J Optim Theory Appl (2014) 163:900–925
DOI 10.1007/s10957-014-0523-9

Constrained Nonconvex Nonsmooth Optimization via Proximal Bundle Method

Yang Yang · Liping Pang ·
Xuefei Ma · Jie Shen

Received: 26 October 2012 / Accepted: 8 January 2014 / Published online: 23 January 2014

Abstract In this paper, we consider a constrained nonconvex nonsmooth optimization, in which both objective and constraint functions may not be convex or smooth. With the help of the penalty function, we transform the problem into an unconstrained one and design an algorithm in proximal bundle method in which local convexification of the penalty function is utilized to deal with it. We show that, if adding a special constraint qualification, the penalty function can be an exact one, and the sequence generated by our algorithm converges to the KKT points of the problem under a moderate assumption. Finally, some illustrative examples are given to show the good performance of our algorithm.

Keywords Nonconvex optimization · Nonsmooth optimization · Constrained programming · Exact penalty functions · Proximal bundle methods · Lower-C^2

Mathematics Subject Classification (2000) 90C26

Y. Yang · L. Pang (✉)
School of Mathematical Sciences, Dalian University of Technology, Dalian, China
e-mail: lppang@dlut.edu.cn

Y. Yang
e-mail: yangyangdick@163.com

X. Ma
Department of Mathematics, Syracuse University, Syracuse, NY, USA
e-mail: xma15@syr.edu

J. Shen
School of Mathematics, Liaoning Normal University, Dalian, China
e-mail: shenjie73@dl.cn

1 Introduction

The bundle method is recognized as one of the most efficient optimization methods for solving nonsmooth problems, especially for convex cases [1–5]. The linearization error plays an important role in these methods. At a given point, the linearization error is the difference between f and a given linear approximation of f (see [2]). If f is convex, any graph of the linear approximation at a given point is below the graph of f and the linearization error is always non-negative. This fact is crucial in proving the convergence of the bundle methods. But in nonconvex cases, the linearization error can be negative, thus the corresponding linear approximation may not stay below f and even cut off a region that contains a minimizer. In order to force linearization error to be non-negative, several techniques have been proposed. For example, we can take the absolute value of the linearization error [6], or partition the linearization error into non-negative and negative parts [7], or take a maximum of the linearization error and an absolute value term [8–10].

Recently, different variants of proximal bundle methods for solving unconstrained nonconvex minimization are proposed and worked well. Hare and Sagastizábal proposed "redistributed proximal bundle method" in [11,12], where objective function f is convexified in local and prox-parameter is split into two parts. Limited memory bundle method for large-scale problem can be found in [8], and was extended to the problem with bound constraints in [13,14]. Other proximal point methods refer to [6,7,15–18].

To deal with constrained optimization problem, we extend the redistributed proximal bundle method [11] by convexifing not only nonconvex objective function, but also nonconvex constraint function in local. Different from the unconstrained optimization, the penalty approach is used here. The piecewise linear model in this paper no longer approximates the objective function f or its local convexification as in [2,11], but the local convexification of the penalty function at the current serious step.

As the methods in [19], an exact penalty function is used in our method. For convex programming, Kiwiel in [19] supposes that if Slater constraint qualification holds, then penalty method can be exact. In our work, we present an analogous constraint qualification, and prove that under this assumption the exact penalty function can exist for our nonconvex problem. Other approaches solving constrained nonconvex nonsmooth optimization problems refer to [9,20], where penalty function is not used. The "α-function" which unifies the linearization errors of the objective and constraint functions is defined in [9], and the "improvement function" which combines the objective and constraint functions is proposed in [20].

Another feature of our algorithm is that three important parameters, including penalty parameter, are all calculated "on the fly", which are not necessary to be preset. We prove these parameters are all stabilized and our algorithm converges to the KKT points of our problem.

The rest of the paper is organized as follows. In Sect. 2, we review some concepts and give some assumptions about our problem. Our main ideas and implementable algorithm are presented in Sect. 3. Convergence analysis of the algorithm is studied in Sect. 4. In Sect. 5, we list some numerical experiments. We present our conclusion in Sect. 6.

2 Definitions and Assumptions

We consider the following inequality constrained optimization problem

$$\min\ f(x) \qquad \text{s.t. } F_j(x) \le 0,\ \ j = 1, \dots, m, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $F_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, 2, \dots, m$ are nonconvex and nonsmooth functions. Now we recall some concepts and results related to Variational Analysis and give some assumptions on (1).

2.1 Lower-C^2 and Proximal Mapping

In this paper, we use the concept of limiting subdifferential ∂f (see [21]). First, we define the regular subdifferential. Consider a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\bar{x}$ with $f(\bar{x})$ finite, the regular subdifferential of f at $\bar{x}$ is defined by

$$\hat{\partial} f(\bar{x}) := \left\{ g \in \mathbb{R}^n |\ f(x) \ge f(\bar{x}) + \langle g, x - \bar{x} \rangle + o\big(|x - \bar{x}|\big), \forall x \in \mathbb{R}^n \right\},$$

where $|\cdot|$ denotes Euclidean norm. The limiting subdifferential is $\partial f(\bar{x}) := \limsup_{x \xrightarrow{f} \bar{x}} \hat{\partial} f(x)$, where $x \xrightarrow{f} \bar{x}$ means $x \to \bar{x}$ with $f(x) \to f(\bar{x})$. We call elements of this subdifferential subgradients. If the function f is proper and convex at $\bar{x}$, the limiting subdifferential and the regular subdifferential of f are the same, and they have (see [21, Proposition 8.12])

$$\partial f(\bar{x}) = \hat{\partial} f(\bar{x}) := \left\{ g \in \mathbb{R}^n |\ f(x) \ge f(\bar{x}) + \langle g, x - \bar{x} \rangle, \forall x \in \mathbb{R}^n \right\}.$$

Moreover, if f is a convex function and finite at $\bar{x}$, we also can define its ε-subgradient at $\bar{x}$ as

$$\partial_\varepsilon f(\bar{x}) := \left\{ g \in \mathbb{R}^n |\ f(x) \ge f(\bar{x}) + \langle g, x - \bar{x} \rangle - \varepsilon, \forall x \in \mathbb{R}^n \right\}.$$

A function f is called prox-bounded if there exists $r \ge 0$ such that $f + \frac{r}{2}|\cdot|^2$ is bounded below. The corresponding threshold is the smallest $r_{pb} \ge 0$ such that $f + \frac{r}{2}|\cdot|^2$ is bounded below for all $r > r_{pb}$.

In order to extend the proximal bundle method to nonconvex case, we consider lower-C^2 functions.

Definition 2.1 [21, Definition 10.29] (*Lower-C^2*) A function f is said to be lower-C^2 on an open set O if for each $\bar{x} \in O$ there is a neighborhood V of $\bar{x}$ upon which a representation $f(x) = \max_{t \in T} f_t(x)$ holds, where T is a compact set and the functions f_t are twice continuously differentiable jointly in x and t.

The following equivalent definition given in [21, Theorem 10.33] may be more convenient for our discussion. The function f is lower-C^2 on an open set O if f is finite on O, and for any point $x \in O$ there exists a threshold $R > 0$ such that $f + \frac{r}{2}|\cdot|^2$ is convex on an open neighborhood V of x for all $r \geq R$.

There is a wide variety of lower-C^2 functions. For example, C^2 functions must be lower-C^2; any finite convex function is lower-C^2; if f is l.s.c., proper, and prox-bounded, then the opposite of Moreau envelopes $e_\lambda f$ is lower-C^2 (see [21, Excercise 10.32]); a function which is strongly amenable at O is also lower-C^2 (see [21, Excercise 10.36]). We also can see that all lower-C^2 functions are locally Lipschitz continuous functions (see [21, Theorem 10.31]).

Another important concept in proximal bundle method is proximal mapping.

Definition 2.2 [21, Definition 1.22] (*Proximal mapping*) Given a positive parameter R, the proximal mapping of the function f at x is defined by $p_R f(x) := \arg\min_y \{f(y) + \frac{R}{2}|x-y|^2\}$.

We call R and x the prox-parameter and prox-center of proximal mapping, respectively. As in [11], there is a relation of the decomposition which plays a fundamental role in our method.

$$p_R f(\bar{x}) = p_\mu\left(f + \frac{\eta}{2}|\cdot - \bar{x}|^2\right)(\bar{x}), \tag{2}$$

where $\bar{x}$ is a given point, and two non-negative terms μ and η satisfy $R = \mu + \eta$.

2.2 Assumptions

Now we return to our basic problem (1), and give some necessary assumptions.

Assumption 2.1 For the objective function f given in (1), we assume that

(i) f is lower-C^2 on an open bounded set O;
(ii) Given a point $x^0 \in \mathbb{R}^n$ and a parameter $M_0 \geq 0$, the level set satisfies

$$L_0 := \{x \in \mathbb{R}^n \,|\, f(x) \leq f(x^0) + M_0\} \subseteq O. \tag{3}$$

This assumption depends on x^0 and M_0, which are chosen as initial parameters in our algorithm. Under this assumption, f is not necessary to be smooth or convex. If a convex function is finite on O and satisfies (3), this assumption holds. Some other examples of such functions can be found in [11].

Let us turn to the constraint functions F_j, $j = 1, \ldots, m$. We define the total constraint function

$$F(x) := \max\{F_j(x) \,|\, j = 1, \ldots, m\}$$

for all $x \in \mathbb{R}^n$. Then (1) is equivalent to the following problem

$$\min\ f(x) \quad \text{s.t. } F(x) \leq 0. \tag{4}$$

For the constrained functions, we make the following assumption.

Assumption 2.2 For $j = 1, \ldots, m$, the constraint functions F_j are lower-C^2 on the above open bounded set O, so is the total constraint function $F(x)$.

Some important results will be obtained when Assumptions 2.1 and 2.2 are satisfied.

Proposition 2.1 *Under Assumption* 2.1*, the following conclusions hold:*

(1) *The level set* L_0 *is nonempty and compact;*
(2) *The function* f *is bounded below and prox-bounded;*
(3) *There exists a threshold* $\eta_f > 0$ *such that for any* $\eta \geq \eta_f$ *and any given point* $y \in L_0$*, the function* $f + \frac{\eta}{2}|\cdot - y|^2$ *is convex on* L_0*;*
(4) *The function* f *is Lipschitz continuous on* L_0.

The proof of Proposition 2.1 can be found in [11]. As function F is lower-C^2 on O, (2)–(4) in Proposition 2.1 are also valid for it, then we denote the threshold of F by η_F. Moreover, we set $D := \sup_{x,y \in L_0} |x - y|$, which is well defined as the compactivity of L_0.

We say the Slater constraint qualification holds, if $F(\tilde{x}) < 0$ for some $\tilde{x}$. And we also assume the following "augment" Slater constraint qualification holds.

Assumption 2.3 For problem (4), there exists a point $\tilde{x} \in L_0$ satisfying $F(\tilde{x}) < -\frac{\eta_F}{2} D^2$.

Assumption 2.3 is the variant of the Slater constraint qualification in the nonconvex case. It is similar to the Slater constraint qualification of augmented functions $F + \frac{\eta_F}{2}|\cdot - y|^2$ $(\forall y \in L_0)$, which can convexify the constraint function F in local. This assumption is used to prove the existence of the exact penalty function, which is important in our convergence analysis. In addition, it also can ensure that the domain L_0 where all the candidate points y^{k+1} generated from (see the remarks on the algorithm below) and the feasible domain $\{x | F(x) \leq 0\}$ have common parts. In this sense, this assumption seems strong, and future work will seek a weaker assumption that ensures convergence. In Sect. 5.1.1, we explore whether Assumption 2.3 is strictly necessary for numerical convergence.

As F is lower-C^2 on O, it has $F(x) = \max_{t \in T} g_t(x)$ according to the definition, where g_t is of class C^2. For the threshold η_F of F, we have the following result (see the proof of Theorem 10.33 in [21]):

$$\eta_F = \sup_{t \in T, x \in O} |\nabla^2 g_t(x)|. \tag{5}$$

3 The Method

In this section, we present our approach which adds exact penalty function into proximal bundle methods, and apply it for solving constrained nonconvex nonsmooth optimization problem.

3.1 Bundling for Constrained Nonconvex Problem

In order to solve (4), we use the penalty function

$$P(x; c_k) := f(x) + c_k F(x)_+, \ x \in \mathbb{R}^n, \tag{6}$$

where $F(x)_+ := \max\{F(x), 0\}$ and $c_k \in \mathbb{R}$ is a penalty parameter. Under our assumptions, f and F_+ are lower-C^2 functions, so is the penalty function P. Let η_P^k denotes the threshold of P (related to the penalty parameter c_k), which is not greater than $\eta_f + c_k \eta_F$.

At the k-th iteration, the bundle algorithm generates two sequences of points, the candidate points denoted by y^k and the stability centers denoted by x^k (see [2]). In classical bundle methods, by using the past candidate points y^i and their subgradients $g^i \in \partial f(y^i)$, the piecewise linear model of the convex function f is constructed as $\hat{f}(x; k) := \max\{f(y^i) + \langle g^i, x - y^i\rangle \mid i \in I_k\}$, with $I_k \subseteq \{0, 1, \ldots, k\}$. For problems involving the constraint function F, we can extend the piecewise linear model to the penalty function P as $\hat{P}(x; c; k) := \max\{f(y^i) + c_k F(y^i)_+ + \langle g^i + c_k h^i, x - y^i\rangle \mid i \in I_k\}$, where $h^i \in \partial F(y^i)_+$. Since the function $F(x)_+ = \max\{F(x), 0\}$, the subdifferential of $F(x)_+$ at a point x can be defined as

$$\partial F(x)_+ = \begin{cases} \{0\}, & \text{if } F(x) < 0, \\ \operatorname{conv}\{0, \partial F(x)\}, & \text{if } F(x) = 0, \\ \operatorname{conv}\{\partial F(x)\}, & \text{if } F(x) > 0, \end{cases}$$

where *conv* denotes the convex hull (see [22]), and the subgradient can still be denoted by h for convenience. We define the linearization errors for f and F at x^k, respectively by

$$e_{f,i}^k := f(x^k) - f(y^i) - \langle g^i, x^k - y^i\rangle, \quad e_{F,i}^k := F(x^k)_+ - F(y^i)_+ - \langle h^i, x^k - y^i\rangle, \tag{7}$$

and with these notations, the bundle gathering the past information includes such couples:

$$\left\{(e_{f,i}^k, g^i);\ i \in I_k\right\} \text{ and } \left\{c_k(e_{F,i}^k, h^i);\ i \in I_k\right\}.$$

If f and F are convex functions, then $g^i \in \partial_{e_{f,i}^k} f(x^k)$ and $h^i \in \partial_{e_{F,i}^k} F(x^k)_+$ hold.

For convex functions, linearization error is always non-negative [2]. But in nonconvex cases, it may yield negative linearization error which has a great influence on the convergence of algorithm. To overcome this drawback, we use the augmented functions defined by current stability center x^k, which were proposed by Hare and Sagastizábal in [11], and now extend it to our penalty function P,

$$P_k(x) := f(x) + c_k F(x)_+ + \frac{\eta_k}{2}|x - x^k|^2, \tag{8}$$

where η_k is called convexification parameter. Therefore, the bundle includes some additional information:

$$d_i^k := |y^i - x^k|^2/2 \text{ and } \Delta_i^k := y^i - x^k. \tag{9}$$

Our motivation is to "convexify" the penalty function P by P_k in (8), and then we use the relationship $p_{R_k}P(x) = p_{\mu_k}(P_k)(x)$ from (2) to establish the connection between them. Then R_k is split into two non-negative terms η_k and μ_k satisfying $R_k = \eta_k + \mu_k$. We refer to μ_k as the prox-parameter.

Similarly, the past information is used to define the piecewise linear model $\widehat{P}_k$ of P_k,

$$\begin{aligned} \widehat{P}_k(x) &= \max_{i\in I_k}\left\{ f(y^i) + c_k F(y^i)_+ + \eta_k d_i^k + \left\langle (g^i + c_k h^i + \eta_k \Delta_i^k), x - y^i \right\rangle \right\} \\ &= f(x^k) + c_k F(x^k)_+ + \max_{i\in I_k}\left\{ -(e_i^k + \eta_k d_i^k) + \left\langle (g^i + c_k h^i + \eta_k \Delta_i^k), x - x^k \right\rangle \right\}, \end{aligned} \tag{10}$$

where $e_i^k := e_{f,i}^k + c_k e_{F,i}^k$.

In proximal bundle methods [2], x^k represents the prox-center for the current iteration and the algorithm proceeds by defining the next candidate point y^{k+1} as the solution to a certain proximal mapping. In our method, y^{k+1} is the solution to the following quadratic programming (QP) problem:

$$\min_x \left\{ \widehat{P}_k(x) + \frac{\mu_k}{2}|x - x^k|^2 \right\}. \tag{11}$$

That is $y^{k+1} := \arg\min_x \left\{ \widehat{P}_k(x) + \frac{\mu_k}{2}|x - x^k|^2 \right\} = p_{\mu_k}\widehat{P}_k(x^k)$. Note that the point y^{k+1} is uniquely obtained due to the convexity of $\widehat{P}_k$ (see [3]).

Theorem 3.1 *Let y^{k+1} be the unique solution to (11), then*

$$y^{k+1} = x^k - \frac{1}{\mu_k}\left[\sum_{i\in I_k} \bar{\alpha}_i (g^i + c_k h^i + \eta_k \Delta_i^k) \right], \tag{12}$$

where $\bar{\alpha} \in \mathbb{R}_+^{|I_k|}$ is the solution to the following problem

$$\min_\alpha \left\{ \frac{1}{2\mu_k}\Big| \sum_{i\in I_k} \alpha_i (g^i + c_k h^i + \eta_k \Delta_i^k) \Big|^2 + \sum_{i\in I_k} \alpha_i (e_i^k + \eta_k d_i^k) \right\} \quad \text{s.t.} \sum_{i\in I_k} \alpha_i = 1. \tag{13}$$

Proof Obviously QP problem (11) is equivalent to solving the following problem

$$\min_{x,v} \left(v + \frac{\mu_k}{2}|x - x^k|^2 \right) \quad \text{s.t.} -(e_i^k + \eta_k d_i^k) + \left\langle g^i + c_k h^i + \eta_k \Delta_i^k, x - x^k \right\rangle \le v,\ i \in I_k.$$

The corresponding Lagrangian function [3] is

$$L(x, v, \alpha) = v + \frac{\mu_k}{2}|x - x^k|^2 + \sum_{i \in I_k} \alpha_i \Big(-(e_i^k + \eta_k d_i^k) + \langle g^i + c_k h^i + \eta_k \Delta_i^k, x - x^k \rangle - v \Big),$$

for $\alpha \in \mathbb{R}_+^{|I_k|}$. Note that $v \in \mathbb{R}$ and the term L which includes v is $(1 - \sum_{i \in I_k} \alpha_i)v$, so $\sum_{i \in I_k} \alpha_i$ is forced to be 1 for L to be finite and then v vanishes. So y^{k+1} and $\bar{\alpha}(\sum_{i \in I_k} \alpha_i = 1)$ solve the primal and dual problems: $\min_x \max_\alpha L(x, \alpha) = \max_\alpha \min_x L(x, \alpha)$. Consider the dual problem, for each fixed α, set $x(\alpha) := \arg\min_x L(x, \alpha)$. It follows from $0 = \nabla_x L\big(x(\alpha), \alpha\big)$ that

$$0 = \mu_k\big(x(\alpha) - x^k\big) + \sum_{i \in I_k} \alpha_i (g^i + c_k h^i + \eta_k \Delta_i^k). \tag{14}$$

In particular, when $\alpha = \bar{\alpha}$, since $x(\bar{\alpha}) = y^{k+1}$, we obtain (12). And from (14) we obtain

$$\begin{aligned} \mu_k \big|x(\alpha) - x^k\big|^2 &= \frac{1}{\mu_k} \Big| \sum_{i \in I_k} \alpha_i (g^i + c_k h^i + \eta_k \Delta_i^k) \Big|^2 \\ &= -\sum_{i \in I_k} \alpha_i \big\langle g^i + c_k h^i + \eta_k \Delta_i^k, x(\alpha) - x^k \big\rangle. \end{aligned}$$

Then $L(x(\alpha), \alpha)$, which is $\frac{\mu_k}{2}|x(\alpha) - x^k|^2 + \sum_{i \in I_k} \alpha_i \Big(-(e_i^k + \eta_k d_i^k) + \langle g^i + c_k h^i + \eta_k \Delta_i^k, x(\alpha) - x^k \rangle \Big)$, equals to

$$-\frac{1}{2\mu_k} \Big| \sum_{i \in I_k} \alpha_i (g^i + c_k h^i + \eta_k \Delta_i^k) \Big|^2 - \sum_{i \in I_k} \alpha_i (e_i^k + \eta_k d_i^k). \tag{15}$$

Since $\bar{\alpha}$ solves $\max_\alpha L\big(x(\alpha), \alpha\big)$, which is equal to $-\min_\alpha -L\big(x(\alpha), \alpha\big)$, we get the conclusion. □

For convenience, we simplify some notations. First, we define $I_k^{act} := \{i \in I_k, \bar{\alpha}_i > 0\}$, then set

$$\hat{s}^k := \sum_{i \in I_k^{act}} \bar{\alpha}_i (g^i + c_k h^i + \eta_k \Delta_i^k), \quad e^k := \sum_{i \in I_k^{act}} \bar{\alpha}_i e_i^k, \quad d^k := \sum_{i \in I_k^{act}} \bar{\alpha}_i d_i^k,$$

$$\Delta^k := \sum_{i \in I_k^{act}} \bar{\alpha}_i \Delta_i^k \tag{16}$$

$$\varepsilon_k := e^k + \eta_k d^k. \tag{17}$$

Theorem 3.2 *Let $\hat{s}^k$ and ε_k be defined as above, then the following conclusions hold:*

$$(1)\ \hat{s}^k \in \partial \widehat{P}_k(y^{k+1}). \qquad (2)\ \hat{s}^k \in \partial_{\varepsilon_k} P_k(x^k), \text{ if } \eta_k \geq \eta_P^k.$$

Proof Since $y^{k+1} := p_{\mu_k} \widehat{P}_k(x^k)$ is the unique solution to (11), the optimality condition implies $0 \in \partial \widehat{P}_k(y^{k+1}) + \mu_k(y^{k+1} - x^k)$, so (1) is true by $y^{k+1} - x^k = -\frac{1}{\mu_k}\hat{s}^k$ from (12), and $\forall x \in \mathbb{R}^n$, we have

$$\begin{aligned}\widehat{P}_k(x) &\geq \widehat{P}_k(y^{k+1}) + \langle \hat{s}^k, x - y^{k+1}\rangle = \widehat{P}_k(y^{k+1}) + \langle \hat{s}^k, x - x^k\rangle - \langle \hat{s}^k, y^{k+1} - x^k\rangle \\ &= \widehat{P}_k(y^{k+1}) + \langle \hat{s}^k, x - x^k\rangle + \frac{1}{\mu_k}|\hat{s}^k|^2 = P_k(x^k) + \langle \hat{s}^k, x - x^k\rangle \\ &\quad -\Big(P_k(x^k) - \widehat{P}_k(y^{k+1}) - \frac{1}{\mu_k}|\hat{s}^k|^2\Big).\end{aligned}$$

According to the definition of Lagrangian function L in Theorem 3.1,

$$\begin{aligned}L(y^{k+1}, \bar{\alpha}) &= \min_x \max_\alpha L(x, \alpha) = \widehat{P}_k(y^{k+1}) + \frac{\mu_k}{2}|y^{k+1} - x^k|^2 - \big(f(x^k) + c_k F(x^k)_+\big) \\ &= \widehat{P}_k(y^{k+1}) + \frac{\mu_k}{2}|y^{k+1} - x^k|^2 - P_k(x^k),\end{aligned}$$

take $\alpha = \bar{\alpha}$ in (15), since $x(\bar{\alpha}) = y^{k+1}$, then (15) is $L(y^{k+1}, \bar{\alpha}) = -\frac{1}{2\mu_k}|\hat{s}^k|^2 - \varepsilon_k$, we have

$$\varepsilon_k = P_k(x^k) - \widehat{P}_k(y^{k+1}) - \frac{\mu_k}{2}|y^{k+1} - x^k|^2 - \frac{1}{2\mu_k}|\hat{s}^k|^2 = P_k(x^k) - \widehat{P}_k(y^{k+1}) - \frac{1}{\mu_k}|\hat{s}^k|^2.$$

Finally, if $\eta_k \geq \eta_P^k$, the function P_k is convex, then the inequality $P_k(x) \geq \widehat{P}_k(x)$ holds for each x, and

$$P_k(x) \geq P_k(x^k) + \langle \hat{s}^k, x - x^k\rangle - \varepsilon_k,$$

which means $\hat{s}^k \in \partial_{\varepsilon_k} P_k(x^k)$. □

In contrast to cutting plane methods (Cheney and Goldstein [23]; Kelley [24]), which has to solve LP subproblems [2], proximal bundle methods need to solve QP problem. The additional computational burden of solving QP subproblems is the price to be paid to clean the model, that is, to compress the bundle by using the aggregation techniques (see [2]). In our method, we call $\hat{s}^k$ the aggregate subgradient, and $\hat{s}^k$, e^k, d^k, and Δ^k in (16) are all attained from aggregation techniques. The corresponding aggregate linearization function is

$$\widehat{P}_k(y^{k+1}) + \langle \hat{s}^k, x - y^{k+1}\rangle. \tag{18}$$

Proposition 3.1 *The aggregate linearization function in (18) can be expressed as*

$$P_k(x^k) + \langle \hat{s}^k, x - x^k\rangle - \varepsilon_k.$$

Proof In the proof of Theorem 3.2, we know $\varepsilon_k = P_k(x^k) - \widehat{P}_k(y^{k+1}) - \frac{1}{\mu_k}|\hat{s}^k|^2$, then

$$\begin{aligned}\widehat{P}_k(y^{k+1}) + \langle \hat{s}^k, x - y^{k+1}\rangle &= P_k(x^k) - \frac{1}{\mu_k}|\hat{s}^k|^2 - \varepsilon_k + \langle \hat{s}^k, x - y^{k+1}\rangle \\ &= P_k(x^k) + \langle \hat{s}^k, y^{k+1} - x^k\rangle - \varepsilon_k + \langle \hat{s}^k, x - y^{k+1}\rangle \\ &= P_k(x^k) + \langle \hat{s}^k, x - x^k\rangle - \varepsilon_k.\end{aligned}$$

□

Let us turn to the parameters in our method. First, we consider the convexification parameter η_k. In classical bundle methods for unconstrained convex problem, linearization error of f at x^k is $e^k_{f,i}$, and we have $g^i \in \partial_{e^k_{f,i}} f(x^k)$. Here, linearization error is expanded to $e^k_i + \eta_k d^k_i$, and the following holds:

$$g^i + c_k h^i + \eta_k \Delta^k_i \in \partial_{e^k_i + \eta_k d^k_i} \widehat{P}_k(x^k) \quad \text{if } e^k_i + \eta_k d^k_i \geq 0 \text{ for all } i \in I_k. \tag{19}$$

At the first few iterations, $P_k(x)$ in (8) may still not be convex, because η_k may be less than its threshold η^k_P. But here, we can guarantee the linearization error in (19) to be non-negative by setting η_k not less than the minimal value:

$$\eta^{\min}_k := \max_{\substack{i \in I_k \\ d^k_i \neq 0}} -\frac{e^k_i}{d^k_i} = \max_{\substack{i \in I_k \\ d^k_i \neq 0}} -\frac{e^k_{f,i} + c_k e^k_{F,i}}{d^k_i}. \tag{20}$$

Other principles for choosing η_k refer to [11] and [12].

Then let us turn to the prox-parameter μ_k. In bundle methods, the next candidate point y^{k+1} may not bring the decline of the objective function, sometimes even make the objective function increase too much. Similar to the idea in [11], our model prox-parameter μ_k is designed to avoid this phenomenon. If $f(y^{k+1}) > f(x^k) + M_0$ happens, i.e., the increase of the objective function is unacceptable, we adjust the model prox-parameter μ_k by multiplying a given constant $\Gamma_2 > 1$, else we do not change its value.

At last, we discuss the updating for the penalty parameter c_k. Before that, we define the predicted decrease δ_{k+1} when the next candidate point y_{k+1} is generated, which is used to measure the decrease between convexified function $P_k(x)$ and its piecewise linear model $\widehat{P}_k(x)$ at the point y^{k+1}, i.e.,

$$\delta_{k+1} := f(x^k) + c_k F(x^k)_+ + \frac{\eta_k}{2}|y^{k+1} - x^k|^2 - \widehat{P}_k(y^{k+1}). \tag{21}$$

We update c_k with the help of the predicted decrease δ_{k+1}, that is, if

$$\delta_{k+1} < \kappa F(x^k), \tag{22}$$

c_k is increased, where $\kappa \in [0, 1)$ is called a contraction factor. We will prove that δ_{k+1} is non-negative in Lemma 3.1, so (22) holds only if the current stability center x^k is infeasible. The idea of this rule can be found in [19]: when the fraction of constraint violation at x^k is greater than the predicted decrease, increase the penalty parameter because larger penalty parameter in the algorithm may make iteration points more close to the feasible domain. Some other principles for updating c_k can also be found in [19].

3.2 Algorithm

In this subsection, we present our method in details.

Algorithm

Step 0 (Initialization) Choose an initial starting point x^0, an unacceptable increase parameter $M_0 > 0$, an accuracy tolerance ε_P, a feasibility tolerance ε_F, an improvement parameter $m \in (0, 1)$, three growth parameters $\Gamma_1 > 1$, $\Gamma_2 > 1$, $\Gamma_3 > 1$, a prox-parameter $R_0 > 0$, an infeasibility contraction factor $\kappa \in (0, 1)$, and an initial penalty coefficient $c_0 > 0$. Set $y^0 = x^0$, and compute $g^0 \in \partial f(y^0)$, $h^0 \in \partial F(y^0)_+$. Initialize $e^0_{f,0} = e^0_{F,0} = d^0_0 = \Delta^0_0 = 0$, and choose $\mu_0 = R_0$, $\eta_0 = R_0 - \mu_0 = 0$. Set the counter $k = I_0 = 0$.

Step 1 (Calculation of the candidate point) The bundle gathers the information obtained from the past iteration points $\bigcup_{i\in I_k}\{(e^k_i, g^i, h^i, d^k_i, \Delta^k_i)\}$, and $\eta_k = R_k - \mu_k$. Construct the piecewise linear model $\widehat{P}_k$ as (10), and obtain $\bar{\alpha}$ by solving (13), and compute y^{k+1} through (12). Then compute the predicted decrease δ_{k+1} through (21).

Step 2 (Stopping criterion) If $\delta_{k+1} \le \varepsilon_P$ and $F(x^k) \le \varepsilon_F$, terminate; otherwise, continue.

Step 3 (Update of prox-parameter) If $f(y^{k+1}) > f(x^k) + M_0$, the objective increase is unacceptable, then $\mu_k := \Gamma_2\mu_k$, go to Step 1; otherwise, set $\mu_{k+1} := \mu_k$.

Step 4 (Update of penalty parameter) If $\delta_{k+1} < \kappa F(x^k)$, set $c_{k+1} := \Gamma_3 c_k$; otherwise, set $c_{k+1} := c_k$.

Step 5 (Serious step test) If $f(y^{k+1})+c_k F(y^{k+1})_+ \le f(x^k)+c_k F(x^k)_+ - m\delta_{k+1}$, set $x^{k+1} := y^{k+1}$ (serious step); otherwise, set $x^{k+1} := x^k$ (null step).

Step 6 (Bundle update) Calculate $g^{k+1} \in \partial f(y^{k+1})$, $h^{k+1} \in \partial F(y^{k+1})_+$, and update

$$\begin{aligned}
e^{k+1}_{f,i} &:= e^k_{f,i} + f(x^{k+1}) - f(x^k) - \langle g^i, x^{k+1} - x^k\rangle,\\
d^{k+1}_i &:= d^k_i + |x^{k+1} - x^k|^2/2 - \langle \Delta^k_i, x^{k+1} - x^k\rangle,\\
e^{k+1}_{F,i} &:= e^k_{F,i} + F(x^{k+1})_+ - F(x^k)_+ - \langle h^i, x^{k+1} - x^k\rangle,\\
\Delta^{k+1}_i &:= \Delta^k_i + x^k - x^{k+1}.
\end{aligned}$$

Step 7 (Update of convexification parameter) Set

$$\begin{cases} \eta_{k+1} := \eta_k, & \text{if } \eta^{\min}_{k+1} \le \eta_k,\\ \eta_{k+1} := \Gamma_1\eta^{\min}_{k+1}, & \text{if } \eta^{\min}_{k+1} > \eta_k, \end{cases} \tag{23}$$

where $\eta^{\min}_{k+1}$ is given by (20). Increase k by 1 and go to Step 1.

Some remarks on the algorithm are given below.

In Step 3, the aim to increase μ_k is to force all the candidate points y^{k+1} to go into the set L_0. But it may lead to the cycle between Steps 1 and 3. We show that the cycle must be finite in the next section.

In Step 7, the updating for the convexification parameters guarantee $\eta_k \geq \eta_k^{\min}$ for all iterations, so that we have $e_i^k + \eta_k d_i^k \geq 0$ and $\varepsilon_k \geq 0$.

The following lemma is needed for convergence analysis.

Lemma 3.1 *The predicted decrease defined in* (21) *is non-negative and can be rewritten as*

$$\delta_{k+1} = \frac{R_k + \mu_k}{2} |y^{k+1} - x^k|^2 + \varepsilon_k. \tag{24}$$

Proof We have known that $\widehat{P}_k(y^{k+1}) = P_k(x^k) + \langle \hat{s}^k, y^{k+1} - x^k \rangle - \varepsilon_k$ and $y^{k+1} = x^k - \frac{1}{\mu_k}\hat{s}^k$ from Proposition 3.1 and Theorem 3.1, respectively, then

$$\begin{aligned}
\delta_{k+1} &= f(x^k) + c_k F(x^k)_+ + \frac{\eta_k}{2}|y^{k+1} - x^k|^2 - \widehat{P}_k(y^{k+1}) \\
&= \frac{\eta_k}{2}|y^{k+1} - x^k|^2 - \langle \hat{s}^k, y^{k+1} - x^k \rangle + \varepsilon_k = \frac{R_k + \mu_k}{2}|y^{k+1} - x^k|^2 + \varepsilon_k.
\end{aligned}$$

At last, the updating for the convexification parameters can guarantee $\varepsilon_k \geq 0$, so is the δ_{k+1}. □

4 Convergence Analysis

4.1 The Stability of Parameters

First, we show that the prox-parameter sequence $\{\mu_k\}$ will become constant as the loop of the algorithm, so the cycle between Step 1 and Step 3 is also finite. Therefore, our algorithm is well defined.

Lemma 4.1 *If f and F are lower-C^2 on an open bounded set O and f satisfies* (3), *with the running of the algorithm, the model prox-parameter sequence $\{\mu_k\}$ becomes a constant sequence.*

Proof We divide the proof into two parts. First, suppose the penalty sequence $\{c_k\}$ is bounded, i.e., there exists a constant $M > 0$ such that $|c_k| \leq M$ for all k. According to the definition of $p_{\mu_k}\widehat{P}_k(x^k)$, it is in the set $\{x | \widehat{P}_k(x) + \frac{\mu_k}{2}|x - x^k|^2 \leq \widehat{P}_k(x^k) + \frac{\mu_k}{2}|x^k - x^k|^2\}$. Taking the current serious step $i = i_k$, (19) becomes $g^{i_k} + c_{i_k}h^{i_k} \in \partial \widehat{P}_{i_k}(x^{i_k})$, then we have $g^{i_k} + c_{i_k}h^{i_k} \in \partial \widehat{P}_k(x^k)$ obviously. Note that f is Lipschitz continuous on L_0 with the Lipschitz constant T_1, $g^{i_k} \in \partial f(x^k)$, and $x^k \in L_0$, thus $|g^{i_k}| \leq T_1$. Similarly, F has the Lipschitz constant T_2 on L_0, and $|h^{i_k}| \leq T_2$, hence,

$$\begin{aligned}
p_{\mu_k}\widehat{P}_k(x^k) &\in \{x | \widehat{P}_k(x^k) + \langle g^{i_k} + c_{i_k}h^{i_k}, x - x^k \rangle + \frac{\mu_k}{2}|x - x^k|^2 \leq \widehat{P}_k(x^k)\} \\
&\subseteq \{x | \frac{\mu_k}{2}|x - x^k|^2 - |g^{i_k} + c_{i_k}h^{i_k}||x - x^k| \leq 0)\}
\end{aligned}$$

$$\subseteq \left\{x \middle| \frac{\mu_k}{2}|x - x^k| \le |g^{i_k}| + |c_{i_k}||h^{i_k}|)\right\}$$
$$\subseteq \left\{x \middle| \frac{\mu_k}{2}|x - x^k| \le T_1 + |c_{i_k}|T_2\right\} \subseteq \left\{x \middle| |x - x^k| \le \frac{2T}{\mu_k}\right\},$$

where $T = T_1 + MT_2$. As $k \to \infty$, μ_k will become large enough so that $\frac{2T}{\mu_k}$ is sufficiently small, note that the continuity of f, we cannot have $f(y^{k+1}) > f(x^k) + M_0$, the update of μ_k will stop finally.

Secondly, if $\{c_k\}$ is unbounded, for convenience, we assume that $\delta_{k+1} < \kappa F(x^k)$ for all k (Step 4), it follows from (24) that $\frac{R_k+\mu_k}{2}|y^{k+1} - x^k|^2 < \kappa F(x^k)$ as $\varepsilon_k \ge 0$. Then

$$|y^{k+1} - x^k|^2 < \frac{2\kappa F(x^k)}{\eta_k + 2\mu_k}.$$

Since $x^k \in L_0$, $|F(x^k)|$ is bounded. As μ_k increases, the denominator $\eta_k + 2\mu_k$ will become large enough, so y^{k+1} approaches to x^k sufficiently, and μ_k will be stabilized. □

Next we turn to the penalty parameter c_k. If c_k does not stabilize, the "nonconvexity" of $c_k F(x)$ may be "magnified" as c_k increases, and so is η_k. So the exactness of the penalty function is important.

Theorem 4.1 *If the function F satisfies the "augment" Slater constraint qualification, i.e., there is a point $\tilde{x} \in L_0$ satisfying $F(\tilde{x}) < -\frac{\eta_F}{2}D^2$, then there exist k_c and $\bar{c}$ such that $c_k = \bar{c}$ for all $k \ge k_c$.*

Proof For contradiction, we suppose $c_k \to \infty$ as $k \to \infty$, then there exists a subsequence x^{k_j} such that $\delta_{k_j+1} < \kappa F(x^{k_j})$, and $\delta_{k_j+1} \ge 0$ by (24), so $F(x^{k_j}) > 0$ for all j. As in classical bundle method for convex function, we also have that aggregation linearization function stays below our convexification penalty function, i.e., $f(x) + c_k F(x)_+ + \frac{\eta_P^k}{2}|x - x^k|^2 \ge P_k(x^k) + \langle \hat{s}^k, x - x^k\rangle - \varepsilon_k$, for all $x \in \mathbb{R}^n$. The threshold satisfies $\eta_P^k \le \eta_f + c_k\eta_F$, and taking $k = k_j$ we have

$$\begin{aligned} f(x) + c_{k_j}F(x)_+ + \frac{\eta_f + c_{k_j}\eta_F}{2}|x - x^{k_j}|^2 \\ \ge f(x^{k_j}) + c_{k_j}F(x^{k_j}) + \langle \hat{s}^{k_j}, x - x^{k_j}\rangle - \varepsilon_{k_j}, \end{aligned} \tag{25}$$

for all $x \in \mathbb{R}^n$. We find that $F(x)_+$ can be substituted by $F(x)$ in (25) because the left-hand side of the above inequality is convex and all the cutting planes are tangent at x^{k_j} with $F(x^{k_j}) > 0$. Notice $\hat{s}^{k_j} = \mu_{k_j}(x^{k_j} - y^{k_j+1})$, and $\varepsilon_{k_j} \le \delta_{k_j+1} < \kappa F(x^{k_j})$, setting $x = \tilde{x}$, we obtain

$$\begin{aligned} f(\tilde{x}) + c_{k_j}F(\tilde{x}) + \frac{\eta_f + c_{k_j}\eta_F}{2}|\tilde{x} - x^{k_j}|^2 > f(x^{k_j}) + c_{k_j}F(x^{k_j}) \\ -|\mu_{k_j}||x^{k_j} - y^{k_j+1}||\tilde{x} - x^{k_j}| - \kappa F(x^{k_j}), \end{aligned} \tag{26}$$

we have $\tilde{x}$, x^{k_j}, and y^{k_j+1} belong to L_0 for all j, $f(\tilde{x})$, $|x^{k_j} - y^{k_j+1}|$ together with $|\tilde{x} - x^{k_j}|$ are all bounded. Divide (4.2) by c_{k_j}, let $j \to \infty$ and suppose $x^{k_j} \to \bar{x} \in L_0$ (L_0 is compact). We have known μ_{k_j} is convergent by Lemma 4.1, then $F(\tilde{x}) + \frac{\eta_F}{2}|\tilde{x} - \bar{x}|^2 \geq F(\bar{x}) \geq 0$, which leads to a contradiction. □

According to the theorem above, c_k can be stabilized, so is the threshold sequence $\{\eta_P^k\}$. Let η_P be the stabilized value for $\{\eta_P^k\}$. Then we consider the convexification parameter η_k.

Lemma 4.2 *There exist k_η and $\bar{\eta}$ such that $\eta_k = \bar{\eta}$ for all $k \geq k_\eta$.*

Proof The update of the convexification parameter η_k in Step 7 is nondecreasing. If the sequence does not stabilize, i.e., for some iteration k_η, algorithm produces a convexification parameter $\eta_{k_\eta} \geq \eta_P$, and the function $P_k(x)$ is convex on L_0. After this particular iteration, we have $e_i^k + \eta_{k_\eta} d_i^k \geq 0$ for all $i \in I_k$ and k because the linearization error for a cutting-plane model of a convex function is always non-negative. Therefore, $\eta_{k_\eta} \geq \eta_k^{\min}$ for all $k \geq k_\eta$ and η_k is not updated any more. □

Now we know that all these three parameters in our algorithm stabilize. Set k' (not less than k_c and k_η) to be a special iteration such that

$$\mu_k = \bar{\mu},\ \eta_k = \bar{\eta},\ c_k = \bar{c} \text{ and } R_k = \bar{R} := \bar{\mu} + \bar{\eta} \quad \text{for all } k \geq k'.$$

4.2 Asymptotic Feasibility and Optimality

In this part, we will study the convergence properties of our algorithm. We assume the tolerances $\varepsilon_P = \varepsilon_F = 0$, and thus explore the asymptotic behavior of the algorithm. Firstly, if the algorithm stops at some iteration k with $\delta_{k+1} = 0$, from (24) we have $\frac{R_k+\mu_k}{2}|y^{k+1} - x^k|^2 + \varepsilon_k = 0$. Suppose that η_k is sufficiently large such that $\eta_k \geq \eta_P^k$, then $P_k(x)$ is convex on L_0. Notice that $\varepsilon_k \geq 0$, so we have $y^{k+1} = x^k$, which implies $x^k = p_{\mu_k}\widehat{P}_k(x^k)$. Substitute $\delta_{k+1} = 0$ into (21), we obtain

$$f(x^k) + c_k F(x^k)_+ + \frac{\eta_k}{2}|y^{k+1} - x^k|^2 = \widehat{P}_k(y^{k+1}).$$

Taking $y^{k+1} = x^k$, then by using the definition of $p_{\mu_k}\widehat{P}_k(x^k)$ and the convexity of P_k, we have

$$\begin{aligned} f(x^k) + c_k F(x^k)_+ = \widehat{P}_k(x^k) &\leq \widehat{P}_k(x) + \frac{\mu_k}{2}|x - x^k|^2 \\ &\leq f(x) + c_k F(x)_+ + \frac{\eta_k}{2}|x - x^k|^2 + \frac{\mu_k}{2}|x - x^k|^2 \\ &= f(x) + c_k F(x)_+ + \frac{R_k}{2}|x - x^k|^2 \end{aligned}$$

for all $x \in L_0$. In other words, $x^k = p_{R_k}(f(x^k) + c_k F(x^k)_+)$, which means $0 \in \partial(f(x^k) + c_k F(x^k)_+)$ by [12, Proposition 2.1(g)]. Because $\varepsilon_F = 0$, $F(y^{k+1}) = F(x^k) \leq 0$, and x^k is feasible.

The convergence analysis of bundle methods usually includes two different asymptotic cases, depending on whether the algorithm produces a finite or an infinite number of serious steps. Now we will prove that if the stabilized value of the convexification parameter sequence $\{\eta_k\}$ is large enough, then our method, for both cases, converges to the KKT point of the primal problem (1).

Theorem 4.2 *If f and F satisfy Assumption 2.1–2.3, i.e., f and F are lower-C^2 on an open bounded set O, f satisfies (3) and F satisfies the "augment" Slater constraint qualification. Suppose that the tolerances $\varepsilon_P = \varepsilon_F = 0$ and there is no termination. Let $\bar{\eta}$ and η_P be the stabilized values for the convexification parameter sequence $\{\eta_k\}$ and the threshold sequence $\{\eta_P^k\}$, respectively, suppose $\bar{\eta} \geq \eta_P$, then one of the following two conclusions hold,*

(1) *there are infinite serious steps. The sequence $\{x^k\}$ converges to the KKT point of (1).*
(2) *it generates the last serious step $\hat{x}$, followed by infinitely many null steps. The sequence $\{y^k\}$ converges to $\hat{x}$, and $\hat{x}$ is the KKT point of (1).*

Proof For (1), considering a subsequence required, we suppose every candidate point satisfies the descent condition, i.e., $f(x^{k+1}) + c_k F(x^{k+1})_+ \leq f(x^k) + c_k F(x^k)_+ - m\delta_{k+1}$. With infinite serious steps, we have either $f(x^k) + c_k F(x^k)_+ \to -\infty$ or $\delta_{k+1} \to 0$ as $k \to \infty$. By Proposition 2.1, f is bounded below, and $F(x^k)_+ \geq 0$, so $\delta_{k+1} \to 0$ as $k \to \infty$. And $\varepsilon_k \geq 0$, (24) means both $|x^{k+1} - x^k|^2$ and ε_k must converge to 0. Because $\{x^k\}$ is in a compact set L_0, it has a subsequence $\{x^{k_i}\} \to \bar{x} \in L_0$ as $i \to \infty$, thus x^{k_i+1}, as the subsequence of x^{k+1}, also converges to $\bar{x}$ since $|x^{k_i+1} - x^{k_i}|^2 \to 0$. Note that $x^{k+1} = p_{\bar{\mu}} \widehat{P}_k(x^k)$ for $k > k'$, so $\widehat{P}_k(x^{k+1}) + \frac{\bar{\mu}}{2}|x^{k+1} - x^k|^2 \leq \widehat{P}_k(x) + \frac{\bar{\mu}}{2}|x - x^k|^2$, for all $x \in L_0$. Since we suppose $\bar{\eta} \geq \eta_P$, P_k is convex on L_0, and it is above the piecewise linear model $\widehat{P}_k$,

$$\begin{aligned}\widehat{P}_k(x) + \frac{\bar{\mu}}{2}|x - x^k|^2 &\leq f(x) + \bar{c}F(x)_+ + \frac{\bar{\eta}}{2}|x - x^k|^2 + \frac{\bar{\mu}}{2}|x - x^k|^2 \\ &= f(x) + \bar{c}F(x)_+ + \frac{\bar{R}}{2}|x - x^k|^2.\end{aligned}$$

Combine the last two inequalities and take the subsequence $\widehat{P}_{k_i}$ of $\widehat{P}_k$, one has

$$\widehat{P}_{k_i}(x^{k_i+1}) + \frac{\bar{\mu}}{2}|x^{k_i+1} - x^{k_i}|^2 \leq f(x) + \bar{c}F(x)_+ + \frac{\bar{R}}{2}|x - x^{k_i}|^2,$$

let $i \to \infty$, we have $\widehat{P}_{k_i}(x^{k_i+1}) \to f(\bar{x}) + \bar{c}F(\bar{x})_+$ as $\delta_{k_i+1} \to 0$, and $f(\bar{x}) + \bar{c}F(\bar{x})_+ \leq f(x) + \bar{c}F(x)_+ + \frac{\bar{R}}{2}|x - \bar{x}|^2$., i.e., $\bar{x} = p_{\bar{R}}(f(\bar{x}) + \bar{c}F(\bar{x})_+)$, then $0 \in \partial(f(\bar{x}) + \bar{c}F(\bar{x})_+)$ according to [12, Proposition 2.1(g)].

Because c_k is unchanged for $k > k'$, then $\delta_{k+1} \geq \kappa F(x^k)$, we have $\kappa F(\bar{x}) \leq 0$, $\bar{x}$ is feasible since one has $\kappa > 0$. Since f and F are Lipschitz continuous on L_0, and $\bar{x} \in L_0$, so they are strictly continuous (see [21, Definition 9.1]) at $\bar{x}$. According to the conclusion in [21, Excercise 10.10], we have

$$0 \in \partial(f(\bar{x}) + \bar{c}F(\bar{x})_+) \subseteq \partial f(\bar{x}) + \bar{c}\partial F(\bar{x})_+.$$

Notice that $F(\bar{x})_+ = \max\{0, F_1(\bar{x}), \ldots, F_m(\bar{x})\}$ and the results in [21, Excercise 10.51], there exist $m+1$ numbers $\lambda_0, \lambda_1, \ldots, \lambda_m$, satisfying $\lambda_0 + \sum_{j=1}^m \lambda_j = \bar{c}$ and $\lambda_j F_j(x^k) = 0,\ j = 1, \ldots, m$, such that

$$0 \in \partial f(\bar{x}) + \left\{ \sum_{j=1}^m \lambda_j \partial F_j(\bar{x}) \right\}.$$

For (2), suppose $\hat{x}$ is the last serious step generated at the kth iteration, and after that there are only null steps left. The piecewise linear model $\widehat{P}_k$ is a convex function, according to its definition in (10), we immediately have $\widehat{P}_k(\hat{x}) \le f(\hat{x}) + \bar{c}F(\hat{x})$ and $\widehat{P}_k(x) \ge f(y^k) + \bar{c}F(y^k) + \bar{\eta} d_k^k + \langle g^k + \bar{c}h^k, x - y^k\rangle + \bar{\eta}\Delta_k^k$ for $k > k'$ and $\bar{\eta} \ge \eta_P^k$. (notice that $x^k \equiv \hat{x}$ and $\Delta_k^k = y^k - \hat{x}$, $d_k^k = |y^k - \hat{x}|^2/2$). On the other hand, by (12) and $\hat{s}^k \in \partial \widehat{P}_k(y^{k+1})$ in Theorem 3.2 one has

$$\widehat{P}_k(y^{k+1}) + \bar{\mu}\langle \hat{x} - y^{k+1}, x - y^{k+1}\rangle = \widehat{P}_k(y^{k+1}) + \langle \hat{s}^k, x - y^{k+1}\rangle \le \widehat{P}_k(x).$$

Together with $\widehat{P}_k(x) \le \widehat{P}_{k+1}(x)$ (the stability center $\hat{x}$ is fixed), conditions (3c) in [12] is satisfied:

$$\widehat{P}_{k+1}(x) \ge \widehat{P}_k(y^{k+1}) + \bar{\mu}\langle \hat{x} - y^{k+1}, x - y^{k+1}\rangle,$$

for $k > k'$ and $\bar{\eta} \ge \eta_P$. Thus, from Theorem 6 in [12], we have the sequence $\{y^{k+1}\} \to p := p_{\bar{R}}(f(\hat{x}) + \bar{c}F(\hat{x})_+)$ with $\widehat{P}_k(y^{k+1}) \to f(p) + \bar{c}F(p)_+ + \bar{\eta}/2|p - \hat{x}|^2$ as $k \to \infty$. Thus, the predicted decrease is

$$\begin{aligned}
\delta_{k+1} &= f(\hat{x}) + \bar{c}F(\hat{x})_+ + \frac{\bar{\eta}}{2}|y^{k+1} - \hat{x}|^2 - \widehat{P}_k(y^{k+1}) \\
&\to f(\hat{x}) + \bar{c}F(\hat{x})_+ + \frac{\bar{\eta}}{2}|p - \hat{x}|^2 - f(p) - \bar{c}F(p)_+ - \frac{\bar{\eta}}{2}|p - \hat{x}|^2 \\
&= f(\hat{x}) - f(p) + \bar{c}(F(\hat{x})_+ - F(p)_+).
\end{aligned}$$

Since the serious step test does not hold, that is, $f(y^{k+1}) + \bar{c}F(y^{k+1})_+ > f(\hat{x}) + \bar{c}F(\hat{x})_+ - m\delta_{k+1}$ for $k > k'$. Taking the limit we have $f(\hat{x}) + \bar{c}F(\hat{x})_+ \le f(p) + \bar{c}F(p)_+$. But at the same time we have $p = p_{\bar{R}}(f(\hat{x}) + \bar{c}F(\hat{x})_+)$, which means $f(p) + \bar{c}F(p)_+ + \frac{\bar{R}}{2}|p - \hat{x}|^2 \le f(\hat{x}) + \bar{c}F(\hat{x})_+$, so $\hat{x} = p$ and $\hat{x} = p_{\bar{R}}(f(\hat{x}) + \bar{c}F(\hat{x})_+)$, which implies $0 \in \partial(f(\hat{x}) + \bar{c}F(\hat{x})_+)$. Similar as the proof in (1), we obtain $F(\hat{x}) \le 0$ and $\hat{x}$ are the KKT points of our primal problem (1). □

In convergence analysis, we suppose the stabilized convexification parameter $\bar{\eta}$ is greater than the stabilized threshold η_P, it is an essential assumption because $\bar{\eta} < \eta_P$ will lead to the fact that all the linearization errors for the augment function P_k at every stability center x^i $(i \le k)$ are non-negative, but P_k is yet nonconvex. Such situation may happen, the Fig. 5.2 in [11] gave an approximate example, but providing a precise example seems difficult. To avoid this problem, [11] provided an idea which adds a small random element to the solution of the QP subproblem.

5 Numerical Experiments

5.1 The Choice of the Functions

Our numerical experiments are divided into two parts: deterministic tests and stochastic tests.

5.1.1 Deterministic Tests

We prefer a series of polynomial functions as our deterministic objective functions which were used in[11]. For each $i = 1, 2, \dots, n$, the function $h_i : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$h_i(x) := (ix_i^2 - 2x_i - K_1) + \sum_{j=1}^{n} x_j,$$

where K_1 is a fixed constant. There are five classes of test functions defined by h_i in [11], we choose two of them as our objective functions:

$$f_1(x) := \sum_{i=1}^{n} |h_i(x)|, \qquad f_2(x) := \max_{i \in \{1,\dots,n\}} |h_i(x)|. \tag{27}$$

It has been proved in [11] that they are nonsmooth, nonconvex, and globally lower-C^2 and if constant $K_1 = 0$, we have $0 = \min_x f_k$ and $\{0\} \in \arg\min_x f_k$ for $k = 1, 2$.

For deterministic constraint functions, we consider the pointwise maximum of a finite collection of quadratic functions, i.e.,

$$F(x) := \max_{i=1,2,\dots,N} \big\{\langle x, A_i x\rangle + \langle B_i, x\rangle + C_i\big\}, \tag{28}$$

where A_i are $n \times n$ matrices, $B_i \in \mathbb{R}^n$ and $C_i \in \mathbb{R}$ are for $i = 1, 2, \dots, N$. Functions as (28) have many numerical advantages [12]. They are always lower-C^2 and prox-bounded, but may be nonconvex since A_i are not necessary to be positive definite. It is convenient to find their subgradients at any point, and from (5) we can directly evaluate the threshold of each constraint function as $\eta_F = \max_{i=1,2,\dots,N} |A_i + A_i^\top|$.

Our deterministic constraint functions are given in Appendix A in details. First, we give an illustration of convex constraint in three dimension by using functions in [19] (Case 1), then we give some nonconvex constraint functions from 2 to 5 dimension (Case 2–Case 5).

5.1.2 Stochastic Tests

Our stochastic tests include two Groups. We use piecewise quadratic functions (28) as our objective and constraint functions for the first Group (Group I). Here all the coefficients A_i, B_i, and C_i are uniformly distributed in $[-5, 5]$, which are chosen randomly by computer. Since objective and constraint functions are generated randomly, we may not evaluate their optimal values previously. Therefore, for the same stochastic test, we compare our algorithm with the public software SolvOpt [25], which is one of the most efficient codes available for solving nonsmooth optimization.

The second Group (Group II) is designed as follows: choosing functions f_2 in (27) as objective functions and piecewise quadratic functions (28) as constraint functions. Here the coefficients A_i and B_i of the constraint functions are chosen randomly in $[-5, 5]$ by computer, and coefficients C_i are chosen randomly in $[-10, 0]$ for the feasible domain contain the optimal value 0 of our objective functions, which is used to verify the accuracy of our numerical results.

In all of our stochastic tests, we do not check whether the "augment" Slater constraint qualification (Assumption 2.3) is satisfied and see how the algorithm performs.

5.2 Numerical Results

In deterministic tests, we have 2 objective functions and 5 constraint functions, which provide 10 tests. And in stochastic ones, we report 4 tests for Group I which from 2 to 5 dimension, and report 40 tests from 3 to 10 dimension for Group II. All of these tests are implemented in Matlab v7.10.0 and the parameters of these tests are the same and listed below:

– the initial starting point $x_0 = (1, 1, \ldots, 1)$,

– the accuracy tolerance $\varepsilon_P = 10^{-6}$,

– the feasibility tolerance $\varepsilon_F = 10^{-6}$,

– the unacceptable increase parameter $M_0 = 5$,

– the prox-parameter $R_0 = 10$,

– the infeasibility contraction factor $\kappa = 0.1$,

– the growth parameters $\Gamma_1 = \Gamma_2 = \Gamma_3 = 1.1$,

– the improvement parameter $m = 0.05$,

– the initial penalty coefficient $c_0 = 10$,

– the constant $K_1 = 0$.

Our results for deterministic tests are summarized in Tables 1–2, in which n denotes dimension, x^* represents the solution to our problems, $f_i(x^*)$ represents the optimal value, $F(x^*)$ represents the value of constraint function, and k is the number of iterations.

Our results for stochastic tests in Group I are summarized in Table 3, in which the results of our algorithm and SolvOpt are listed. $f(x^*)$ represents the optimal value, and n, x^*, $F(x^*)$, and k are the same as the ones in Tables 1–2.

Table 1 Results for objective function f_1

n	x^*	$f_1(x^*)$	$F(x^*)$	k
3°	$(-0.000776, 0.006061, -0.003538)$	0.022465	−0.616238	8
2	$(0.114980, 0.179219)$	0.077459	−13.9222	13
3	$(-0.030893, -0.033260, 0.001411)$	0.071550	−31.9462	20
4	$(-0.000036, 0.000036, 0.000023, 0.000023)$	0.000144	−3.00103	15
5	$(0.000009, -0.000017, 0.000009, 0.000040, -0.000020)$	0.000183	−35.0012	15

$\circ$ means the result for convex constraint optimization

Table 2 Results for objective function f_2

n	x^*	$f_2(x^*)$	$F(x^*)$	k
3°	$(0.000000, -0.000277, -0.000000)$	0.000278	−0.616722	17
2	$(0.001207, 0.001207)$	0.000002	−9.03864	38
3	$(-0.000465, 0.000000, 0.000000)$	0.000466	−23.0051	18
4	$(-0.000097, -0.000590, 0.000097, 0.000098)$	0.000689	−2.98424	18
5	$(0.000013, 0.000014, 0.000013, 0.000014, 0.000013)$	0.000042	−35.0017	23

$\circ$ means the result for convex constraint optimization

Table 3 Results for stochastic tests

n	Algorithm type	x^*	$f(x^*)$	$F(x^*)$	k
2	Algorithm	$(1.02070, 2.58478)$	−5.01095	0.000000	151
2	SolvOpt	$(1.02132, 2.58736)$	−5.01096	0.000000	34
3	Algorithm	$(0.560605, -0.028945, 0.754963)$	−2.17442	−0.000001	50
3	SolvOpt	$(0.616570, -0.035292, 0.782782)$	−2.17851	−0.246929	41
4	Algorithm	$(0.607571, 0.259681, 0.627346, 0.243467)$	−3.56993	−0.000280	144
4	SolvOpt	$(0.625475, 0.188429, 0.688740, 0.315877)$	−3.58366	0.000000	88
5	Algorithm	$(-0.199213, -0.636026, 0.151654, 1.04669, 1.60657)$	−15.9228	−4.16289	163
5	SolvOpt	$(-0.165700, -0.723463, 0.048232, 0.972789, 1.51051)$	−14.4608	−5.0017	123

The graphs of the objective functions and the feasible domain we performed are presented below when $n=2$, in which the red areas under the figures of the objective functions represent the feasible domain. (a) and (b) represent deterministic tests and (c) is the stochastic case in Group I (Fig. 1).

Our results for stochastic tests in Group II are summarized in Tables 4, 5, 6, 7, 8, 9, 10, and 11 in Appendix B. Notations x^*, $f_2(x^*)$, $F(x^*)$, and k are the same as the ones in Tables 1–2, also the initial value of objective and constraint functions $f_2(x_0)$, $F(x_0)$ are listed.

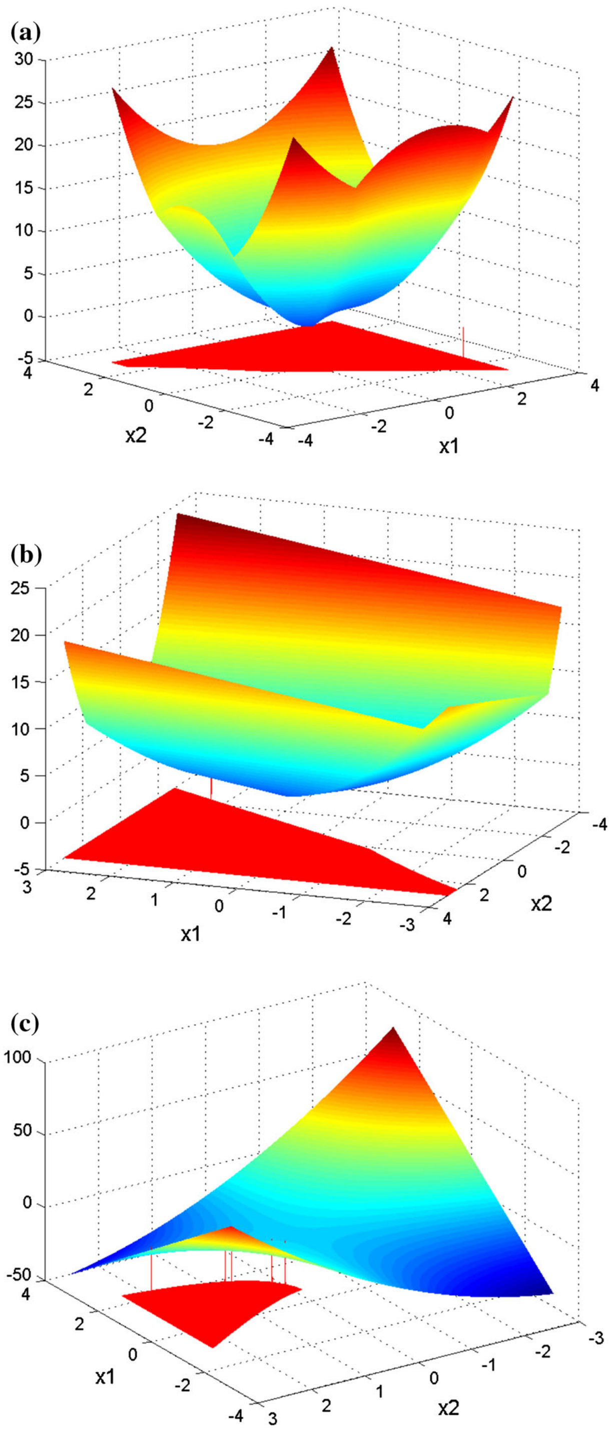

Fig. 1 The objective functions and the feasible domain in $\mathbb{R}^2$

6 Conclusions

In this paper, we provide an implementable algorithm for solving constraint nonconvex nonsmooth optimization problems. Our method just requires that the objective functions and the constraint functions are lower-C^2 functions, and this assumption covers a rich family of interesting problems.

With the help of penalty functions, we transform the constrained program into an unconstrained one, and we can make sure the sequence of penalty parameters eventually stabilizes by adding a constraint qualification to the problem. The rest of our work is to extend the local convexification technique in [11] to the penalty function, so make it possible to solve the constrained optimization.

All the three parameters in the algorithm can be chosen as any non-negative initial values, and updated by certain rules with the running of the program. If the convexification parameter equals to zero or to an arbitrary positive constant, our algorithm can be used to solve the convex programming.

From the obtained numerical results, we could see that there are 8 results below 0.05 in the all 10 deterministic tests, whose theoretical results are zero. As for stochastic tests, the results of our algorithm are very close to the public software SolvOpt in Group I. And in Group II, the numerical results are still acceptable when dimension is bigger than five. Our limited computational experiments suggest the good performance of the proposed method for a large class of problems.

Acknowledgments The authors are grateful to the anonymous referees for their helpful suggestions and comments. This work was partially supported by the Natural Science Foundation of China, Grant 11171049, 11301246.

7 Appendix A

Case 1 Convex constraint functions with $n = 3$.

$F_i(x) =< a^i, x > -b_i; i = 1, \dots, n$,

where $a^i_j = 1/(i + j), b_i = \sum_{j=1}^n a^i_j, c_i = -(b_i + 1/(1 + i))$. Since constraint functions are convex, the "augment" Slater constraint qualification here degenerates Slater constraint qualification, i.e. $F(\tilde{x}) < 0$.

Case 2 Nonconvex constraint functions with $n = 2$.

$$F_1 : A_1 = \begin{pmatrix} -1 & 0 \\ -2 & -1 \end{pmatrix};_{B_1=(-14,-18)^\top; C_1=-9.}$$

$$F_2 : A_2 = \begin{pmatrix} -1 & 0 \\ -1 & -1 \end{pmatrix};_{B_2=(-17,-12)^\top; C_2=-13.}$$

Case 3 Nonconvex constraint functions with $n = 3$.

$$F_1 : A_1 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{pmatrix};_{B_1=(-17,-13,-19)^\top; C_1=-35.}$$

$$F_2 : A_2 = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}; \quad B_2 = (-20, -13, -21)^\top; C_2 = -39.$$

$$F_3 : A_3 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}; \quad B_3 = (-21, -13, -18)^\top; C_3 = -33.$$

Case 4 Nonconvex constraint functions with $n = 4$.

$$F_1 : A_1 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{pmatrix}; \quad B_1 = (-27, -23, -21, -22)^\top; C_1 = -9.$$

$$F_2 : A_2 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}; \quad B_2 = (-28, -29, -21, -21)^\top; C_2 = -3.$$

$$F_3 : A_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad B_3 = (-27, -22, -21, -24)^\top; C_3 = -5.$$

$$F_4 : A_4 = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}; \quad B_4 = (-22, -23, -31, -22)^\top; C_4 = -3.$$

Case 5 Nonconvex constraint functions with $n = 5$.

$$F_1 : A_1 = \begin{pmatrix} -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 \end{pmatrix}; \quad B_1 = (-27, -33, -21, -32, -23)^\top; C_1 = -39.$$

$$F_2 : A_2 = \begin{pmatrix} -1 & 0 & 0 & -2 & 0 \\ 0 & -1 & -2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{pmatrix}; \quad B_2 = (-29, -52, -37, -12, -26)^\top; C_2 = -41.$$

$$F_3 : A_3 = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 0 & -1 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 & -1 \end{pmatrix}; \quad B_3 = (-17, -14, -41, -32, -21)^\top; C_3 = -35.$$

$$F_4 : A_4 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}; \quad B_4 = (-17, -13, -11, -12, -19)^\top; C_4 = -49.$$

$$F_5 : A_5 = \begin{pmatrix} -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 \end{pmatrix}; \quad B_5 = (-12, -24, -29, -41, -14)^\top; C_5 = -43.$$

8 Appendix B

Table 4 Results for dimension 3 ($f_2(x_0) = 4$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(−0.000466, 0.000000, 0.000000)	0.000466	19	−1.0	18
(0.000000, 0.000000, −0.000002)	0.000002	6	−0.0	76
(0.000000, 0.000000, 0.000000)	0.000000	1	−1.0	28
(0.001961, −0.000009, −0.002165)	0.004131	10	−5.0	109
(0.006498, 0.006519, 0.006541)	0.006604	9	−2.0	109

Table 5 Results for dimension 4 ($f_2(x_0) = 6$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(0.000001, 0.000000, −0.000002, −0.000000)	0.000003	0	−1.0	136
(0.041590, 0.042534, 0.043573, 0.044726)	0.090972	8	−3.6	63
(0.001316, 0.001317, 0.001318, 0.001319)	0.002640	11	−1.0	188
(0.005823, 0.005840, 0.005858, 0.005875)	0.011784	11	−0.9	214
(−0.000649, −0.000202, 0.000203, 0.000203)	0.000853	−13	−1.0	18

Table 6 Results for dimension 5 ($f_2(x_0) = 8$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(0.018181, 0.018352, 0.018531, 0.018717, 0.018907)	0.056662	23	−0.8	141
(0.005773, 0.005790, 0.005807, 0.005825, 0.005842)	0.017525	30	−0.0	93
(−0.179985, −0.167968, −0.158499, −0.150738, 0.429427)	0.164601	8	−1.5	124
(0.009040, 0.009082, 0.009124, 0.009167, 0.009211)	0.027627	4	−1.1	120
(0.000741, 0.000741, 0.000742, 0.000742, 0.000742)	0.002227	4	−3.0	128

Table 7 Results for dimension 6 ($f_2(x_0) = 10$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(0.081054, −0.066497, −0.064662, −0.063060, 0.202168, −0.060008)	0.171022	17	−1.4	148
(0.009449, 0.009495, 0.009541, 0.009589, 0.009637, 0.009686)	0.038588	42	−0.0	99
(0.000218, 0.000218, 0.000218, 0.000218, 0.000218, 0.000218)	0.000873	−1	−0.0	27
(0.000309, 0.000309, 0.000310, 0.000310, 0.000310, 0.000310)	0.001239	−6	−0.0	29
(−0.189414, 0.322598, −0.155142, 0.483417, −0.150346, 0.381744)	1.107563	9	−0.0	177

Table 8 Results for dimension 7 ($f_2(x_0) = 12$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(−0.000716, 0.000129, 0.000480, 0.000497, −0.000517, 0.000793, −0.000894)	0.001810	4	−5.0	231
(0.014970, 0.015085, 0.015204, 0.015328, 0.029868, 0.015587, 0.015724)	0.092051	24	−1.2	214
(0.005579, 0.009368, 0.061210, 0.098214, 0.005641, 0.005621, 0.005653)	0.180234	23	−0.0	182
(0.004374, 0.004383, 0.004393, 0.004403, 0.004413, 0.004423, 0.004433)	0.022093	36	−0.1	95
(0.280265, −0.117991, −0.112820, −0.108408, −0.104573, 0.193365, 0.111651)	0.405315	21	−0.5	235

Table 9 Results for dimension 8 ($f_2(x_0) = 14$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(−0.051379, −0.050180, −0.049084, −0.048075, −0.047142, −0.046274, 0.284927, −0.044704)	0.053487	55	−1.7	198
(−0.050161, −0.049046, −0.048001, −0.184760, 0.084701, 0.017983, −0.029931, −0.043876)	0.169580	19	−1.6	290
(−0.173786, −0.162485, −0.153530, 0.646161, −0.139933, −0.134564, −0.129862, −0.125692)	0.004082	27	−1.7	130
(0.094593, −0.115131, −0.111143, 0.278631, −0.102391, 0.099300, 0.142354, −0.093384)	0.253573	59	−0.9	219
(−0.018054, 0.038265, −0.017745, −0.017597, −0.017455, 0.073415, −0.017184, 0.015410)	0.075490	19	−0.7	190

Table 10 Results for dimension 9 ($f_2(x_0) = 16$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(0.000011, 0.000025, 0.000035, 0.000048, −0.000035, 0.000021, −0.000043, −0.000117, 0.000112)	0.000291	−8	−1.0	107
(−0.088182, −0.084867, −0.081987, 0.514852, −0.077178, −0.075134, −0.073276, −0.071577, −0.070012)	0.076778	64	−2.9	156
(−0.271062, 0.672139, 0.404954, −0.176239, −0.203842, 0.025276, −0.186269, −0.179242, −0.173045)	0.528267	60	−1.6	215
(0.182305, 0.264523, −0.074437, −0.0072297, −0.070370, −0.068631, 0.080603, −0.065568, −0.064204)	0.277453	32	−0.5	240
(0.005238, 0.005253, 0.035833, 0.005281, 0.005295, 0.032876, 0.005325, 0.005340, 0.005355)	0.095347	44	−0.2	205

Table 11 Results for dimension 10 ($f_2(x_0) = 18$)

x^*	$f_2(x^*)$	$F(x_0)$	$F(x^*)$	k
(−0.078884, −0.076191, −0.073825, −0.071712, −0.069815, 0.399531, −0.066513, −0.065066, 0.113528, −0.027453)	0.147601	57	−0.8	231
(−0.145549, −0.137291, 0.504132, 0.303905, −0.120092, −0.007609, −0.112145, −0.108793, −0.105781, −0.103047)	0.280055	35	−3.0	198
(−0.000002, −0.000029, 0.000033, −0.000002, −0.000002, −0.000016, −0.000001, 0.000020, −0.000002, −0.000002)	0.000070	29	−3.0	211
(0.815016, 0.599267, 0.543009, 0.280708, 0.351105, 0.473398, −0.140943, 0.386273, 0.354268, 0.325067)	4.408275	30	−0.1	231
(−0.132812, −0.125840, 0.780142, −0.115147, −0.071317, −0.107189, −0.103876, −0.100909, −0.098229, −0.095773)	0.112401	51	−1.4	175

References

1. Balinski, M.L., Wolfe, P.: Nondifferentiable Optimization. North-Holland, Amsterdam (1975)
2. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.: Numerical Optimization: Theoretical and Pratical Aspects, 2nd edn. Springer, Berlin (2000)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
4. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable minimization. Math. Program. **46**, 105–122 (1990)
5. Lemaréchal, C.: An extension of davidon methods to nondifferentiable problems. Nondiffer. Optim. Math. Program. Study **3**, 95–109 (1975)
6. Kiwiel, K.C.: A linearization algorithm for nonsmooth minimization. Math. Oper. Res. **10**(2), 185–194 (1985)
7. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. SIAM J. Optim. **14**, 743–756 (2004)
8. Haarala, N., Miettinen, K., Mäkelä, M.M.: Globally convergent limited memory bundle method for large-scale nonsmooth optimization. Math. Program. **109**, 181–205 (2007)
9. Mifflin, R.: A modification and an extension of Lemarechal's algorithm for nonsmooth minimization. Math. Program. Study **17**, 77–90 (1982)

10. Vlček, J., Lukšan, L.: Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. J. Optim. Theory Appl. **111**(2), 407–430 (2001)
11. Hare, W., Sagastizábal, C.: A redistributed proximal bundle method for nonconvex optimization. SIAM J. Optim. **20**, 2442–2473 (2010)
12. Hare, W., Sagastizábal, C.: Computing proximal points of nonconvex functions. Math. Program. **116**(1–2, Ser.B), 221–258 (2009)
13. Karmitsa, N., Mäkelä, M.M.: Adaptive limited memory bundle method for bound constrained large-scale nonsmooth optimization. Optimization **59**, 945–962 (2010)
14. Karmitsa, N., Mäkelä, M.M.: Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis. Optim. Methods Softw. **25**, 895–916 (2010)
15. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise and model and a bundling technique in nonconvex nonsmooth minimization. Optim. Methods Softw. **19**, 89–102 (2004)
16. Lukšan, L., Vlček, J.: A bundle-Newton method for nonsmooth unconstrained minimization. Math. Program. **83**, 373–391 (1998)
17. Schramm, H.: Zowe. J.: A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis. SIAM J Optim. **1**, 121–152 (1992)
18. Kiwiel, K.C.: Restricted step and Levenberg-Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization. SIAM J. Optim. **6**(1), 227–249 (1996)
19. Kiwiel, K.C.: Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. Math. Program. **52**, 285–302 (1991)
20. Mäkelä, M.M., Neittaanmäki, P.: Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control. Utopia Press, Singapore (1992)
21. Rockafellar, R.T., Wets, J.J.-B.: Variational Analysis. Springer, Berlin (1998)
22. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton, NJ (1970)
23. Cheney, E.W., Goldstein, A.A.: Newton's method for convex programming and tchebycheff approximation. Numerische Mathematik **1**, 253–268 (1959)
24. Kelley, J.E.: The cutting plane method for solving convex programs. SIAM J. Optim. **8**, 703–712 (1960)
25. Kuntsevich, A., Kappel, F.: SolvOpt—The Solver for Local Nonlinear Optimization Problems: Matlab, C and Fortran Source Codes. Institute for Mathematics, Karl-Franzens University of Graz (1997)