Reports of the Department of Mathematical Information Technology
Series B. Scientific Computing
No. B. 3/2007

Limited Memory Bundle Algorithm for Inequality Constrained Nondifferentiable Optimization

Napsu Karmitsa   Marko M. Mäkelä
Montaz M. Ali

University of Jyväskylä
Department of Mathematical Information Technology
P.O. Box 35 (Agora)
FI–40014 University of Jyväskylä
FINLAND
fax +358 14 260 2731
http://www.mit.jyu.fi/

ISBN 978-951-39-2785-1
ISSN 1456-436X

Limited Memory Bundle Algorithm for Inequality Constrained Nondifferentiable Optimization[*]

Napsu Karmitsa[†] Marko M. Mäkelä[‡] Montaz M. Ali[§]

Abstract

Many practical optimization problems involve nonsmooth (that is, not necessarily differentiable) functions of hundreds or thousands of variables with various constraints. In this paper, we describe a new efficient adaptive limited memory interior point bundle method for large, possible nonconvex, nonsmooth inequality constrained optimization. The method is a hybrid of the nonsmooth variable metric bundle method and the smooth limited memory variable metric method, and the constraint handling is based on the primal-dual feasible direction interior point approach. The preliminary numerical experiments to be presented confirm the effectiveness of the method.

Keywords: Nonsmooth optimization, large-scale problems, constrained optimization, bundle methods, limited memory methods, feasible direction interior point methods.

1 Introduction

In this paper, we propose a new adaptive limited memory interior point bundle algorithm for solving large inequality constrained nonsmooth optimization problems. We write this problem as

$$\begin{cases} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & g_i(\boldsymbol{x}) \leq 0, \quad \text{for } i = 1, \ldots, p, \end{cases} \tag{1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the constraint functions $g_i : \mathbb{R}^n \to \mathbb{R}$ for each $i \in \mathcal{P} = \{1, \ldots, p\}$ are supposed to be locally Lipschitz continuous and

[*]The work was financially supported by University of the Witwatersrand, South Africa; University of Jyväskylä, Finland; and University of Turku, Finland.

[†]Department of Mathematical Information Technology, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, `hamasi@mit.jyu.fi`

[‡]Department of Mathematics, University of Turku, FI-20014 Turku, Finland, `makela@utu.fi`

[§]School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa, `mali@cam.wits.ac.za`

the number of variables n is supposed to be large. Note that no differentiability or convexity assumptions are made.

Many practical optimization problems involve nonsmooth functions with large amounts of variables (see, e.g., [2, 3, 30]). These kind of problems are in general difficult to solve even when they are unconstrained. The direct application of smooth gradient-based methods to nonsmooth problems usually leads to a failure in convergence, in optimality conditions, or in gradient approximation (see, e.g., [6, 24]). On the other hand, derivative free methods like genetic algorithms (see, e.g., [10]) or Powell's method (see, e.g., [7]) may be unreliable and become inefficient whenever the dimension of the problem increases. Thus, special tools for solving large-scale nonsmooth optimization problems are needed.

At the moment, different variants of bundle methods (see, e.g., [20, 31, 36]) are regarded as the most effective and reliable globally convergent methods for nonsmooth optimization. The basic idea of these methods is to approximate the subdifferential [6] of the nonsmooth objective by gathering subgradients (generalized gradients) from previous iterations into a bundle. A descent search direction can then be found as a solution of a quadratic subproblem and the global convergence of the methods, with limited number of stored subgradients, can be guaranteed by using a subgradient aggregation strategy [20], which accumulates information from previous iterations.

Naturally, the presence of constraints makes nonsmooth optimization problems even more complex. The problems with simple constraints (such as bound or linear constraints) may be solved by including the constraints directly to the quadratic subproblem (see, e.g., [21, 22]) and the more generally constrained problems may be solved by applying bundle methods to an equivalent unconstrained problem with an *exact penalty* objective (see, e.g., [19, 23]) or by using bundle methods with so-called *improvement function* (see, e.g., [20, 25, 31, 35]) or with *filter* [8, 18]. However, all these methods are suitable only for relatively small problems and most of them (excluding [21, 31]) are capable of handling only convex problems.

In [11, 13, 14] we have proposed a limited memory bundle method for general, possibly nonconvex, nonsmooth large-scale unconstrained optimization. The method is a hybrid of the variable metric bundle methods [27, 38] and the limited memory variable metric methods (see, e.g., [5, 9, 33]), where the first ones have been developed for small- and medium-scale nonsmooth optimization and the latter ones, on the contrary, for smooth large-scale optimization. In [12] the variant of the method suitable for bound constrained problems was introduced.

In this paper, we combine the adaptive limited memory bundle method [11] with a modification of the feasible direction interior point method by Herskovits and Santos [15, 16] in order to make the method suitable for solving more generally constrained problems. We have used the approach of the feasible direction interior point method because it does not involve penalty or barrier functions, active set strategies, or quadratic subproblems but merely a solution of two internal linear systems with the same matrix at each iteration. By this way, the computational demand of the method is kept relatively low and we obtain a method suitable for solving large-scale problems.

Our method is feasible, in that, given an initial point that satisfy the constraints, it construct a sequence of points which all satisfy them as well. This kind of interior-point approach is essential in case the objective function and/or the constraint functions are not defined in infeasible points. Furthermore, it can be an advantage in many industrial applications, where function evaluation may be very expensive. Since any intermediate solution can be employed, the iterations can be stopped whenever the result is satisfactory.

The rest of this paper is organized as follows. In the following section, we first recall some basic results and definitions of nonsmooth analysis. In Section 3, we describe the adaptive limited memory interior point bundle method for inequality constrained optimization and, in Section 4, we analyze its convergence properties. In Section 5, some preliminary results of numerical experiments are presented and finally, in Section 6, we conclude. A detailed description of limited memory matrix updating is given in Appendix.

2 Background and Optimality Conditions

In this section we give the optimality conditions for nonsmooth inequality constrained optimization problem (1). Thereby, we first recall some basic definitions and results from nonsmooth analysis based on Clarke [6]. For details and proofs we refer to [6, 31].

In what follows, we assume that all the functions considered are locally Lipschitz continuous and by *Rademacher's Theorem* a locally Lipschitz continuous function is differentiable almost everywhere. Thus, we can use the following theorem to define the *subdifferential*.

THEOREM 2.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at $\boldsymbol{x} \in \mathbb{R}^n$. Then the subdifferential of f at $\boldsymbol{x}$ is a set $\partial f(\boldsymbol{x})$ of vectors $\boldsymbol{\xi}_f \in \mathbb{R}^n$ such that*

$$\partial f(\boldsymbol{x}) = \operatorname{conv} \{ \boldsymbol{\xi}_f \in \mathbb{R}^n \mid \textit{there exists } (\boldsymbol{x}_i) \subset \mathbb{R}^n \setminus \Omega_f \textit{ such that } \boldsymbol{x}_i \to \boldsymbol{x} \textit{ and } \nabla f(\boldsymbol{x}_i) \to \boldsymbol{\xi}_f \}.$$

Here "conv" denotes the convex hull of the set and Ω_f is a set where f fails to be differentiable. Each vector $\boldsymbol{\xi}_f \in \partial f(\boldsymbol{x})$ is called a subgradient *of f at $\boldsymbol{x}$.*

The subdifferential $\partial f(\boldsymbol{x})$ is a nonempty, convex, and compact set such that $\partial f(\boldsymbol{x}) \subset B(0; L)$, where $L > 0$ is the Lipschitz constant of f at $\boldsymbol{x}$.

In iterative optimization methods it is necessary to find a direction such that the objective function values decrease when moving in that direction. Next we define a descent direction.

DEFINITION 2.2. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is said to be a *descent direction* for $f : \mathbb{R}^n \to \mathbb{R}$ at $\boldsymbol{x} \in \mathbb{R}^n$, if there exists $\varepsilon > 0$ such that for all $t \in (0, \varepsilon]$

$$f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x}).$$

LEMMA 2.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\boldsymbol{x} \in \mathbb{R}^n$. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is a descent direction for f at $\boldsymbol{x}$ if $\boldsymbol{\xi}_f^T \boldsymbol{d} < 0$ for all $\boldsymbol{\xi}_f \in \partial f(\boldsymbol{x})$.*

In constrained optimization it is not enough to find any descent direction, since we are not allowed to violate the constraints. Thus, we need to define the feasible direction.

DEFINITION 2.4. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is said to be a *feasible direction* for problem (1), if there exists $\varepsilon > 0$ such that for all $t \in (0, \varepsilon]$

$$\boldsymbol{x} + t\boldsymbol{d} \in S,$$

where $S = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \le 0 \text{ for all } i \in \mathcal{P}\}$ is the *feasible region* for problem (1).

LEMMA 2.5. *Let $g_i : \mathbb{R}^n \to \mathbb{R}$ for each $i \in \mathcal{P}$ be locally Lipschitz continuous functions at $\boldsymbol{x} \in S$. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is a feasible direction for problem (1) if $\boldsymbol{\xi}_{g_i}^T \boldsymbol{d} < 0$ for all $\boldsymbol{\xi}_{g_i} \in \partial g_i(\boldsymbol{x})$ such that $g_i(\boldsymbol{x}) = 0$.*

PROOF. Follows directly from the fact that by Lemma 2.3 direction $\boldsymbol{d}$ is a descent direction for all g_i such that $g_i(\boldsymbol{x}) = 0$. □

In order to formulate necessary Karush-Kuhn-Tucker (KKT) type optimality conditions, we first need to define the following regularity assumption.

DEFINITION 2.6. Problem (1) satisfies the *Cottle constraint qualification* at $\boldsymbol{x}$, if either $g_i(\boldsymbol{x}) < 0$ for all $i \in \mathcal{P}$ or $\mathbf{0} \notin \operatorname{conv}\{\partial g_i(\boldsymbol{x}) \mid g_i(\boldsymbol{x}) = 0\}$.

THEOREM 2.7. (KKT optimality condition.) *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for each $i \in \mathcal{P}$ be locally Lipschitz continuous functions at $\boldsymbol{x} \in \mathbb{R}^n$ and suppose that problem (1) satisfies the Cottle constraint qualification. If $\boldsymbol{x}$ is a local minimum of (1), then there exist Lagrange multipliers $\mu_i \ge 0$ such that $\mu_i g_i(\boldsymbol{x}) = 0$ for all $i \in \mathcal{P}$ and*

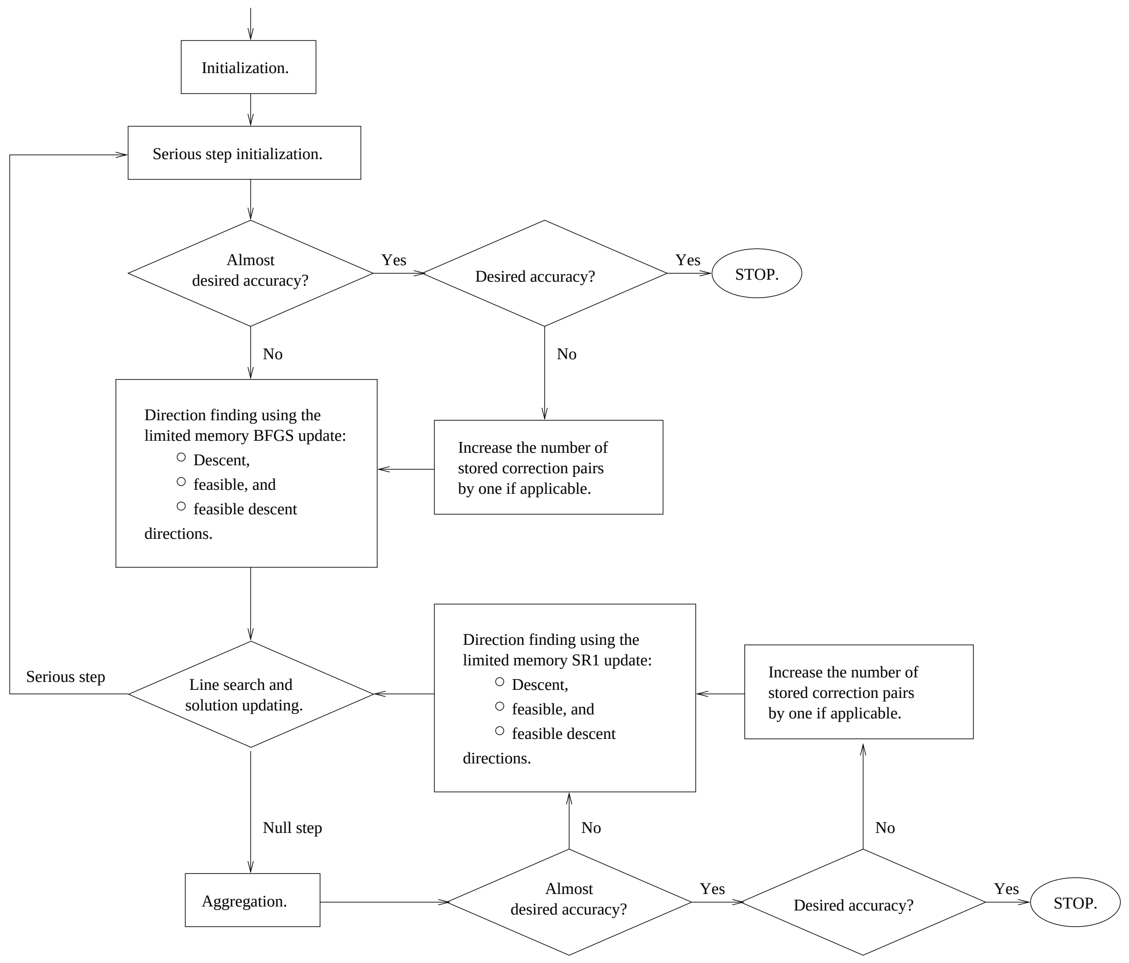
$$\mathbf{0} \in \partial f(\boldsymbol{x}) + \sum_{i \in \mathcal{P}} \mu_i \partial g_i(\boldsymbol{x}).$$

A point $\boldsymbol{x}$ is said to be a *KKT point* associated to problem (1) if it is feasible and it satisfies the KKT optimality condition (Theorem 2.7). If all the functions f and g_i ($i \in \mathcal{P}$) are convex, the KKT optimality condition is sufficient and the KKT point is a global minimum for problem (1).

In what follows we denote by $L(\boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i \in \mathcal{P}} \mu_i g_i(\boldsymbol{x})$ the Lagrangian of problem (1) and by $\partial L(\boldsymbol{x}, \boldsymbol{\mu})$ its subdifferential. Moreover, we denote by $\partial \hat{L}(\boldsymbol{x}, \boldsymbol{\mu}) = \partial f(\boldsymbol{x}) + \sum_{i \in \mathcal{P}} \mu_i \partial g_i(\boldsymbol{x})$. Note that we have the following inclusion $\partial L(\boldsymbol{x}, \boldsymbol{\mu}) \subset \partial \hat{L}(\boldsymbol{x}, \boldsymbol{\mu})$ (see, e.g., [31]).

3 Method

In this section, we describe the adaptive limited memory interior point bundle method for inequality constrained large-scale nonsmooth optimization. We start by giving a simple flowchart (in Figure 1) to point out the basic ideas of the algorithm.



Initialization.
Serious step initialization.
Almost desired accuracy?
Yes
Desired accuracy?
Yes
STOP.
No
No
Direction finding using the limited memory BFGS update:
Descent,
feasible, and
feasible descent
directions.
Increase the number of stored correction pairs by one if applicable.
Serious step
Line search and solution updating.
Direction finding using the limited memory SR1 update:
Descent,
feasible, and
feasible descent
directions.
Increase the number of stored correction pairs by one if applicable.
Null step
No
No
Aggregation.
Almost desired accuracy?
Yes
Desired accuracy?
Yes
STOP.

Figure 1: Adaptive limited memory interior point bundle method.

The limited memory interior point bundle method is characterized by the usage of null steps together with the aggregation of subgradients. At each iteration we first solve a linear system of KKT type optimality conditions in order to find a descent direction for the objective. After that, we (by means of computing a feasible direction) perturb the linear system to deflect the descent direction to a feasible descent direction. Finally, the line search is performed along this direction either to obtain a new interior point with lower objective or to take a null step. Using null steps gives sufficient information about the nonsmooth objective in the case the search direction is not “good enough”. On the other hand, a simple aggregation of subgradients guarantees the convergence of the aggregate subgradients to zero and makes it possible to evaluate a termination criterion.

The limited memory approach (see, e.g., [5, 9, 33]) is utilized both in the calculation of the search direction and the aggregate values. The idea of limited memory matrix updating is that instead of storing the large matrices we store a certain (usually small constant) number of vectors, so-called correction pairs, obtained at the previous iterations of the algorithm and we use these vectors to implicitly define the variable metric matrices. When the storage space available is used up, the oldest correction pairs are deleted to make room for new ones. In the adaptive limited memory bundle method [11] the number of stored correction pairs may change during the computation. This means that we can start the optimization with a small number of stored correction pairs and then, when we are closer to the optimal point, the number of stored correction pairs may be increased until some upper limit is achieved. The aim of this adaptability is to improve the accuracy of the basic method without loosing much from efficiency, that is, without increasing computational costs too much.

3.1 Limited memory interior point bundle method.

We now describe with more details the limited memory interior point bundle algorithm for solving nonsmooth optimization problems of type (1). The algorithm to be presented generates a sequence of basic points $(\boldsymbol{x}_k) \subset \operatorname{int} S$ together with a sequence of auxiliary points $(\boldsymbol{y}_k) \subset \operatorname{int} S$. A new iteration point $\boldsymbol{x}_{k+1}$ and a new auxiliary point $\boldsymbol{y}_{k+1}$ are produced using a special constrained line search procedure such that

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + t_L^k \boldsymbol{d}_k \qquad && \text{and} \\ \boldsymbol{y}_{k+1} &= \boldsymbol{x}_k + t_R^k \boldsymbol{d}_k, \qquad && \text{for } k \geq 1 \end{aligned} \tag{2}$$

with $\boldsymbol{y}_1 = \boldsymbol{x}_1 \in \operatorname{int} S$, where $t_R^k \in (0, t_{max}]$ and $t_L^k \in [0, t_R^k]$ are step sizes, $t_{max} \geq 1$ is the upper bound for the step size, and $\boldsymbol{d}_k$ is a search direction.

A necessary condition for a serious step is to have $t_R^k = t_L^k > 0$,

$$\begin{aligned} f(\boldsymbol{y}_{k+1}) &\leq f(\boldsymbol{x}_k) - \varepsilon_L t_R^k w_k^1, \qquad && \text{and} \\ g_i(\boldsymbol{y}_{k+1}) &< 0 && \text{for all } i \in \mathcal{P}, \end{aligned} \tag{3}$$

where $\varepsilon_L \in (0, 1/2)$ is a line search parameter and $w_k^1 > 0$ represents the desirable amount of descent of f at $\boldsymbol{x}_k$. If condition (3) is satisfied, we have $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1}$ and a serious step is taken.

Otherwise, we take a null step. In this case, the usage of special line search procedure guarantees that we have $t_R^k > t_L^k = 0$,

$$\begin{aligned} -\beta_{L,k+1} - \tilde{\boldsymbol{\xi}}_{L,k}^T D_k \boldsymbol{\xi}_{L,k+1} &\geq -\varepsilon_R w_k^1, \qquad && \text{and} \\ g_i(\boldsymbol{y}_{k+1}) &< 0 && \text{for all } i \in \mathcal{P}. \end{aligned} \tag{4}$$

Here $\varepsilon_R \in (\varepsilon_L, 1/2)$ is a line search parameter, $\boldsymbol{\mu}_k$ is a dual variable, $\tilde{\boldsymbol{\xi}}_{L,k} \in \partial L(\boldsymbol{x}_k, \boldsymbol{\mu}_k)$ is the aggregate subgradient of the Lagrangian, $\boldsymbol{\xi}_{L,k+1} \in \partial L(\boldsymbol{y}_{k+1}, \boldsymbol{\mu}_k)$ is the new auxiliary subgradient of the Lagrangian, D_k is a positive definite limited memory variable metric update that, in smooth case, approximates the inverse of the Hessian of the Lagrangian associated to problem (1), and $\beta_{L,k+1}$ is the subgradient locality measure [26, 32] defined by

$$\beta_{L,k+1} = \max\{|L(\boldsymbol{x}_k, \boldsymbol{\mu}_k) - L(\boldsymbol{y}_{k+1}, \boldsymbol{\mu}_k) + (\boldsymbol{y}_{k+1} - \boldsymbol{x}_k)^T \boldsymbol{\xi}_{L,k+1})|, \gamma \|\boldsymbol{y}_{k+1} - \boldsymbol{x}_k\|^2\}. \tag{5}$$

In the case of a null step, we have $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ but information about the objective function is increased because we store the auxiliary point $\boldsymbol{y}_{k+1}$ and the corresponding auxiliary subgradient $\boldsymbol{\xi}_{L,k+1}$.

Search direction. The search direction $\boldsymbol{d}_k$ is calculated using a modification of the feasible direction interior point method [15, 16]. First, we formulate the necessary KKT type optimality conditions for problem (1) by

$$\tilde{\boldsymbol{\xi}}_f + \tilde{\boldsymbol{\xi}}_{\boldsymbol{g}}\boldsymbol{\mu} = \mathbf{0}, \tag{6}$$

$$G(\boldsymbol{x})\boldsymbol{\mu} = \mathbf{0}, \tag{7}$$

$$\boldsymbol{g}(\boldsymbol{x}) \le \mathbf{0}, \tag{8}$$

$$\boldsymbol{\mu} \ge \mathbf{0}, \tag{9}$$

where $\tilde{\boldsymbol{\xi}}_f$ is the aggregate subgradient of the objective function, $\tilde{\boldsymbol{\xi}}_{\boldsymbol{g}}$ is the $p \times n$ matrix of the aggregate subgradients of the constraint functions, $\boldsymbol{g}(\boldsymbol{x})$ denotes the p -vector of constraints values, and $G(\boldsymbol{x})$ denotes a diagonal $p \times p$ -matrix such that $G(\boldsymbol{x}) = \operatorname{diag}[g_1(\boldsymbol{x}), \dots, g_p(\boldsymbol{x})]$. Moreover, $\boldsymbol{\mu}$ denotes the p -vector of dual variables.

A quasi-Newton's -type iteration to solve the system of equations (6), (7) can be defined by

$$\begin{bmatrix} B_k & \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k} \\ \Lambda_k \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}^T & G(\boldsymbol{x}_k) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{k+1}^{\alpha} - \boldsymbol{x}_k \\ \boldsymbol{\mu}_{k+1}^{\alpha} - \boldsymbol{\mu}_k \end{bmatrix} = - \begin{bmatrix} \tilde{\boldsymbol{\xi}}_{f,k} + \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}\boldsymbol{\mu}_k \\ G(\boldsymbol{x}_k)\boldsymbol{\mu}_k \end{bmatrix}, \tag{10}$$

where $(\boldsymbol{x}_k, \boldsymbol{\mu}_k)$ is the starting point of the iteration, $(\boldsymbol{x}_{k+1}^{\alpha}, \boldsymbol{\mu}_{k+1}^{\alpha})$ is the new estimate (index α refers to descent direction calculations), Λ_k denotes the diagonal $p \times p$ -matrix such that $\Lambda_k = \operatorname{diag}[\mu_{1,k}, \dots, \mu_{p,k}]$, and B_k is a positive definite limited memory variable metric update ($B_k = D_k^{-1}$). We solve the system of equations (6), (7) such that (8), (9) are verified at each iteration.

The iterations in (10) are modified such that, for the given interior pair $(\boldsymbol{x}_k, \boldsymbol{\mu}_k)$, we obtain a new interior estimate with a better objective. For this purpose, a primal direction $\boldsymbol{d}_k^{\alpha} = \boldsymbol{x}_{k+1}^{\alpha} - \boldsymbol{x}_k$ is defined. Now, we can write (10) as a linear system in $\boldsymbol{d}_k^{\alpha}$ and $\boldsymbol{\mu}_{k+1}^{\alpha}$ by

$$B_k \boldsymbol{d}_k^{\alpha} + \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}\boldsymbol{\mu}_{k+1}^{\alpha} = -\tilde{\boldsymbol{\xi}}_{f,k} \tag{11}$$

$$\Lambda_k \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}^T \boldsymbol{d}_k^{\alpha} + G(\boldsymbol{x}_k)\boldsymbol{\mu}_{k+1}^{\alpha} = \mathbf{0} \tag{12}$$

and we have the following result for $\boldsymbol{d}_k^{\alpha}$:

LEMMA 3.1. *Suppose that $\boldsymbol{\mu}_k > \mathbf{0}$ for all $k \ge 1$. The direction $\boldsymbol{d}_k^{\alpha}$ defined by (11) and (12) satisfies*

$$(\boldsymbol{d}_k^{\alpha})^T \tilde{\boldsymbol{\xi}}_{f,k} \le -(\boldsymbol{d}_k^{\alpha})^T B_k \boldsymbol{d}_k^{\alpha} \qquad \textit{for all } k \ge 1.$$

PROOF. By multiplying both sides of (11) by $\boldsymbol{d}_k^{\alpha}$, we obtain

$$(\boldsymbol{d}_k^{\alpha})^T \tilde{\boldsymbol{\xi}}_{f,k} = -(\boldsymbol{d}_k^{\alpha})^T B_k \boldsymbol{d}_k^{\alpha} - (\boldsymbol{d}_k^{\alpha})^T \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}\boldsymbol{\mu}_{k+1}^{\alpha}.$$

Now, by using (12), this can be written

$$(\boldsymbol{d}_k^\alpha)^T\tilde{\boldsymbol{\xi}}_{f,k} = -(\boldsymbol{d}_k^\alpha)^T B_k \boldsymbol{d}_k^\alpha + (\boldsymbol{\mu}_{k+1}^\alpha)^T \Lambda_k^{-1} G(\boldsymbol{x}_k)\boldsymbol{\mu}_{k+1}^\alpha$$

and the result follows from the fact that $\Lambda_k^{-1}G(\boldsymbol{x}_k)$ is negative definite for all k due to strict feasibility of $\boldsymbol{x}_k$. □

Due to the preceding lemma, the positiveness of $\boldsymbol{\mu}_k$, and the positive definiteness of B_k used in our proposal, we have

$$\tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k^\alpha \leq 0 \qquad \text{for all } k \geq 1.$$

Thus, by Lemma 2.3 and the fact that $\tilde{\boldsymbol{\xi}}_{f,k}$ is a convex combination of the previous subgradients (the proof is similar to that of Lemma 3.2 in [38]) direction $\boldsymbol{d}_k^\alpha$ seems to be a suitable choice for a descent direction. Nevertheless, from (12) it comes out that when any constraint goes to zero $\boldsymbol{d}_k^\alpha$ tends to be orthogonal to the aggregate subgradient of that constraint and we may obtain an infeasible direction. Thus, we deflect $\boldsymbol{d}_k^\alpha$ towards the interior of the feasible region by means of the vector $\boldsymbol{d}_k^\beta$ defined by the linear system

$$B_k \boldsymbol{d}_k^\beta + \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k} \boldsymbol{\mu}_{k+1}^\beta = \mathbf{0} \tag{13}$$

$$\Lambda_k \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}^T \boldsymbol{d}_k^\beta + G(\boldsymbol{x}_k)\boldsymbol{\mu}_{k+1}^\beta = -\boldsymbol{\mu}_k. \tag{14}$$

Now, the search direction can be calculated by

$$\boldsymbol{d}_k = \boldsymbol{d}_k^\alpha + \rho_k \boldsymbol{d}_k^\beta. \tag{15}$$

Here the deflection bound $\rho_k > 0$ (see (17) and (18)) is selected such that the condition

$$\tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k \leq \nu \tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k^\alpha$$

with predefined $\nu \in (0,1)$ is satisfied (see Lemma 4.3 in [15]). Thus, we have $\tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k \leq 0$ and, from (12), (14), and (15), we obtain $\tilde{\boldsymbol{\xi}}_{g_i,k}^T \boldsymbol{d}_k = -\rho_k < 0$ for all active constraints (i.e., for constraints with $g_i(\boldsymbol{x}_k) = 0$, $i \in \mathcal{P}$). Therefore, due to Lemmas 2.3 and 2.5 the direction $\boldsymbol{d}_k$ seems to be suitable choice for a feasible descent direction.

There are different possibilities of updating $\boldsymbol{\mu}_k$ (see, e.g., [15] or [1] in a slightly different contents). In this work we have adopted and modified the combination of two used in [15], since these updating rules guarantee the boundedness of $\boldsymbol{\mu}_k$ (see [15]). More specifically, we initialize $\mu_{i,1}$ for all $i \in \mathcal{P}$ by $\mu_{i,1} = \min\{-1/g_i(\boldsymbol{x}_1), \mu_{max}\}$, where $\mu_{max} > 0$ is a predefined upper limit. Note that we assume the starting point $\boldsymbol{x}_1$ to be strictly feasible and, thus, $g_i(\boldsymbol{x}_1) < 0$ for all $i \in \mathcal{P}$. In the subsequent iterations we set $\mu_{i,k+1} = \max\{\mu_{i,k+1}^\alpha, \epsilon\|\boldsymbol{d}_k^\alpha\|^2\}$ with some $\epsilon > 0$, if the next step is a serious step, and $\mu_{i,k+1} = \mu_{i,k}$, if the next step is a null step. If $g_i(x) \geq g_{max}$ and $\mu_{i,k+1} < \mu_{min}$ for some predefined $\mu_{min} > 0$ and $g_{max} < 0$, we set $\mu_{i,k+1} = \mu_{min}$.

The limited memory interior point bundle algorithm uses, for the direction determination, the original subgradients after the serious step and the aggregate subgradients after the null step. The aggregation procedure (see Step 5 in Algorithm 3.1) is similar to that of the original variable metric bundle methods [27, 38] except two matters. First, we use the subgradients and the variable metric approximations of the Lagrangian instead of those of the objective function and, secondly, the variable metric updates are calculated using limited memory approach, namely the limited memory BFGS and SR1 updates: If the previous step was a null step, the matrix D_k (i.e., B_k^{-1}) is formed using the limited memory SR1 update (see Appendix, (30)), since this update formula gives us a possibility to preserve the boundedness and some other properties of generated matrices that are needed to guarantee the convergence of aggregate subgradients to zero. Otherwise, since these properties are not required after a serious step, the more efficient limited memory BFGS update (see Appendix, (29)) is employed. The individual updates that would violate positive definiteness are skipped (for more details, see [11, 13, 14] and Appendix).

Algorithm. We now present a model algorithm for the adaptive limited memory interior point bundle method for solving the inequality constrained minimization problems of type (1). In what follows, we assume that at every point $\boldsymbol{x} \in \mathbb{R}^n$ we can evaluate the values $f(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$ for all $i \in \mathcal{P}$ and the corresponding arbitrary subgradients $\boldsymbol{\xi}_f \in \partial f(\boldsymbol{x})$, $\boldsymbol{\xi}_{g_i} \in \partial g_i(\boldsymbol{x})$, and $\boldsymbol{\xi}_L \in \partial L(\boldsymbol{x}, \boldsymbol{\mu})$ (for some $\boldsymbol{\mu} \in \mathbb{R}^p$). In addition, we assume that the problem considered satisfies Cottle constraint qualification (Def. 2.6) and that the feasible region $S \subset \mathbb{R}^n$ is nonempty and has an interior.

ALGORITHM 3.1. *(Limited memory interior point bundle method.)*

Data: Choose the final accuracy tolerances $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, the positive line search parameters $\varepsilon_L \in (0, 1/2)$ and $\varepsilon_R \in (\varepsilon_L, 1/2)$, and the distance measure parameter $\gamma \geq 0$ (with $\gamma = 0$ if all the functions involved are convex). Select the lower and the upper bounds $t_{min} \in (0, 1)$ and $t_{max} > 1$ for serious steps. Select the control parameters $C > 0$ for the length of the direction vector and $\varrho > 0$ and $\nu \in (0, 1)$ for the deflection bound. Select the parameter $\epsilon > 0$, the limit $g_{max} < 0$, and the auxiliary lower and upper bounds $\mu_{min} > 0$ and $\mu_{max} > 0$ for dual variables. Select an upper limit $\hat{m}_u \geq 3$ for the number of stored correction pairs.

Step 0: (*Initialization.*) Choose a strictly feasible starting point $\boldsymbol{x}_1 \in \operatorname{int} S$ and a positive initial vector $\boldsymbol{\mu}_1 \in \mathbb{R}^p$ (e.g., $\mu_{i,1} = \min\{-1/g_i(\boldsymbol{x}_1), \mu_{max}\}$). Choose an initial maximum number of stored correction pairs $\hat{m}_c$ ($3 \leq \hat{m}_c \leq \hat{m}_u$) and initialize the limited memory matrices $S_1 = U_1 = [\,]$ (empty matrices) and the scaling parameter $\vartheta_1 = 1$ (see Appendix). Set $\boldsymbol{y}_1 = \boldsymbol{x}_1$, $\boldsymbol{\mu}_1^\alpha = \boldsymbol{\mu}_1$, and $\beta_{L,1} = 0$. Compute $f_1 = f(\boldsymbol{x}_1)$, $\boldsymbol{\xi}_{f,1} \in \partial f(\boldsymbol{x}_1)$, $\boldsymbol{g}_1 = \boldsymbol{g}(\boldsymbol{x}_1)$, $\boldsymbol{\xi}_{\boldsymbol{g},1} \in \partial \boldsymbol{g}(\boldsymbol{x}_1)$ (i.e., a generalized Jacobian [6]), and $\boldsymbol{\xi}_{L,1} \in \partial L(\boldsymbol{x}_1, \boldsymbol{\mu}_1)$. Set the iteration counter $k = 1$.

Step 1: (*Serious step initialization.*) Set the aggregate subgradients $\tilde{\boldsymbol{\xi}}_{f,k} = \boldsymbol{\xi}_{f,k}$, $\tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k} = \boldsymbol{\xi}_{\boldsymbol{g},k}$, and $\tilde{\boldsymbol{\xi}}_{L,k} = \boldsymbol{\xi}_{L,k}$, and the aggregate subgradient locality measure $\tilde{\beta}_{L,k} = 0$. Set an index for the serious step $m = k$.

Step 2: (*Stopping criterion.*) Calculate w_k^1 and w_k^2 by

$$w_k^1 = \tilde{\boldsymbol{\xi}}_{L,k}^T D_k \tilde{\boldsymbol{\xi}}_{L,k} + 2\tilde{\beta}_{L,k} \qquad \text{and} \qquad (16)$$
$$w_k^2 = -\sum_{i \in \mathcal{P}} \mu_{i,k} g_{i,k},$$

respectively. Use the limited memory BFGS update for calculation of D_k if $m = k$ and the limited memory SR1 update, otherwise (see Appendix). If $w_k^1 \leq \varepsilon_1$ and $w_k^2 \leq \varepsilon_2$, then stop with $\boldsymbol{x}_k$ as the final solution. Otherwise, if $w_k^1 \leq 10^3\varepsilon_1$ and $\hat{m}_c < \hat{m}_u$, set $\hat{m}_c = \hat{m}_c + 1$.

Step 3: (*Direction finding.*)

(i) (*Descent direction.*) Solve the values $\boldsymbol{d}_k^\alpha \in \mathbb{R}^n$ and $\boldsymbol{\mu}_{k+1}^\alpha \in \mathbb{R}^p$ satisfying the linear equations (11) and (12) using the same updating formula for D_k (i.e., for B_k^{-1}) as in step 2.
If

$$\|\boldsymbol{d}_k^\alpha\| \leq \varepsilon_1,$$

then stop with $\boldsymbol{x}_k$ as the final solution.

(ii) (*Feasible direction.*) Solve the values $\boldsymbol{d}_k^\beta \in \mathbb{R}^n$ and $\boldsymbol{\mu}_{k+1}^\beta \in \mathbb{R}^p$ satisfying the linear equations (13) and (14) (use the same updating formula as before). If $\tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k^\beta > 0$, set

$$\rho_k = \min\left\{ \varrho\|\boldsymbol{d}_k^\alpha\|^2, \frac{(\nu - 1)\tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k^\alpha}{\tilde{\boldsymbol{\xi}}_{f,k}^T \boldsymbol{d}_k^\beta} \right\}. \qquad (17)$$

Otherwise, set

$$\rho_k = \varrho\|\boldsymbol{d}_k^\alpha\|^2. \qquad (18)$$

(iii) (*Feasible descent direction.*) Compute the search direction

$$\boldsymbol{d}_k = \boldsymbol{d}_k^\alpha + \rho_k \boldsymbol{d}_k^\beta.$$

Step 4: (*Line search and solution updating.*) Set the scaling parameter for the length of the direction vector and for line search $\theta_k = \min\{1, C/\|\boldsymbol{d}_k\|\}$. Choose the initial step size $t_I^k \in [t_{min}, t_{max})$. Determine the step sizes $t_R^k \in (0, t_I^k]$ and $t_L^k \in$

$[0, t_R^k]$ by the constrained line search Algorithm 3.2. Set the corresponding values

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + t_L^k \theta_k \boldsymbol{d}_k, & f_{k+1} &= f(\boldsymbol{x}_{k+1}), & \boldsymbol{g}_{k+1} &= \boldsymbol{g}(\boldsymbol{x}_{k+1}), \\ \boldsymbol{y}_{k+1} &= \boldsymbol{x}_k + t_R^k \theta_k \boldsymbol{d}_k, & \boldsymbol{\xi}_{f,k+1} &\in \partial f(\boldsymbol{y}_{k+1}), & \boldsymbol{\xi}_{\boldsymbol{g},k+1} &\in \partial \boldsymbol{g}(\boldsymbol{y}_{k+1}), \end{aligned}$$

and

$$\hat{\boldsymbol{\xi}}_{L,k+1} \in \partial L(\boldsymbol{y}_{k+1}, \boldsymbol{\mu}_k).$$

Set $\boldsymbol{u}_k = \hat{\boldsymbol{\xi}}_{L,k+1} - \boldsymbol{\xi}_{L,m}$ and $\boldsymbol{s}_k = \boldsymbol{y}_{k+1} - \boldsymbol{x}_k = t_R^k \theta_k \boldsymbol{d}_k$ and update the limited memory matrices U_{k+1} and S_{k+1} (see Appendix).

If condition (3) is valid (i.e., we take a serious step), then set

$$\begin{aligned} &\mu_{i,k+1} = \max\{\mu_{i,k+1}^{\alpha}, \epsilon \|\boldsymbol{d}_k^{\alpha}\|^2\} \qquad \text{for all } i \in \mathcal{P} \text{ and} \\ &\boldsymbol{\xi}_{L,k+1} \in \partial L(\boldsymbol{y}_{k+1}, \boldsymbol{\mu}_{k+1}) \end{aligned}$$

and, if $g_i(\boldsymbol{y}_{k+1}) \geq g_{max}$ and $\mu_{i,k+1} < \mu_{min}$ for some $i \in \mathcal{P}$, then set $\mu_{i,k+1} = \mu_{min}$. Set $\beta_{L,k+1} = 0$, $k = k+1$, and go to Step 1. Otherwise (i.e., condition (4) is valid), set $\boldsymbol{\xi}_{L,k+1} = \hat{\boldsymbol{\xi}}_{L,k+1}$, calculate the locality measure $\beta_{L,k+1}$ by (5). and set $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$.

Step 5: (*Aggregation.*) Determine multipliers λ_i^k satisfying $\lambda_i^k \geq 0$ for all $i \in \{1,2,3\}$, and $\sum_{i=1}^{3} \lambda_i^k = 1$ that minimize the function

$$\begin{aligned} \varphi(\lambda_1, \lambda_2, \lambda_3) = (\lambda_1 \boldsymbol{\xi}_{L,m} + \lambda_2 \boldsymbol{\xi}_{L,k+1} + \lambda_3 \tilde{\boldsymbol{\xi}}_{L,k})^T D_k (\lambda_1 \boldsymbol{\xi}_{L,m} + \lambda_2 \boldsymbol{\xi}_{L,k+1} + \lambda_3 \tilde{\boldsymbol{\xi}}_{L,k}) \\ + 2(\lambda_2 \beta_{L,k+1} + \lambda_3 \tilde{\beta}_{L,k}), \qquad (19) \end{aligned}$$

where, as before, D_k is calculated by the same updating formula as in Step 2. Set

$$\begin{aligned} \tilde{\boldsymbol{\xi}}_{f,k+1} &= \lambda_1^k \boldsymbol{\xi}_{f,m} + \lambda_2^k \boldsymbol{\xi}_{f,k+1} + \lambda_3^k \tilde{\boldsymbol{\xi}}_{f,k}, \\ \tilde{\boldsymbol{\xi}}_{g_i,k+1} &= \lambda_1^k \boldsymbol{\xi}_{g_i,m} + \lambda_2^k \boldsymbol{\xi}_{g_i,k+1} + \lambda_3^k \tilde{\boldsymbol{\xi}}_{g_i,k} \qquad \text{for all } i \in \mathcal{P}, \qquad (20) \\ \tilde{\boldsymbol{\xi}}_{L,k+1} &= \lambda_1^k \boldsymbol{\xi}_{L,m} + \lambda_2^k \boldsymbol{\xi}_{L,k+1} + \lambda_3^k \tilde{\boldsymbol{\xi}}_{L,k}, \end{aligned}$$

and

$$\tilde{\beta}_{L,k+1} = \lambda_2^k \beta_{L,k+1} + \lambda_3^k \tilde{\beta}_{L,k}. \qquad (21)$$

Set $k = k+1$ and go to Step 2.

Assumptions to matrices. To ensure the convergence of aggregate subgradients to zero, we assume that matrices D_k are uniformly positive definite and uniformly bounded (we say that a matrix is bounded if its eigenvalues lie in the compact interval that does not contain zero). This requires some modifications to the model algorithm, for instance, corrections of matrices D_k when necessary. In this way we obtain more complicated algorithm which, in unconstrained case, is described in detail in [11, 14]. The basic assumption for bundle method to converge, that is, after a null step we have $\boldsymbol{z}^T D_{k+1} \boldsymbol{z} \leq \boldsymbol{z}^T D_k \boldsymbol{z}$ for all $\boldsymbol{z} \in \mathbb{R}^n$, is guaranteed by the special limited memory SR1 update [11, 14]. In addition to these, we assume that there exist positive numbers ω_1 and ω_2 such that $\omega_1 \|\boldsymbol{d}\|^2 \leq \boldsymbol{d}^T B_k \boldsymbol{d} \leq \omega_2 \|\boldsymbol{d}\|^2$ for all $\boldsymbol{d} \in \mathbb{R}^n$ (see [37] for less restrictive conditions for B_k).

Line search procedure. The initial step size $t_I^k \in [t_{min}, t_{max})$ (see Step 4 in Algorithm 3.1) is selected by using a bundle containing auxiliary points and corresponding function values and subgradients. The procedure used is similar to that in the original variable metric bundle method for nonconvex objective functions [38].

We now present the line search algorithm, which is used to determine the step sizes t_L^k and t_R^k in the limited memory interior point bundle method. The line search procedure used is rather similar to that given in [11, 14] which, on the other hand, was derived from [38]. However, in interior point algorithms we are not allowed to violate constraints. Thus, we need to extend the line search procedure used in the previous variants of the limited memory bundle method to the constrained case.

ALGORITHM 3.2. *(Modified line search for inequality constrained problems).*

Data: Suppose that we have the current iteration point $\boldsymbol{x}_k$, the current dual variables $\boldsymbol{\mu}_k$, the current search direction $\boldsymbol{d}_k$, the current scaling parameter $\theta_k \in (0, 1]$, the current vector $\tilde{\boldsymbol{\xi}}_{L,k}^T D_k$, and the positive line search parameters $\varepsilon_L \in (0, 1/2)$, $\varepsilon_R \in (\varepsilon_L, 1/2)$, $\varepsilon_A \in (0, \varepsilon_R - \varepsilon_L)$, and $\varepsilon_T \in (\varepsilon_L, \varepsilon_R - \varepsilon_A)$ available. In addition, suppose that we have given the initial step size t_I^k, an auxiliary lower bound for serious steps $t_{min} \in (0, 1)$, the distance measure parameter $\gamma \geq 0$, the desirable amount of descent w_k^1, the maximum number of additional interpolations i_{max}, and the number of consecutive null steps $i_{null} \geq 0$.

Step 0: (*Initialization.*) Set $t_A = 0$, $t = t_U = t_I^k$, and $i_I = 0$, and calculate the interpolation parameter

$$\kappa = 1 - \frac{1}{2(1 - \varepsilon_T)}.$$

Step 1: (*New values.*) Compute $f(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k)$, $g_i(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k)$ for all $i \in \mathcal{P}$, $\hat{\boldsymbol{\xi}}_L \in \partial L(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k, \boldsymbol{\mu}_k)$, and

$$\beta_L = \max \{ |L(\boldsymbol{x}_k, \boldsymbol{\mu}_k) - L(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k, \boldsymbol{\mu}_k) + t\theta_k \boldsymbol{d}_k^T \hat{\boldsymbol{\xi}}_L|, \gamma \, (t\theta_k \|\boldsymbol{d}_k\|)^2 \}. \quad (22)$$

If

$$g_i(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k) \geq 0 \qquad \text{for some } i \in \mathcal{P},$$

then set $t_U = t$ and go to Step 5. Else, if $f(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) - \varepsilon_T t w_k^1$, then set $t_A = t$. Otherwise, set $t_U = t$.

Step 2: (*Serious step.*) If

$$f(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) - \varepsilon_L t w_k^1,$$

and either

$$t \geq t_{min} \qquad \text{or} \qquad \beta_L > \varepsilon_A w_k^1,$$

then set $t_R^k = t_L^k = t$ and stop.

Step 3: (*Test for additional interpolation.*) If $f(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k) > f(\boldsymbol{x}_k)$, $i_{null} > 0$, and $i_I < i_{max}$, then set $i_I = i_I + 1$ and go to Step 5.

Step 4: (*Null step.*) If

$$-\beta_L - \tilde{\boldsymbol{\xi}}_{L,k}^T D_k \hat{\boldsymbol{\xi}}_L \geq -\varepsilon_R w_k^1,$$

then set $t_R^k = t$, $t_L^k = 0$ and stop.

Step 5: (*Interpolation.*) If $t_A = 0$, then set

$$t = \max\left\{\kappa t_U, \frac{-\frac{1}{2} t_U^2 w_k^1}{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + t\theta_k \boldsymbol{d}_k) - t_U w_k^1}\right\}.$$

Otherwise, set $t = \frac{1}{2}(t_A + t_U)$. Go to Step 1.

It can be proved that Algorithm 3.2 terminates in a finite number of iterations (the proof is rather similar to that given in [38]) if the problem satisfies the following modified semi-smoothness assumption: For all $\boldsymbol{\xi}_L \in \partial L(\boldsymbol{x}, \boldsymbol{\mu}_k)$ and for any $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{d} \in \mathbb{R}^n$, and sequences $(\hat{\boldsymbol{\xi}}_{L,i}) \subset \mathbb{R}^n$ and $(t_i) \subset \mathbb{R}_+$ satisfying $\hat{\boldsymbol{\xi}}_{L,i} \in \partial L(\boldsymbol{x} + t_i \boldsymbol{d}, \boldsymbol{\mu}_k)$ and $t_i \downarrow 0$, we have

$$-\limsup_{i\to\infty} \boldsymbol{\xi}_L^T D \hat{\boldsymbol{\xi}}_{L,i} \geq \liminf_{i\to\infty} \frac{f(\boldsymbol{x} + t_i \boldsymbol{d}) - f(\boldsymbol{x})}{t_i},$$

where D is the inverse variable metric approximation calculated at point $\boldsymbol{x}$. Note that if we have no constraints present (i.e., $p = 0$ in problem (1)), then this modified semi-smoothness assumption reverts very similar to classical semi-smoothness assumption [4].

On the output of Algorithm 3.2 (see Steps 2 and 4), the step sizes t_L^k and t_R^k satisfy the serious descent criterion

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \leq -\varepsilon_L t_L^k w_k^1 \tag{23}$$

and, in the case of $t_L^k = 0$ (a null step), also condition (4). Moreover, in both cases, for all $i \in \mathcal{P}$, we have $g_i(\boldsymbol{y}_{k+1}) < 0$.

4 Convergence Analysis

In this section, we study the convergence properties of Algorithm 3.1. In addition to assuming that all the functions involved are locally Lipschitz continuous, the set $S \cap \{\, \boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_1) \,\}$ is supposed to be compact and the problem is assumed to satisfy Cottle constraint qualification (Def. 2.6). Furthermore, we assume that each execution of the line search procedure is finite (i.e., the modified semismoothness assumption is valid) and that the matrices D_k and B_k satisfy the assumptions given before.

We start the theoretical analysis of limited memory interior point bundle method by noting that the solutions of linear systems (11), (12), and (13), (14) are unique. After that, we study the case when the algorithm terminates after a finite number of iterations: we prove that if Algorithm 3.1 stops at iteration k, then the point $\boldsymbol{x}_k$ is a KKT point for problem (1) (note that if all the functions involved are convex, this is also a global minimum for the problem). Finally, we prove that every accumulation point $\bar{\boldsymbol{x}}$ of the sequence $(\boldsymbol{x}_k) \subset S$ is a KKT point for problem (1) provided the complementary condition (see Theorem 2.7) is satisfied. For these purposes, we assume that the final accuracy tolerances ε_1 and ε_2 are equal to zero.

REMARK 4.1. The sequence $(\boldsymbol{x}_k)$ is bounded by assumption and the monotonicity of the sequence (f_k) obtained due to serious descent criterion (23). Since $\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1}$ for serious steps and $\|\boldsymbol{y}_{k+1} - \boldsymbol{x}_{k+1}\| \leq t_{max}C$ for null steps by (2) and due to the fact that we use the scaled direction vector $\theta_k \boldsymbol{d}_k$ with $\theta_k = \min\{\, 1, C/\|\boldsymbol{d}_k\| \,\}$ and predefined $C > 0$ in the line search, the sequence $(\boldsymbol{y}_k)$ is also bounded. By the local boundedness and the upper semicontinuity of subdifferential we obtain the boundedness of subgradients $\boldsymbol{\xi}_{L,k}$, $\boldsymbol{\xi}_{f,k}$, and $\boldsymbol{\xi}_{g_i,k}$ for all $i \in \mathcal{P}$, as well as all their convex combinations (see [6]).

The fact that the solutions of linear systems (11), (12), and (13), (14) are unique is a consequence of Lemma 3.1 in [34] stated as follows (using the notation of this paper)

LEMMA 4.1. *For any vector $\boldsymbol{x} \in S$, any positive definite matrix $B \in \mathbb{R}^{n \times n}$ and any nonnegative vector $\boldsymbol{\mu} \in \mathbb{R}^p$ such that $\mu_i > 0$ if $g_i(\boldsymbol{x}) = 0$, the matrix*

$$M(\boldsymbol{x}, B, \boldsymbol{\mu}) = \begin{bmatrix} B & \tilde{\boldsymbol{\xi}}_{\boldsymbol{g}} \\ \Lambda \tilde{\boldsymbol{\xi}}_{\boldsymbol{g}}^T & G(\boldsymbol{x}) \end{bmatrix}$$

is nonsingular.

REMARK 4.2. Since the matrix B_k is bounded by assumption, $\boldsymbol{x}_k$ lies in a compact set, and $\boldsymbol{\mu}_k$ is bounded (see [15]), the matrix $M(\boldsymbol{x}_k, B_k, \boldsymbol{\mu}_k)$ is bounded away from zero and, thus, $\boldsymbol{d}_k^\alpha$, $\boldsymbol{\mu}_k^\alpha$, $\boldsymbol{d}_k^\beta$, and $\boldsymbol{\mu}_k^\beta$ are bounded from above.

LEMMA 4.2. *Suppose that Algorithm 3.1 is not terminated before the kth iteration. Then,* *there exist numbers $\lambda^{k,j} \geq 0$ for $j = 1, \ldots, k$ and $\tilde{\alpha}_k \geq 0$ such that*

$$(\tilde{\boldsymbol{\xi}}_{L,k}, \tilde{\alpha}_k) = \sum_{j=1}^{k} \lambda^{k,j}(\boldsymbol{\xi}_{L,j}, \|\boldsymbol{y}_j - \boldsymbol{x}_k\|), \qquad \sum_{j=1}^{k} \lambda^{k,j} = 1, \quad \text{and} \quad \tilde{\beta}_{L,k} \geq \gamma\tilde{\alpha}_k^2.$$

PROOF. See the proof of Lemma 3.2 in [38]. □

LEMMA 4.3. *Let $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ be given and suppose that there exist a function $\psi : \mathbb{R}^n \to \mathbb{R}$, vectors $\bar{\boldsymbol{\zeta}}$, $\bar{\boldsymbol{\xi}}_j$, $\bar{\boldsymbol{y}}_j$, and numbers $\bar{\lambda}_j \geq 0$ for $j = 1, \ldots, l$, $l \geq 1$, such that*

$$\begin{aligned}
&(\bar{\boldsymbol{\zeta}}, 0) = \sum_{j=1}^{l} \bar{\lambda}_j(\bar{\boldsymbol{\xi}}_j, \|\bar{\boldsymbol{y}}_j - \bar{\boldsymbol{x}}\|),\\
&\bar{\boldsymbol{\xi}}_j \in \partial\psi(\bar{\boldsymbol{y}}_j), \quad j = 1, \ldots, l, \qquad \text{and}\\
&\sum_{j=1}^{l} \bar{\lambda}_j = 1.
\end{aligned}$$

Then $\bar{\boldsymbol{\zeta}} \in \partial\psi(\bar{\boldsymbol{x}})$.

PROOF. See the proof of Lemma 3.3 in [38]. □

THEOREM 4.4. *If Algorithm 3.1 terminates at the kth iteration, then the point $\boldsymbol{x}_k$ is a KKT point for problem (1).*

PROOF. If Algorithm 3.1 terminates at Step 3(i), then the fact $\varepsilon_1 = 0$ implies that $\boldsymbol{d}_k^\alpha$ is zero. Due to (11), (12), and strict feasibility of $\boldsymbol{x}_k$, we have $\tilde{\boldsymbol{\xi}}_{f,k} + \tilde{\boldsymbol{\xi}}_{\boldsymbol{g},k}\boldsymbol{\mu}_{k+1}^\alpha = \mathbf{0}$ with $\boldsymbol{\mu}_{k+1}^\alpha = \mathbf{0}$ and, thus, the point $\boldsymbol{x}_k$ is a KKT point for problem (1).

Let us now assume that Algorithm 3.1 terminates at Step 2. We point out first that $\tilde{\beta}_{L,k} \geq 0$ for all k by (5), (21), and Step 1 in Algorithm 3.1. Due to (16), the positive definiteness of D_k, and the correction term σI with $\sigma \in (0, 1/2)$ that is added to matrix D_k, if necessary (see [11, 14]), we have

$$w_k^1 \geq 2\tilde{\beta}_{L,k} \qquad \text{and} \qquad w_k^1 \geq \sigma\|\tilde{\boldsymbol{\xi}}_{L,k}\|^2. \tag{24}$$

Since Algorithm 3.1 terminates at Step 2, we have $w_k^1 = 0$. Thus, $\tilde{\boldsymbol{\xi}}_{L,k} = \mathbf{0}$ and $\tilde{\beta}_{L,k} = \tilde{\alpha}_k = 0$ by (24) and Lemma 4.2. Now, by Lemma 4.2 and by using Lemma 4.3 with

$$\begin{aligned}
&\bar{\boldsymbol{x}} = \boldsymbol{x}_k, \qquad && l = k, \qquad && \bar{\boldsymbol{\zeta}} = \tilde{\boldsymbol{\xi}}_{L,k},\\
&\bar{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_{L,j}, && \bar{\boldsymbol{y}}_j = \boldsymbol{y}_j, && \bar{\lambda}_j = \lambda^{k,j} \quad \text{for } j \leq k,
\end{aligned}$$

and

$$\partial\psi(\boldsymbol{y}_j) = \partial L(\boldsymbol{y}_j, \boldsymbol{\mu}_j)$$

we obtain $\mathbf{0} = \tilde{\boldsymbol{\xi}}_{L,k} \in \partial L(\boldsymbol{x}_k, \boldsymbol{\mu}_k) \subset \partial\hat{L}(\boldsymbol{x}_k, \boldsymbol{\mu}_k)$.

Moreover, since Algorithm 3.1 stops at Step 2, we have $w_k^2 = 0$. By feasibility of $\boldsymbol{x}_k$ and positiveness of $\boldsymbol{\mu}_k$ it follows that $G(\boldsymbol{x}_k)\boldsymbol{\mu}_k = \mathbf{0}$ and, thus, $\boldsymbol{x}_k$ is a KKT point for problem (1). □

From now on, we suppose that Algorithm 3.1 does not terminate, that is, $\boldsymbol{d}_k^\alpha \neq \mathbf{0}$ and $w_k^1 > 0$ for all k.

LEMMA 4.5. *Suppose that there exist a point $\bar{\boldsymbol{x}} \in S$, a vector $\bar{\boldsymbol{\mu}} \in \mathbb{R}^p$ and an infinite set $\mathcal{K} \subset \{1, 2, \dots\}$ such that $(\boldsymbol{x}_k)_{k\in\mathcal{K}} \to \bar{\boldsymbol{x}}$, $(\boldsymbol{\mu}_k)_{k\in\mathcal{K}} \to \bar{\boldsymbol{\mu}}$, and $(w_k^1)_{k\in\mathcal{K}} \to 0$, then $\mathbf{0} \in \partial L(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\mu}})$. Moreover, if $w_k^2 \to 0$, then $\bar{\boldsymbol{x}}$ is a KKT point for problem (1).*

PROOF. By using the subdifferential of Lagrangian instead of objective function the first part of the proof is similar to the proof of Lemma 3.4 in [38]. Due to feasibility of $\bar{\boldsymbol{x}}$ and positiveness of $\bar{\boldsymbol{\mu}}$ the second part follows directly from the definition of KKT points (see Theorem 2.7). □

LEMMA 4.6. *Suppose that the number of serious steps in Algorithm 3.1 is finite and the last serious step occurred at the iteration $m-1$. Then there exists a number $k^* \geq m$, such that*

$$\tilde{\boldsymbol{\xi}}_{L,k+1}^T D_{k+1} \tilde{\boldsymbol{\xi}}_{L,k+1} \leq \tilde{\boldsymbol{\xi}}_{L,k+1}^T D_k \tilde{\boldsymbol{\xi}}_{L,k+1} \qquad \text{and} \tag{25}$$

$$\operatorname{tr}(D_k) < \frac{3}{2} n \tag{26}$$

for all $k \geq k^*$, where $\operatorname{tr}(D_k)$ denotes the trace of matrix D_k.

PROOF. The result is due to safeguarded SR1 update used (see [11, 14]). See the proof of Lemma 7 in [14]. □

LEMMA 4.7. *Suppose that the number of serious steps is finite and the last serious step occurred at the iteration $m-1$. Then, $\mathbf{0} \in \partial L(\boldsymbol{x}_m, \boldsymbol{\mu}_m)$. Moreover, if $w_k^2 \to 0$, then $\boldsymbol{x}_m$ is a KKT point for problem (1).*

PROOF. From (16), (19), (20), (21), and Lemma 4.6 we obtain

$$\begin{aligned} w_{k+1}^1 &= \tilde{\boldsymbol{\xi}}_{L,k+1}^T D_{k+1} \tilde{\boldsymbol{\xi}}_{L,k+1} + 2\tilde{\beta}_{L,k+1} \\ &\leq \tilde{\boldsymbol{\xi}}_{L,k+1}^T D_k \tilde{\boldsymbol{\xi}}_{L,k+1} + 2\tilde{\beta}_{L,k+1} \\ &\leq \tilde{\boldsymbol{\xi}}_{L,k}^T D_k \tilde{\boldsymbol{\xi}}_{L,k} + 2\tilde{\beta}_{L,k} = w_k^1 \end{aligned} \tag{27}$$

for $k \geq k^*$ with k^* defined in Lemma 4.6. The last inequality in (27) follows from the fact that the pair $(\tilde{\boldsymbol{\xi}}_{L,k+1}, \tilde{\beta}_{L,k+1})$ minimizes function (19) over all convex combinations of pairs $(\boldsymbol{\xi}_{L,m}, \beta_{L,m})$, $(\boldsymbol{\xi}_{L,k+1}, \beta_{L,k+1})$, and $(\tilde{\boldsymbol{\xi}}_{L,k}, \tilde{\beta}_{L,k})$. In addition, the line search procedure guarantees that we have

$$-\beta_{L,k+1} - \tilde{\boldsymbol{\xi}}_{L,k}^T D_k \boldsymbol{\xi}_{L,k+1} \geq -\varepsilon_R w_k^1$$

for all $k \geq m$. Now, due to boundedness of $\boldsymbol{\xi}_{L,k+1}$, $\tilde{\boldsymbol{\xi}}_{L,k}$, and D_k (see Remark 4.1 and Lemma 4.6) it can be proved that $w_k^1 \to 0$ (the proof is rather similar to the proof, part(ii), of Lemma 3.6 in [38]). Now, since $\boldsymbol{x}_k = \boldsymbol{x}_m$ and $\boldsymbol{\mu}_k = \boldsymbol{\mu}_m$ for all $k \geq m$, we have $\boldsymbol{x}_k \to \boldsymbol{x}_m$ and $\boldsymbol{\mu}_k \to \boldsymbol{\mu}_m$. Therefore, by Lemma 4.5 we have $\mathbf{0} \in \partial L(\boldsymbol{x}_m, \boldsymbol{\mu}_m)$ and $\boldsymbol{x}_m$ is a KKT point for problem (1), if $w_k^2 \to 0$. □

THEOREM 4.8. *For any accumulation point $\bar{\boldsymbol{x}}$ of the sequence $(\boldsymbol{x}_k)$ we have $\mathbf{0} \in \partial L(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\mu}})$, where $\bar{\boldsymbol{\mu}}$ is an accumulation point of the sequence $(\boldsymbol{\mu}_k)$. Moreover, if $w_k^2 \to 0$, then $\bar{\boldsymbol{x}}$ is a KKT point for problem (1).*

PROOF. Let $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{\mu}}$ be accumulation points of $(\boldsymbol{x}_k)$ and $(\boldsymbol{\mu}_k)$, respectively. Further, let $\mathcal{K} \subset \{1, 2, \ldots\}$ be an infinite set such that $(\boldsymbol{x}_k)_{k\in\mathcal{K}} \to \bar{\boldsymbol{x}}$ and $(\boldsymbol{\mu}_k)_{k\in\mathcal{K}} \to \bar{\boldsymbol{\mu}}$. Note that due to boundedness of $(\boldsymbol{\mu}_k)$ (see [15]) such a set exists. In view of Lemma 4.7, we can restrict our consideration to the case where the number of serious steps (with $t_L^k > 0$) is infinite. Let us denote

$$\mathcal{K}' = \{k \mid t_L^k > 0, \text{ there exists } i \in \mathcal{K},\ i \leq k \text{ such that } \boldsymbol{x}_i = \boldsymbol{x}_k \text{ and } \boldsymbol{\mu}_i = \boldsymbol{\mu}_k\}.$$

Obviously, $\mathcal{K}'$ is infinite, $(\boldsymbol{x}_k)_{k\in\mathcal{K}'} \to \bar{\boldsymbol{x}}$, and $(\boldsymbol{\mu}_k)_{k\in\mathcal{K}'} \to \bar{\boldsymbol{\mu}}$. The continuity of f implies that $(f_k)_{k\in\mathcal{K}'} \to f(\bar{\boldsymbol{x}})$ and, thus, $f_k \downarrow f(\bar{\boldsymbol{x}})$ by the monotonicity of the sequence (f_k) obtained due to serious descent criterion (23). Using the fact that $t_L^k \geq 0$ for all $k \geq 1$ and condition (23), we obtain

$$0 \leq \varepsilon_L t_L^k w_k^1 \leq f_k - f_{k+1} \to 0 \qquad \text{for } k \geq 1. \tag{28}$$

If the set $\mathcal{K}_1 = \{k \in \mathcal{K}' \mid t_L^k \geq t_{min}\}$ is infinite, then $(w_k^1)_{k\in\mathcal{K}_1} \to 0$ by (28) and $(\boldsymbol{x}_k)_{k\in\mathcal{K}_1} \to \bar{\boldsymbol{x}}$ and $(\boldsymbol{\mu}_k)_{k\in\mathcal{K}_1} \to \bar{\boldsymbol{\mu}}$. Thus, by Lemma 4.5 we have $\mathbf{0} \in \partial L(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\mu}})$ and $\bar{\boldsymbol{x}}$ is a KKT point for problem (1) if $w_k^2 \to 0$.

If the set $\mathcal{K}_1$ is finite, then the set $\mathcal{K}_2 = \{k \in \mathcal{K}' \mid \beta_{L,k+1} > \varepsilon_A w_k^1\}$ has to be infinite (see Algorithm 3.2, Step 2). To the contrary, let us assume that

$$w_k^1 \geq \delta > 0, \qquad \text{for all } k \in \mathcal{K}_2.$$

From (28), we have $(t_L^k)_{k\in\mathcal{K}_2} \to 0$ and Step 4 in Algorithm 3.1 implies

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| = t_L^k \theta_k \|\boldsymbol{d}_k\| \leq t_L^k C$$

for all $k \geq 1$. Thus, we have $(\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|)_{k\in\mathcal{K}_2} \to 0$. By (22), (28), the boundedness of $\boldsymbol{\xi}_{L,k}$ and $\boldsymbol{\mu}_k^\alpha$ (see Remarks 4.1 and 4.2), and the fact that all the constraint functions are locally Lipschitz continuous (i.e., we have $g_i(\boldsymbol{x}_k) - g_i(\boldsymbol{x}_{k+1}) \leq L\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|$ for all $i \in \mathcal{P}$ with some Lipschitz constant $L > 0$), we obtain $(\beta_{L,k+1})_{k\in\mathcal{K}_2} \to 0$, which is in contradiction with

$$\varepsilon_A \delta \leq \varepsilon_A w_k^1 < \beta_{L,k+1}, \qquad k \in \mathcal{K}_2.$$

Therefore, there exists an infinite set $\mathcal{K}_3 \subset \mathcal{K}_2$ such that $(w_k^1)_{k\in\mathcal{K}_3} \to 0$, $(\boldsymbol{x}_k)_{k\in\mathcal{K}_3} \to \bar{\boldsymbol{x}}$, and $(\boldsymbol{\mu}_k)_{k\in\mathcal{K}_3} \to \bar{\boldsymbol{\mu}}$. By Lemma 4.5, we have $\mathbf{0} \in \partial L(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\mu}})$ and $\bar{\boldsymbol{x}}$ is a KKT point for problem (1), if $w_k^2 \to 0$. $\square$

5 Numerical Experiments

In this section we compare the limited memory interior point bundle method `LMBM-IP` to the proximal bundle method `PBNCGC` with improvement function [31]. We used the solver PBNCGC as a benchmark since the proximal bundle method is the most frequently used bundle method in nonsmooth optimization. In addition, we compare LMBM-IP to the bound constrained version of the method LMBM-B [12]. The experiments were performed in a Intel® Pentium® 4 CPU 3.20GHz and all the algorithms were implemented in Fortran77 with double-precision arithmetic.

The number of variables used in all the experiments was 1000, and the solvers were tested with relatively small sizes of the bundles (m_ξ), that is, $m_\xi = 2$ for LMBM-IP and LMBM-B and $m_\xi = 100$ for PBNCGC (since the previous experiments [11, 14] have shown that a larger bundle usually works better with PBNCGC). For the limited memory bundle solvers LMBM-IP and LMBM-B the upper limit for the stored correction pairs ($\hat{m}_u$) was set to 15 and the initial maximum number of stored corrections pairs ($\hat{m}_c$) was set to 7. The (first) final accuracy tolerance $\varepsilon_1 = 10^{-5}$ was used in all the cases. In addition to the usual stopping criteria of the solvers, we terminated the experiments if the CPU time elapsed exceeded half an hour.

Through the numerical experiments the following parameters were used with the solver LMBM-IP:

$$\varrho = 10^{-9}, \quad \nu = 0.99, \quad \epsilon = 10^{-12}, \quad g_{max} = 0.001, \quad \text{and} \quad \varepsilon_2 = 10^{-4}.$$

In addition, the parameters μ_{min} and μ_{max} were selected from the values $\mu_{max} = 2.0$, 10.0, and 1000.0 and $\mu_{min} = 0.001$, and 0.1. With all the solvers the distance measure parameter value $\gamma = 0$ was used with convex problems and the value $\gamma = 0.5$ with nonconvex problems. Otherwise, the default values of the parameters were used.

Bound constrained problems. The solvers were first tested with 10 nonsmooth academic minimization problems described in [13]. Half of these problems (problems 1–5) were convex and the other half (problems 6–10) were nonconvex. The problems in [13] are unconstrained but we inclosed the additional bounds

$$x_i^* + 0.1 \le x_i \le x_i^* + 1.1 \quad \text{for all odd } i \text{ such that } i \le 100,$$

where $\boldsymbol{x}^*$ denotes the solution for the unconstrained problem. If the original starting point given in [13] was not feasible, we simply projected it to the feasible region and, since the starting point for LMBM-IP have to be strictly feasible, added an additional safeguard of 0.0001.

The results of the bound constrained experiments are given in Table 1, where Ni and Nf denote the numbers of iterations and function evaluations used, respectively, f denotes the value of the objective function at termination, and the time is an average CPU time elapsed per problem and it is given in seconds (only the accurately and successfully terminated problems were included).

To sum up, the new solver LMBM-IP do not beat up LMBM-B that has specially been designed for solving large-scale nonsmooth bound constrained problems. However, even if there exists a special way of dealing with bound constraints in PBNCGC, the solver LMBM-IP was on the average 36 times faster than it.

Table 1: Results for bound constrained problems.

Solver	LMBM-IP		LMBM-B		PBNCGC	
Problem	Ni/Nf	f	Ni/Nf	f	Ni/Nf	f
1	-/-	fail	17364/17652	0.01000	8647/8671	0.01000
2	70/791	0.05148	259/408	0.05187	19/26	0.00006
3	153/1956	-1411.05	37/70	-1409.80	5387/5580	-1411.09
4	114/1391	2031.75	143/690	2031.86	6371/6372	2031.72
5	66/929	2000.15	49/121	2000.18	29/31	2000.15
6	54/552	0.09531	513/514	0.09531	74/2496	0.11502
7	108/1563	10.0809	499/3369	10.0550	302/303	10.0000
8	259/2505	-705.456	114/270	-704.206	5026/5027	-705.671
9	82/856	0.52112	101/290	0.52122	71/140	0.52113
10	130/1227	14.6443	155/570	14.5630	227/232	14.5594
Time	17.07		4.41		619.44	

Inequality constrained problems. In addition to bound constraint problems we tested the solvers LMBM-IP and PBNCGC with 50 inequality constrained problems. The problems were constructed by combining the problems in [13] to the constraints given in Appendix. The constraints were selected such that the original unconstrained minima of problems are not feasible. Note that, due to nonconvexity of the constraints, all the inequality constrained problems used in our experiments were nonconvex. This means that there may exist more than one local minimum for these problems.

The results of the inequality constrained experiments are given in Table 2. The number of the problem in Table 2 is constructed by first taking the number of the problem in [13] and then the number of the constraint in Appendix. Note that due to large dimension, nonconvexity, nonsmoothness, and nonlinear/nonsmooth inequality constraints these problems are very difficult to solve.

In Table 2 we used the term "fail" in three different failure terminations: first, if there was some numerical difficulties that prevent solvers from working ("fail-1", in problem 7-2); second, if, by practically speaking, no optimization occurred ("fail-2", in almost all problems with constraints 3 and 4 with PBNCGC and in problems 1-3 and 1-5 with LMBM-IP): and, third, if the solution found was not even near the optimal solution ("fail-3", in the rest of the failed problems in Table 2). In this last case, the most common reason for termination with PBNCGC was that the time was up and with LMBM-IP it was that the value of the objective function is not changed enough in last ten iterations.

At the end of Table 2 we have calculated the average numbers of iterations and function calls as well as the average CPU time elapsed per problem. In each cases, we omitted the problem that used most function evaluations since, for example, with LMBM-IP problem 9-3 used 91% of all function evaluations required.

In Table 2 we can see the superiority of LMBM-IP when comparing the computational times; the computational time elapsed with LMBM-IP was an average about 40 times shorter than that of PBNCGC. It also succeed to solve many more problems (80%) than PBNCGC (50%). However, there were some inaccuracy results obtained

Table 2: Results for inequality constrained problems.

Solver	`LMBM-IP`			`PBNCGC`		
Problem	Ni/Nf	f	Time	Ni/Nf	f	Time
1-1	4914/10791	0.500065	14.25	-/-	fail-3	-
2-1	110/829	0.043072	12.49	2807/9012	0.000163	202.60
3-1	244/2721	-1408.63	1.51	7673/22478	-1408.35	1800.21
4-1	745/9015	2003.24	6.11	-/-	fail-3	-
5-1	230/1853	1998.36	1.35	1882/4199	1998.80	137.61
6-1	137/1060	0.534851	0.61	68/123	0.535092	0.73
7-1	331/3728	5.00248	7.78	-/-	fail-3	-
8-1	254/2552	-680.628	1.41	-/-	fail-3	-
9-1	247/2992	1.56604	1.34	-/-	fail-3	-
10-1	490/6095	5.99059	3.15	7200/20151	6.05138	1800.15
1-2	10389/43599	0.880569	22.27	-/-	fail-3	-
2-2	152/841	0.221364	12.66	4295/14597	0.008487	319.57
3-2	35/306	-735.874	0.12	167/517	-735.874	7.20
4-2	89/609	2808.45	0.46	462/815	2808.45	30.29
5-2	26/103	2800.19	0.07	334/700	2796.35	20.29
6-2	213/1507	2.79690	0.66	591/4036	2.77674	13.76
7-2	-/-	fail-1	-	-/-	fail-1	-
8-2	23/81	4466.99	0.06	82/124	4507.02	2.41
9-2	450/3062	483.441	1.24	122/480	497.297	3.09
10-2	-/-	fail-3	-	-/-	fail-3	-
1-3	-/-	fail-2	-	-/-	fail-2	-
2-3	42/318	0.303804	4.77	91/517	0.007981	8.85
3-3	85/769	-1412.14	0.29	-/-	fail-2	-
4-3	61/653	2001.63	0.33	-/-	fail-2	-
5-3	-/-	fail-3	-	-/-	fail-2	-
6-3	30/88	0.405473	0.04	26/41	3.04435	0.06
7-3	-/-	fail-3	-	-/-	fail-2	-
8-3	169/1662	-705.910	0.61	-/-	fail-2	-
9-3	82304/2645587	0.250063	655.14	-/-	fail-2	-
10-3	91/2157	1.85396	0.56	-/-	fail-2	-
1-4	9943/164502	0.388891	57.37	-/-	fail-2	-
2-4	51/963	1.02087	14.48	91/517	0.007981	8.74
3-4	74/598	-1412.13	0.29	-/-	fail-2	-
4-4	59/689	2001.72	0.41	-/-	fail-2	-
5-4	-/-	fail-3	-	-/-	fail-2	-
6-4	28/63	0.405549	0.04	26/41	3.04435	0.06
7-4	-/-	fail-3	-	-/-	fail-2	-
8-4	194/1569	-705.926	0.73	-/-	fail-2	-
9-4	203/3168	0.250222	1.12	-/-	fail-2	-
10-4	208/2713	1.39342	1.28	-/-	fail-2	-
1-5	-/-	fail-2	-	12702/12703	0.138009	1800.03
2-5	50/599	0.622148	8.97	200/478	0.600611	9.42
3-5	13/103	-1153.55	0.04	172/466	-1153.55	23.67
4-5	11/74	4043.82	0.04	352/913	4043.82	30.45
5-5	11/74	4043.82	0.04	352/947	4043.82	34.22
6-5	28/302	5.81129	0.17	19/36	5.81129	0.04
7-5	9/46	589.475	0.09	204/521	589.469	29.60
8-5	34/471	-660.304	0.17	33/254	-660.307	0.21
9-5	-/-	fail-3	-	2690/8376	490.173	166.60
10-5	-/-	fail-3	-	-/-	fail-3	-
Average	781/7008		4.60	1457/3357		193.74

with the new solver. Particularly, it failed to solve problem 2 with the desired accuracy regardless of the constraint used. These kind of difficulties were quite predictable, since problem 2 is reported to be difficult to solve with limited memory bundle method even without constraints [13].

In both bound constrained and inequality constrained problems the numbers of function evaluations used with `LMBM-IP` were much larger than the numbers of iterations. For instance, in inequality constrained problems `LMBM-IP` used about nine function evaluations per iteration while with `PBNCGC` this ratio was only about two (see Table 2). This also means that `LMBM-IP` usually needed more function evaluations than `PBNCGC` even if the number of iterations was smaller. The reason for large amount of function evaluations needed with `LMBM-IP` is that near the solution the feasible search direction $\boldsymbol{d}_k$ supports very short feasible segments and many function evaluations are needed in line search before a feasible step size is obtained (see Algorithm 3.2). The possibilities of avoiding this kind of effect by using the curvilinear search inside the feasible region (see, e.g., [17, 34]) are to be studied.

6 Conclusions

In this paper, we have described a new interior point based limited memory bundle method for inequality constrained nonsmooth optimization. The new solver is suitable for general nonlinearly constrained large-scale problems, where neither the objective function nor the constraints are supposed to be continuously differentiable or convex.

We have studied the convergence properties of the method and given some results from numerical experiments. The preliminary numerical experiments confirm that the limited memory interior point bundle solver is efficient for both convex and nonconvex large-scale nonsmooth inequality constrained optimization problems. With large numbers of variables it used significantly less CPU time than the other solver tested.

Acknowledgements

The authors would like to thank Ph. Lic. Elina Madetoja (University of Kuopio, Finland) for valuable comments and ideas.

Appendix

Limited memory matrices. The limited memory variable metric matrices used in our algorithm are represented in the compact matrix form originally described in [5].

Let us denote by $\hat{m}_c$ the user-specified maximum number of stored correction pairs ($3 \leq \hat{m}_c$) and by $\hat{m}_k = \min\{\, k-1, \hat{m}_c \,\}$ the current number of stored correction pairs. Then the $n \times \hat{m}_k$ dimensional correction matrices S_k and U_k are defined by

$$\begin{aligned} S_k &= \begin{bmatrix} \boldsymbol{s}_{k-\hat{m}_k} & \dots & \boldsymbol{s}_{k-1} \end{bmatrix} \qquad \text{and} \\ U_k &= \begin{bmatrix} \boldsymbol{u}_{k-\hat{m}_k} & \dots & \boldsymbol{u}_{k-1} \end{bmatrix}, \end{aligned}$$

where the correction pairs $(\boldsymbol{s}_i, \boldsymbol{u}_i)$, $(i < k)$ are obtained in Step 4 of Algorithm 3.1.

The inverse limited memory BFGS update is defined by the formula

$$D_k = \vartheta_k I + \begin{bmatrix} S_k & \vartheta_k U_k \end{bmatrix} \begin{bmatrix} (R_k^{-1})^T (C_k + \vartheta_k U_k^T U_k) R_k^{-1} & -(R_k^{-1})^T \\ -R_k^{-1} & 0 \end{bmatrix} \begin{bmatrix} S_k^T \\ \vartheta_k U_k^T \end{bmatrix}, \tag{29}$$

where R_k is an upper triangular matrix of order $\hat{m}_k$ given by the form

$$(R_k)_{ij} = \begin{cases} (\boldsymbol{s}_{k-\hat{m}_k-1+i})^T (\boldsymbol{u}_{k-\hat{m}_k-1+j}), & \text{if } i \le j \\ 0, & \text{otherwise,} \end{cases}$$

C_k is a diagonal matrix of order $\hat{m}_k$ such that

$$C_k = \operatorname{diag}\,[\boldsymbol{s}_{k-\hat{m}_k}^T \boldsymbol{u}_{k-\hat{m}_k}, \dots, \boldsymbol{s}_{k-1}^T \boldsymbol{u}_{k-1}],$$

and ϑ_k is a positive scaling parameter.

In addition, the inverse limited memory SR1 update is defined by

$$D_k = \vartheta_k I - (\vartheta_k U_k - S_k)(\vartheta_k U_k^T U_k - R_k - R_k^T + C_k)^{-1} (\vartheta_k U_k - S_k)^T. \tag{30}$$

The similar representations for the direct limited memory BFGS and SR1 updates can be given (see, e.g., [5]). However, the implementation of our algorithm only needs the inverse update formulae to be used.

LEMMA 6.1. *The condition*

$$-\boldsymbol{d}_j^T \boldsymbol{u}_j - \left(\tilde{\boldsymbol{\xi}}_{f,j} + \sum_{i \in \mathcal{P}} \bar{\mu}_{i,j+1} \tilde{\boldsymbol{\xi}}_{g_i,j} \right)^T \boldsymbol{s}_j < 0 \qquad \textit{for all } j = 1, \dots, k-1, \tag{31}$$

where $\bar{\boldsymbol{\mu}}_{j+1} = \boldsymbol{\mu}_{j+1}^{\alpha} + \rho_j \boldsymbol{\mu}_{j+1}^{\beta}$, *assures the positive definiteness of the matrices obtained by the limited memory SR1 update. Furthermore, it implies* $\boldsymbol{u}_j^T \boldsymbol{s}_j > 0$ *for all* $j = 1, \dots, k-1$ *that assures the positive definiteness of the matrices obtained by the limited memory BFGS update.*

PROOF. Using condition (31), the fact that we have $\boldsymbol{s}_j = t_R^j \theta_j \boldsymbol{d}_j$, $t_R^j > 0$, $\theta_j \in (0, 1]$, and $\tilde{\boldsymbol{\xi}}_{f,j} + \sum_{i \in \mathcal{P}} \bar{\mu}_{i,j+1} \tilde{\boldsymbol{\xi}}_{g_i,j} = -B_j \boldsymbol{d}_j$ for all $j = 1, \dots, k-1$, we obtain

$$\boldsymbol{d}_j^T \boldsymbol{u}_j > -t_R^j \theta_j \boldsymbol{d}_j^T \left(\tilde{\boldsymbol{\xi}}_{f,j} + \sum_{i \in \mathcal{P}} \bar{\mu}_{i,j+1} \tilde{\boldsymbol{\xi}}_{g_i,j} \right) = t_R^j \theta_j \boldsymbol{d}_j^T B_j \boldsymbol{d}_j^T \ge t_R^j \theta_j \boldsymbol{d}_j^T B_j' \boldsymbol{d}_j \tag{32}$$

for all $i = 1, \dots, k-1$. Now, by replacing equation (29) in [14] by (32), the first part of the proof proceeds similar to the proof of Lemma 10 in [14].

The second part of the lemma follows from (32) provided by the positive definiteness of (previous matrices) B_j and the fact that we have $\boldsymbol{s}_j = t_R^j \theta_j \boldsymbol{d}_j$. $\square$

In our proposal, the individual updates that would violate positive definiteness are skipped (for more details, see [11, 13, 14]).

Constraints. The following constraint functions were used in our inequality constrained experiments:

1. Modification of Broyden tridiagonal constraint

$$g_i(\boldsymbol{x}) = (3.0 - 2.0x_{i+1})x_{i+1} - x_i - 2.0x_{i+2} + 1.0, \qquad i = 1, \dots, 5,$$

for problems 1, 2, 6, 7, 9, and 10 in [13] and

$$g_i(\boldsymbol{x}) = (3.0 - 2.0x_{i+1})x_{i+1} - x_i - 2.0x_{i+2} + 2.5, \qquad i = 1, \dots, 5,$$

for problems 3, 4, 5, and 8 in [13] (for original Broyden tridiagonal constraint, see, e.g., [28]).

2. Modification of Broyden tridiagonal constraint II

$$g_1(\boldsymbol{x}) = \sum_{i=1}^{n-2} \left((3.0 - 2.0x_{i+1})x_{i+1} - x_i - 2.0x_{i+2} + 1.0\right),$$

for problems 1, 2, 6, 7, 9, and 10 in [13] and

$$g_1(\boldsymbol{x}) = \sum_{i=1}^{n-2} \left((3.0 - 2.0x_{i+1})x_{i+1} - x_i - 2.0x_{i+2} + 2.5\right),$$

for problems 3, 4, 5, and 8 in [13].

3. Modification of MAD1

$$\begin{aligned} g_1(\boldsymbol{x}) &= \max\left\{x_1^2 + x_2^2 + x_1x_2 - 1.0,\ \sin x_1,\ -\cos x_2\right\}, \\ g_2(\boldsymbol{x}) &= -x_1 - x_2 + 0.5, \end{aligned}$$

(for original problem, see, e.g., [29]).

4. Modification of MAD1 II

$$\begin{aligned} g_1(\boldsymbol{x}) &= x_1^2 + x_2^2 + x_1x_2 - 1.0, \\ g_2(\boldsymbol{x}) &= \sin x_1, \\ g_3(\boldsymbol{x}) &= -\cos x_2, \\ g_4(\boldsymbol{x}) &= -x_1 - x_2 + 0.5. \end{aligned}$$

5. Simple modification of MAD1

$$g_1(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(x_i^2 + x_{i+1}^2 + x_ix_{i+1} - 2.0x_i - 2.0x_{i+1} + 1.0\right)$$

for problems 1, 2, 6, 7, 9, and 10 in [13] and

$$g_1(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(x_i^2 + x_{i+1}^2 + x_i x_{i+1} - 1.0 \right)$$

for problems 3, 4, 5, and 8 in [13].

In all cases the starting point were chosen to be feasible.

References

[1] BAKHTIARI, S., AND TITS, A. A simple primal-dual feasible interior-point method for nonlinear programming with monotone descent. *Computational Optimization and Applications 25* (2003), 17–38.

[2] BELIAKOV, G., MONSALVE TOBON, J. E., AND BAGIROV, A. M. Parallelization of the discrete gradient method of non-smooth optimization and its applications. In *Computational Science — ICCS 2003*, Sloot et. al., Ed., Lecture Notes in Computer Science. Springer Berlin, Heidelberg, 2003, pp. 592–601.

[3] BEN-TAL, A., AND NEMIROVSKI, A. Non-Euclidean restricted memory level method for large-scale convex optimization. *Mathematical Programming 102*, 3 (2005), 407–456.

[4] BIHAIN, A. Optimization of upper semidifferentiable functions. *Journal of Optimization Theory and Applications 4* (1984), 545–568.

[5] BYRD, R. H., NOCEDAL, J., AND SCHNABEL, R. B. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming 63* (1994), 129–156.

[6] CLARKE, F. H. *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, 1983.

[7] FLETCHER, R. *Practical Methods of Optimization*, 2nd ed. John Wiley and Sons, Chichester, 1987.

[8] FLETCHER, R., AND LEYFFER, S. A bundle filter method for nonsmooth nonlinear optimization. University of Dundee, Numerical Analysis Report NA/195, 1999.

[9] GILBERT, J.-C., AND LEMARÉCHAL, C. Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming 45* (1989), 407–435.

[10] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1998.

[11] HAARALA, M. *Large-Scale Nonsmooth Optimization: Variable Metric Bundle Method with Limited Memory*. PhD thesis, University of Jyväskylä, Department of Mathematical Information Technology, 2004.

[12] HAARALA, M., AND MÄKELÄ, M. M. Limited memory bundle algorithm for large bound constrained nonsmooth minimization problems. Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, B 1/2006 University of Jyväskylä, Jyväskylä, 2006.

[13] HAARALA, M., MIETTINEN, K., AND MÄKELÄ, M. M. New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software 19*, 6 (2004), 673–692.

[14] HAARALA, N., MIETTINEN, K., AND MÄKELÄ, M. M. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Mathematical Programming A 109*, 1 (2007), 181–205.

[15] HERSKOVITS, J. Feasible direction interior-point technique for nonlinear optimization. *Journal of Optimization Theory and Applications 99*, 1 (1998), 121–146.

[16] HERSKOVITS, J., AND SANTOS, G. On the computer implementation of feasible direction interior point algorithms for nonlinear optimization. *Structural Optimization 14* (1997), 165–172.

[17] HERSKOVITS, J., AND SANTOS, G. Feasible arc interior point algorithms for nonlinear optimization. In *Computational Mechanics: New Trends and Applications* (1998), S. Idelsohn, E. Oñate, and E. Dvorkin, Eds., CIMNE, Barcelona, Spain.

[18] KARAS, E., RIBEIRO, A., SAGASTIZÁBAL, C., AND SOLODOV, M. A bundle-filter method for nonsmooth convex constrained optimization. Accepted for publication in *Mathematical Programming B*, 2006.

[19] KIWIEL, K. C. An exact penalty function algorithm for nonsmooth convex constrained minimization problems. *IMA Journal of Numerical Analysis 5* (1985), 111–119.

[20] KIWIEL, K. C. *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics 1133. Springer-Verlag, Berlin, 1985.

[21] KIWIEL, K. C. A method of linearizations for linearly constrained nonconvex nonsmooth minimization. *Mathematical Programming 34* (1986), 175–187.

[22] KIWIEL, K. C. A constraint linearization method for nondifferentiable convex minimization. *Numeriche Mathematik 51* (1987), 395–414.

[23] KIWIEL, K. C. Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. *Mathematical Programming 52* (1991), 285–302.

[24] LEMARÉCHAL, C. Nondifferentiable optimization. In *Optimization*, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, Eds. Elsevier North-Holland, Inc., New York, 1989, pp. 529–572.

[25] LEMARÉCHAL, C., NEMIROVSKII, A., AND NESTEROV, Y. New variants of bundle methods. *Mathematical Programming 69* (1995), 111–147.

[26] LEMARÉCHAL, C., STRODIOT, J.-J., AND BIHAIN, A. On a bundle algorithm for nonsmooth optimization. In *Nonlinear Programming*, O. L. Mangasarian, R. R. Mayer, and S. M. Robinson, Eds. Academic Press, New York, 1981, pp. 285–281.

[27] LUKŠAN, L., AND VLČEK, J. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *Journal of Optimization Theory and Applications 102* (1999), 593–613.

[28] LUKŠAN, L., AND VLČEK, J. Sparse and partially separable test problems for unconstrained and equality constrained optimization. Technical Report 767, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 1999.

[29] LUKŠAN, L., AND VLČEK, J. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2000.

[30] MAJAVA, K., HAARALA, N., AND KÄRKKÄINEN, T. Solving variational image denoising problems using limited memory bundle method. In *Proceedings of The 2nd International Conference on Scientific Computing and Partial Differential Equations and The First East Asia SIAM Symposium, Hongkong, December 12-16, 2005.* (to appear, 2006), L. Wenbin, N. Michael, and S. Zhong-Ci, Eds.

[31] MÄKELÄ, M. M., AND NEITTAANMÄKI, P. *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control.* World Scientific Publishing Co., Singapore, 1992.

[32] MIFFLIN, R. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. *Matematical Programming Study 17* (1982), 77–90.

[33] NOCEDAL, J. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation 35*, 151 (1980), 773–782.

[34] PANIER, E. R., TITS, A. L., AND HERSKOVITS, J. N. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Control and Optimization 26*, 4 (1988), 788–811.

[35] SAGASTIZÁBAL, C., AND SOLODOV, M. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization 16*, 1 (2005), 146–169.

[36] Schramm, H., and Zowe, J. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization 2*, 1 (1992), 121–152.

[37] Tits, A. L., Wächter, A., Bakhtiari, S., Urban, T. J., and Lawrence, C. T. A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. *SIAM Journal on Optimization 14*, 1 (2003), 173–199.

[38] Vlček, J., and Lukšan, L. Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. *Journal of Optimization Theory and Applications 111*, 2 (2001), 407–430.