq 0 \text{ for all } y \in T_\Omega(x) \}. \tag{12}$$

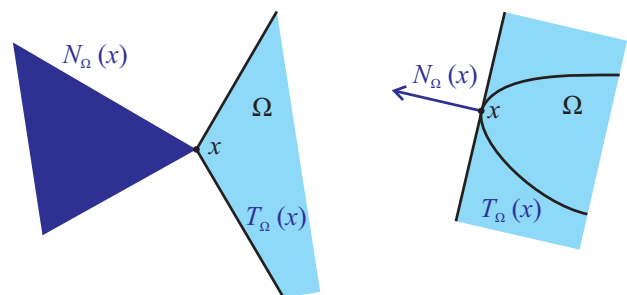


Fig. 4: Examples of the tangent and normal cones of given convex set Ω .

We can see some examples of the tangent and normal cones in Fig. 4. Now, we can write the corresponding conditions.

Theorem 3. Let f be Lipschitz continuous at $x \in \Omega$ and let x be a minimizer of the function f over the set Ω , then

$$0 \in \partial f(x) + N_\Omega(x). \tag{13}$$

Theorem 4. If the function f is convex and if the set Ω is also convex, then the following conditions are equivalent:

- $0 \in \partial f(x) + N_\Omega(x)$,
- f attains its global minimum at x over the set Ω .

For proofs of Thm. 1, Thm. 2, Thm. 3, Thm. 4 and further details on the theory of nonsmooth analysis the interested reader is referred to [4], [6] or [5]. Further definitions, pictures and examples can be found there.

2.3. Derivation of the Method

Consider the following nonlinear constrained optimization problem:

$$\left. \begin{array}{l} \text{minimize} \quad f(x) \\ \text{subject to} \quad Cx \leq b, \\ \quad \quad \quad x_{\min} \leq x \leq x_{\max}, \end{array} \right\} \quad (14)$$

where the objective function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is locally Lipschitz continuous in \mathbb{R}^n , $C \in \mathbb{R}^{m \times n}$ is a constraint matrix, $b \in \mathbb{R}^m$ is a right-hand side vector and $x_{\max} \in \mathbb{R}^n$, $x_{\min} \in \mathbb{R}^n$ are bound vectors. To make these notations simple we suppose that the simple bounds x_{\min}, x_{\max} are included in the linear system $Cx \leq b$. For further details on the proximal bundle method the interested reader is referred to [4].

1) Direction Finding

Our aim is to solve the problem with respect to $d \in \mathbb{R}^n$

$$\left. \begin{array}{l} \text{minimize} \quad f(x_k + d) - f(x_k), \\ \text{subject to} \quad x_k + d \in \Omega, \end{array} \right\} \quad (15)$$

where $\Omega = \{x \in \mathbb{R}^n \mid Cx \leq b\}$ and d is the descent direction.

Suppose that we have some starting point $x_1 \in \Omega$, the current iteration point $x_k \in \Omega$ and that we have subgradients $g_j^f \in \partial f(y_j)$ for all $j \in J_f^k$, where $J_f^k \subset \{1, \dots, k\}$ is a nonempty index set and where $y_j \in \Omega$ is an auxiliary point. Denoting

$$f_j^k = f(y_j) + (g_j^f)^T (x_k - y_j), \quad (16)$$

the linearization of our cost function is

$$\bar{f}_j(x) = f_j^k + (g_j^f)^T (x - x_k) \text{ for all } j \in J_f^k, \quad (17)$$

but we can rewrite the formulation Eq. (16) into its recursive form

$$f_j^{k+1} = f_j^k + (g_j^f)^T (x_{k+1} - x_k) \text{ for all } j \in J_f^k. \quad (18)$$

Moreover, we can employ this linearization for polyhedral approximation of the objective function (e.g. in Fig. 5)

$$\hat{f}^k(x) = \max\{\bar{f}_j(x) \mid j \in J_f^k\}, \quad (19)$$

and then we can define the improved polyhedral function \hat{H}^k

$$\hat{H}^k(x) = \hat{f}^k(x) - f(x_k) \text{ for all } x \in \mathbb{R}^n. \quad (20)$$

By employing the proximal bundle idea (the idea of adding a penalty to be able to limit the step length)

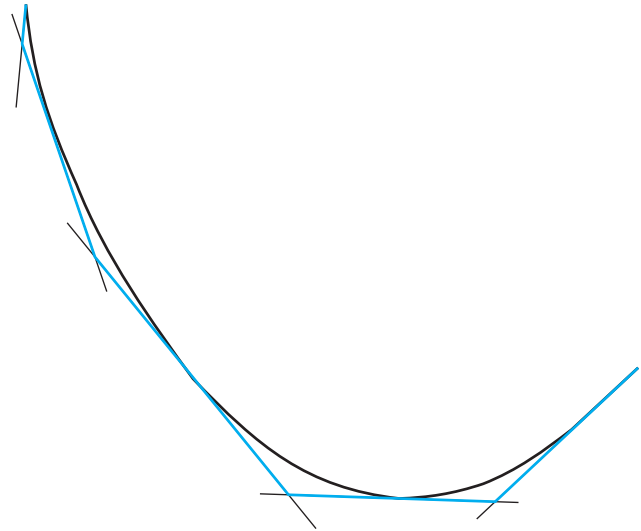


Fig. 5: Illustration of the linearization.

and after a series of adjustments, we can rewrite the whole problem Eq. (15) into its dual form

$$\left. \begin{array}{l} \min_{\lambda, \nu} \quad \frac{1}{2u_k} \left\| \sum_{j \in J_f^k} \lambda_j g_j^k + \sum_{i \in I} \nu_i C_i \right\|^2 \\ \quad \quad \quad + \sum_{j \in J_f^k} \lambda_j \alpha_{f,j}^k + \sum_{i \in I} \nu_i \alpha_{C,i}^k, \\ \text{subject to} \quad \sum_{j \in J_f^k} \lambda_j = 1, \\ \text{and} \quad \quad \quad \lambda, \nu \geq 0, \end{array} \right\} \quad (21)$$

where u_k is the weight, $\alpha_{f,j}^k$ are subgradient errors ($\alpha_{f,j}^k = f(x_k) - f_j^k$, for $j \in J_f^k$) and $\alpha_{C,i}^k$ are errors of the constraints subgradients ($\alpha_{C,i}^k = -C_i x_k + b_i$, for $i \in I = \{1, \dots, m\}$). We denote the solution of the problem Eq. (21) as λ^k and ν^k .

2) Subgradient Aggregation

There is still one hidden but equally important difficulty in the problem Eq. (21). Let us consider the index set J_f^k . How to choose this set? The simplest option seems to let

$$J_f^k = \{1, \dots, k\}. \quad (22)$$

However, this is not the case. Because, in every iteration step, the index set will enlarge, which causes larger and larger memory requirements. We have to find another way.

In 1985, Kiwiel presented the subgradient aggregation strategy. The idea is to aggregate the constraints made up by the previous subgradients. This strategy allows us to keep the quantity of constraints bounded. First, we need to scale the

multipliers for all $j \in J_f^k$ and to be able to do that, we denote $\lambda_f^k = \sum_{j \in J_f^k} \lambda_j^k$, then the aggregate subgradients can be presented.

Definition 10. We define the scaled multipliers for all $j \in J_f^k$ by:

$$\tilde{\lambda}_j^k = \begin{cases} \lambda_j^k / \lambda_f^k & \text{if } \lambda_f^k > 0, \\ 1/|J_f^k| & \text{if } \lambda_f^k = 0, \end{cases} \quad (23)$$

and the aggregate subgradients by:

$$p_f^k = \sum_{j \in J_f^k} \tilde{\lambda}_j^k g_j^f, \quad (24)$$

$$\tilde{f}_p^k = \sum_{j \in J_f^k} \tilde{\lambda}_j^k f_j^k, \quad (25)$$

$$\tilde{\alpha}_{f,p}^k = \sum_{j \in J_f^k} \tilde{\lambda}_j^k \alpha_{f,j}^k, \quad (26)$$

where $\alpha_{f,j}^k = f(x_k) - f_j^k$ is the linearization error. And finally the aggregate linearization is defined by:

$$\tilde{f}_p(x) = \tilde{f}_p^k + (p_f^k)^T(x - x_k). \quad (27)$$

The vector p_f^k is unknown at the beginning, but we can solve this lack by generating p_f^k recursively (see Eq. (31)). Let $x_1 \in \Omega$ be a feasible starting point, then we begin the algorithm with this settings: $y_1 = x_1$, $p_f^0 = g_1^f \in \partial f(y_1)$, $f_p^1 = f_1^1 = f(y_1)$ and $J_1 = \{1\}$ and we also define $\alpha_{f,p}^k = f(x_k) - f_p^k$.

Using stated equations, we are able to rewrite the problem Eq. (21) in suitable form to keep the index set bounded. Via restriction and dualization we obtain

$$\left. \begin{aligned} \min_{\lambda, \nu} \quad & \frac{1}{2u_k} \left\| \sum_{j \in J_f^k} \lambda_j g_j^k + \lambda_p p_f^{k-1} + \sum_{i \in I} \nu_i C_i \right\|^2 \\ & + \sum_{j \in J_f^k} \lambda_j \alpha_{f,j}^k + \lambda_p \alpha_{f,p}^k + \sum_{i \in I} \nu_i \alpha_{C,i}^k, \\ \text{subject to} \quad & \sum_{j \in J_f^k} \lambda_j + \lambda_p = 1, \\ \text{and} \quad & \lambda_j, \lambda_p, \nu \geq 0. \end{aligned} \right\} \quad (28)$$

Suppose that we found the Lagrange multipliers of the problem above λ_p^k , λ_j^k for all $j \in J_f^k$ and ν_i^k for all $i \in I = \{1, \dots, m\}$. First, we sum up the multipliers $\lambda_f^k = \lambda_p^k + \sum_{j \in J_f^k} \lambda_j^k$ and then we can scale the multipliers

$$\tilde{\lambda}_j^k = \begin{cases} \lambda_j^k / \lambda_f^k & \text{if } \lambda_f^k > 0, \\ 1/(|J_f^k|+1) & \text{if } \lambda_f^k = 0, \end{cases} \quad (29)$$

$$\tilde{\lambda}_p^k = \begin{cases} \lambda_p^k / \lambda_f^k & \text{if } \lambda_f^k > 0, \\ 1/(|J_f^k|+1) & \text{if } \lambda_f^k = 0, \end{cases} \quad (30)$$

and figure up the aggregate subgradients

$$p_f^k = \sum_{j \in J_f^k} \tilde{\lambda}_j^k g_j^f + \tilde{\lambda}_p^k p_f^{k-1}, \quad (31)$$

$$\tilde{f}_p^k = \sum_{j \in J_f^k} \tilde{\lambda}_j^k f_j^k + \tilde{\lambda}_p^k f_p^k, \quad (32)$$

where f_p^k is updated recursively by $f_p^k = f_p^{k-1} + t_L^k (p_f^k)^T d_k$.

Now we are able to choose the index set J_f^k bounded without any difficulties. Several types of selection exist and the simplest one (not necessarily the best one) is to keep the length of the index set bounded and replace the oldest element in J_f^k by the new one as it can be seen below:

$$\begin{aligned} J_f^5 = \{1, 2, 3, 4, 5\} &\Rightarrow J_f^6 = \{2, 3, 4, 5, 6\} \Rightarrow \dots \\ &\Rightarrow J_f^{10} = \{6, 7, 8, 9, 10\}. \end{aligned}$$

3) Nonconvexity

Let us recall that $\alpha_{f,j}^k = f(x_k) - f_j^k$ is the linearization error. If f is convex, then $\alpha_{f,j}^k \geq 0$ for all $j \in J_f^k$ and $\bar{f}_j(x) \leq f(x)$ for all $x \in \Omega$. It means that our linearization approximates the cost function f from below and $\alpha_{f,j}^k$ indicates how good our linearization is. But this is true only if the cost function f is convex. Unfortunately, in the nonconvex case, the inequality $\bar{f}_j(x) \leq f(x)$ is not valid at every $x \in \Omega$. The linear approximation can be above the cost function f and the linearization error may takes values less then zero. We can see this situation in Fig. 6.

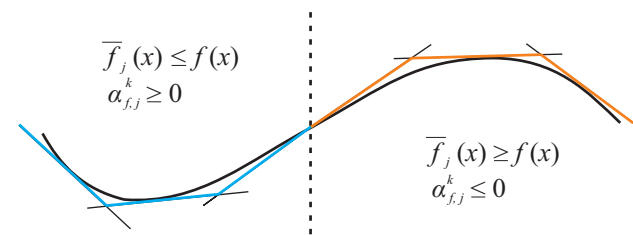


Fig. 6: Linear approximation of a nonconvex function.

So what to do when the cost function f is not convex? Due to this (we can say) difficulty, we need to generalize the subgradient error $\alpha_{f,j}^k$. We will use the subgradient locality measure. To achieve this, we will also need some information about the distance between the trial point y_j and the actual iteration point x_k .

Definition 11. Let us define the distance measure at every iteration k by:

$$s_j^k = \begin{cases} \|x_j - y_j\| + \sum_{i=j}^{k-1} \|x_{i+1} - x_i\| & \text{for } j = 1, \dots, k-1, \\ \|x_k - y_k\| & \text{for } j = k. \end{cases} \quad (33)$$

And now we are able to define the aforementioned subgradient locality measure.

Definition 12. At every iteration step k , the subgradient locality measure is defined by:

$$\beta_j^k = \max \left\{ |\alpha_{f,j}^k|, \gamma (s_j^k)^2 \right\} \text{ for all } j \in J_f^k, \quad (34)$$

where $\gamma \geq 0$ is the distance measure parameter, which is equal to zero, when the cost function is convex.

We choose the parameter γ heuristically. As it can be seen, we did not use the aggregate subgradients, which were described in the previous subsection. It is still important to be able to reduce the memory requirements, so we also define the aggregate version of the distance measures.

Definition 13. Denoting $s_f^1 = 0$, the aggregate distance measures at each iteration step k are defined by:

$$\left. \begin{aligned} \tilde{s}_f^k &= \sum_{j \in J_f^k} \lambda_j^k s_j^k + \tilde{\lambda}_p^k s_f^k, \\ s_f^{k+1} &= \tilde{s}_f^k + \|x_{k+1} - x_k\|, \end{aligned} \right\} \quad (35)$$

and the subgradient locality measures by:

$$\left. \begin{aligned} \beta_j^k &= \max \left\{ |\alpha_{f,j}^k|, \gamma (s_j^k)^2 \right\} \text{ for all } j \in J_f^k, \\ \tilde{\beta}_{f,p}^k &= \max \left\{ |\tilde{\alpha}_{f,p}^k|, \gamma (\tilde{s}_f^k)^2 \right\} \text{ for all } j \in J_f^k, \\ \tilde{\beta}_p^k &= \lambda_f^k \tilde{\beta}_{f,p}^k, \end{aligned} \right\} \quad (36)$$

where $\gamma \geq 0$ is the distance measure parameter. Set the parameter $\gamma = 0$, if f is convex.

4) Line Search

The descent direction $d_k = -\frac{1}{u_k} \left(\lambda_f^k p_f^k + \sum_{i \in I} \nu_i^k C_i \right)$ and the awaited decrease $v_k = \hat{f}^k(x_k + d_k) - f(x_k) < 0$ are known. But we do not know yet how far we can go in the direction d_k to evaluate the next value x_{k+1} . A solution to this problem was presented by Kiwiel in 1990 in his contribution [3].

After the initial setting of parameters, we are searching for the maximum value $t_L^k \in \langle 0, 1 \rangle$ that fulfils following conditions:

$$\text{a) } f(x_k + t_L^k d_k) \leq f(x_k) + m_L t_L^k v_k,$$

$$\text{b) } C(x_k + t_L^k d_k) \leq b,$$

$$\text{c) } t_L^k \geq \bar{t}.$$

If such a parameter exists, we obtain the so-called long serious step

$$\begin{aligned} x_{k+1} &= x_k + t_L^k d_k, \\ y_{k+1} &= x_{k+1}. \end{aligned} \quad (37)$$

If only the first condition a) holds and also $0 < t_L^k < \bar{t}$, we take the short serious step: choose $t_R^k \in \langle t_L^k, 1 \rangle$ and

$$\begin{aligned} x_{k+1} &= x_k + t_L^k d_k, \\ y_{k+1} &= x_k + t_R^k d_k, \end{aligned} \quad (38)$$

and finally, in the case when $t_L^k = 0$, we have the null step

$$\begin{aligned} x_{k+1} &= x_k, \\ y_{k+1} &= x_k + t_R^k d_k. \end{aligned} \quad (39)$$

5) Weight Update

One of the last but still very important questions is the choice of weight update u_k . We cannot keep u_k constant, because it could make some difficulties (e.g. if the parameter u_k is large, values $|v_k|$ and $\|d_k\|$ will be very small and therefore all the steps will be taken as the serious ones and the decrease will be small).

This seemingly small difficulty was also solved by Kiwiel in 1990. The whole weight update strategy can be found in the book [4] and in the article [3].

2.4. Description of the Proximal Bundle Method Algorithm

At the beginning of our algorithm, we need to set several parameters such as stopping tolerance $\varepsilon_S > 0$, which is used in the stopping criterion; the maximum number of stored subgradients $M_g \geq 2$; the line search parameters m_L, m_R, \bar{t} and distance parameters $\gamma_f > 0$.

In the next step of the algorithm, we should find multipliers λ_p^k, λ_j^k by solving the dual problem, see Eq. (28). In the first half of the algorithm, there is also the stopping criterion. We need to evaluate whether $w_k \leq \varepsilon_S$, where $w_k = \frac{1}{2} \|p_k\|^2 + \tilde{\beta}_{f,p}^k$, holds or not. If so, the algorithm stops and we obtain the desired result. Otherwise the algorithm continues by line search and after finding the step size, we make the linearization update.

The final part of the algorithm consists of the weight update and the index set updating $J_f^{k+1} = J_f^k \cup \{k+1\}$, but if the size of $J_f^{k+1} > M_g$, we set $J_f^{k+1} = J_f^{k+1} \setminus \left\{ \min j \mid j \in J_f^{k+1} \right\}$ as we exemplified

in the subsection Subgradient aggregation. Now it remains to increase the iteration counter k by 1 and to repeat the whole algorithm from the part with the dual problem.

3. Numerical Experiment

The proximal bundle method described in the previous sections will now be used to solve a model example. We chose the shape optimization of a discretized two-dimensional contact problem with Coulomb friction as the model example. Shape optimization is a part of the optimal control, in which the control variables are linked to the geometry of elastic bodies that are in contact. The aim of the problem on the lower level, which is contact problem with friction is to find the set of the state variables for the fixed vector of control variables. The state vector contains variables which describe the displacements, and the variables which describe the normal stress on the contact boundary. Hereafter, the contact problem with Coulomb friction will be considered as the state problem. The mapping describing the solution of the state problem (i.e. the vector of the displacements and the contact stresses) for the prescribed control variable is named as the control–state mapping. A typical feature of the contact shape optimization with Coulomb friction is its nonsmooth character due to the fact that the respective control–state mapping is typically nondifferentiable. Shape optimization of a discretized two-dimensional contact problem with Coulomb friction was considered in [1]. Shape optimization of a discretized three-dimensional contact problem with Coulomb friction was considered in [2]. Sensitivity analysis (computation of the subgradients of the minimized function) with help of calculus of Clarke (for 2D case) and calculus of Mordukhovich (for 3D case) was proposed in mentioned papers [1] and [2]. In this contribution, we approximate subgradients only numerically by the forward finite difference approximation.

Example 2. Now let us deal with the shape optimization of a discretized two-dimensional contact problem with Coulomb friction only briefly. Let \mathcal{J} be a cost function. The shape optimization problem is defined generally as follows

$$\left. \begin{aligned} & \text{minimize } \mathcal{J}(\alpha, \mathcal{S}(\alpha)), \\ & \text{subject to } \alpha \in U_{ad}, \end{aligned} \right\} \quad (40)$$

where the admissible set U_{ad} is given by:

$$U_{ad} := \left\{ \alpha \in \mathbb{R}^d : \begin{aligned} & 0 \leq \alpha^i \leq C_0, \quad i = 0, 1, \dots, d-1, \\ & |\alpha^{i+1} - \alpha^i| \leq C_1 h, \quad i = 0, 1, \dots, d-2, \\ & C_{21} \leq \text{meas } \Omega(\alpha) \leq C_{22} \end{aligned} \right\}.$$

We will try to smooth down the peaks of the normal contact stress distribution. To this aim, we should

minimize the max-norm of the discrete normal contact stress λ . The objective function \mathcal{J} , however, must be continuously differentiable to ensure that the composite function $\mathcal{J}(\alpha, \mathcal{S}(\alpha))$ is locally Lipschitz, so we will use the p power of the p norm of the vector λ with $p = 4$ as the objective function \mathcal{J} . The shape optimization problem then reads as follows:

$$\left. \begin{aligned} & \text{minimize } \|\lambda\|_4^4, \\ & \text{subject to } \alpha \in U_{ad}. \end{aligned} \right\} \quad (41)$$

The vector α denotes the control vector, u denotes the displacement and λ denotes the normal stress and mapping $\mathcal{S}: \alpha \in U_{ad} \subset \mathbb{R}^d \rightarrow (u, \lambda) \in \mathbb{R}^{3p}$ denotes the control–state mapping. Number d is the dimension of the control vector α , p is the number of the nodes of the discretized elastic body $\Omega(\alpha)$ and U_{ad} is the set of the admissible control variables. For more detailed description, see [1].

The shape of the elastic body $\Omega(\alpha)$, $\alpha \in U_{ad}$, is defined through a Bezier function F_α as follows (cf. Fig. 7):

$$\Omega(\alpha) = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 \in (0, a), F_\alpha(x_1) < x_2 < b\},$$

where the vector α contains the control points of the Bezier function F_α .

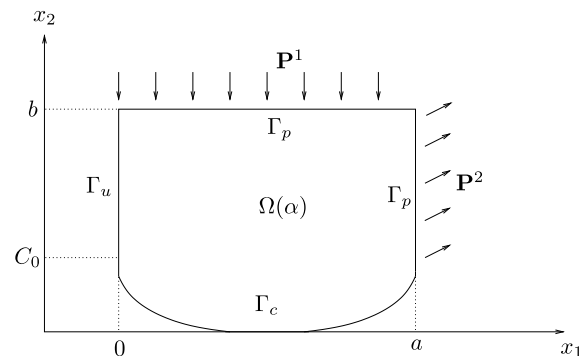


Fig. 7: The elastic body and applied loads.

From Fig. 7 we can also see the distribution of external pressures on the boundary Γ_P , given as $P^1 = (0; -200 \text{ MPa})$ on $(0, 2) \times \{1\}$, while $P^2 = (100 \text{ MPa}; 40 \text{ MPa})$ on $\{2\} \times (0, 1)$ Further, Γ_u is the part of the boundary where the zero displacements are prescribed.

The set of the admissible designs U_{ad} is specified as follows: $a = 2$, $b = 1$ and $C_0 = 0.75$, $C_1 = 1$, $C_{21} = 1.8$, $C_{22} = 2$. The Young modulus $E = 1 \text{ GPa}$ and the Poisson constant $\sigma = 0.3$ are used for the definition of the mapping \mathcal{S} . The value of the coefficient of the Coulomb friction is 0.25. The state problem on $\Omega(\alpha)$ is discretized by isoparametric quadrilateral elements of Lagrange type. The total number of nodes (vertices of

quadrilaterals) is 3976 for any $\alpha \in U_{ad}$. The dimension of the control vector α , generating the Bezier function and defining $\Omega(\alpha)$, is $d = 8$.

The stopping tolerance was set to $\varepsilon_S = 1 \cdot 10^{-6}$. This required precision was reached after 11 iterations. We depict the initial shape in Fig. 8 and the distribution of the von Mises stress in the loaded body in Fig. 9. Figure 10 shows the optimal shape and Fig. 11 shows the von Mises stress in the deformed optimal body. Finally, figures Fig. 12 and Fig. 13 compare the contact normal stresses for the initial and optimal shape, respectively. Note that during the optimization process the initial value $\mathcal{J}(\alpha_0) = 2.8612 \cdot 10^{11}$ of the cost functional dropped to $\mathcal{J}(\alpha_{opt}) = 1.0695 \cdot 10^{11}$. The decrease of the peak stress is also quite significant. The experiment was carried out in Mathworks Matlab.

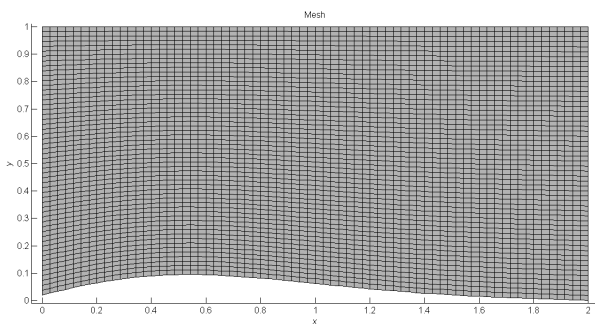


Fig. 8: Example, initial design – the initial shape of the body.

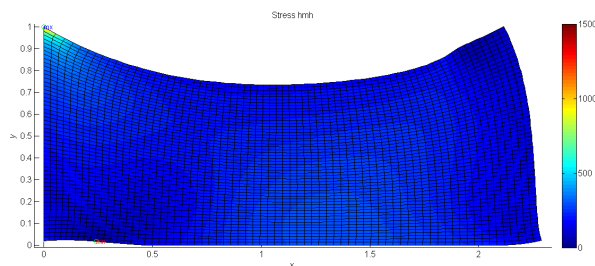


Fig. 9: Example, initial design – the distribution of the von Mises stress.

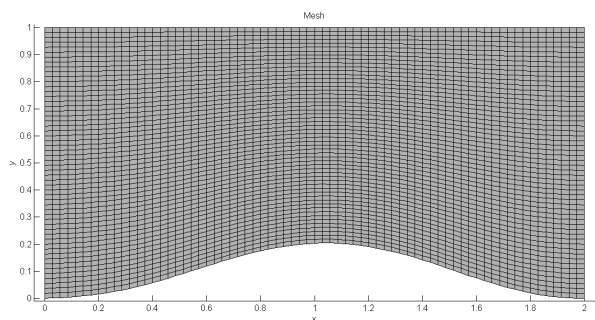


Fig. 10: Example, optimal design – the optimal shape of the body.

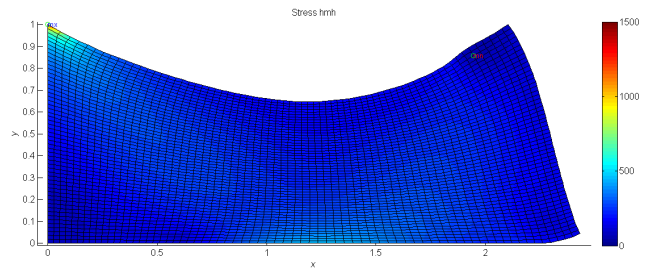


Fig. 11: Example, optimal design – the distribution of the von Mises stress.

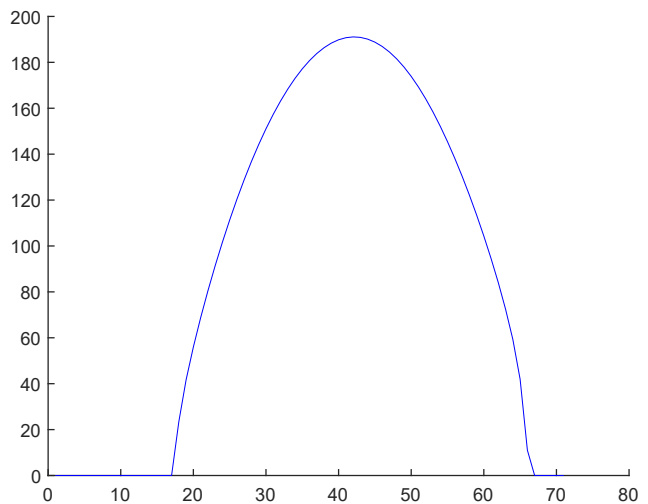


Fig. 12: Example, normal stress for the initial design.

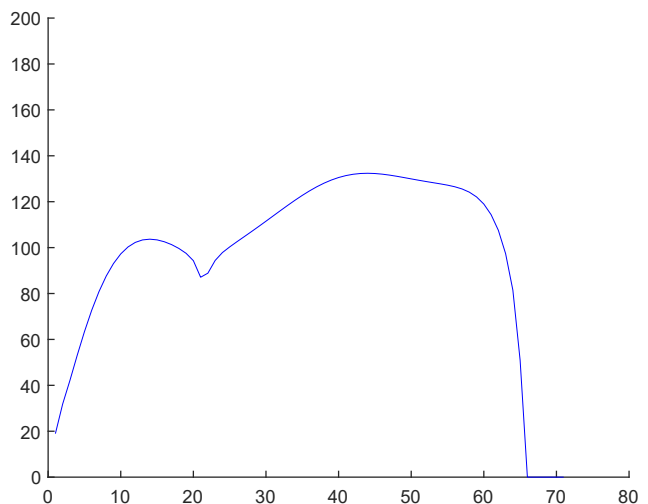


Fig. 13: Example, normal stress for the optimal design.

4. Conclusion

In this contribution we have briefly introduced the proximal bundle method for nonsmooth optimization problems with linear constraints $Cx \leq b$ and with some simple bounds x_{min}, x_{max} . We also outlined the imple-

mented algorithm, which was employed to solve our model example. We tried to deal with the shape optimization of a discretized two-dimensional contact problem with Coulomb friction. To find a solution, we had to solve nonsmooth minimization problem with linear and box constraints. To be able to solve even more complicated nonsmooth problems, we should enhance our algorithm with solver for nonlinear constraints and to improve the line search part.

Acknowledgment

The work is partially supported by Grant of SGS No. SP2017/122, VSB–Technical University of Ostrava, Czech Republic.

References

- [1] BEREMLIJSKI, P., J. HASLINGER, M. KOCVARA and J. OUSRATA. Shape optimization in contact problems with Coulomb friction. *SIAM Journal on Optimization*. 2002, vol. 13, iss. 2, pp. 561–587. ISSN 0233-1934. DOI: 10.1137/S1052623401395061.
- [2] BEREMLIJSKI, P., J. HASLINGER, M. KOCVARA, R. KUCERA and J. OUSRATA. Shape optimization in three-dimensional contact problems with Coulomb friction. *SIAM Journal on Optimization*. 2009, vol. 20, iss. 1, pp. 416–444. ISSN 1052-6234. DOI: 10.1137/080714427.
- [3] KIWIEL, K. C. Proximity Control in Bundle Methods for Convex Nondifferentiable Minimization. *Mathematical Programming*. 1990, vol. 46, iss. 1, pp. 105–122. ISSN 0025-5610. DOI: 10.1007/BF01585731.
- [4] MAKELA, M. M. and P. NEITTAANMAKI. *Nonsmooth Optimization, Analysis and Algorithms with Application to Optimal Control*. Singapore: World Scientific, 1992. ISBN 981-0207735.
- [5] OUSRATA, J., M. KOCVARA and J. ZOWE. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints, Theory, Applications and Numerical Results*. Berlin: Springer Science+Business Media, 1998. ISBN 978-1-4419-4804-5.
- [6] ZOWE, J. Nondifferentiable Optimization. *Computational Mathematical Programming*. Berlin: Springer Berlin Heidelberg, 1985. ISBN 978-3-642-82452-4. DOI: 10.1007/978-3-642-82450-0_12.

About Authors

Nikola PLIVOVA was born in Karvina, Czech Republic. She received her M.Sc. in Applied Mathematics in 2015. Her research interests include nonsmooth optimization, Clarke calculus and marginally compressive sampling.

Petr BEREMLIJSKI was born in Ostrava, Czech Republic. He received his M.Sc. in Computer Science in 1999. He finished Ph.D. in 2007 at VSB–Technical University of Ostrava, Associate Professor since 2016. His research interests include nonsmooth optimization, Clarke and Mordukhovich calculus and shape optimization for contact problems with friction.