A PROXIMAL-PROJECTION PARTIAL BUNDLE METHOD FOR CONVEX CONSTRAINED MINIMAX PROBLEMS

CHUNMING TANG
College of Mathematics and Information Science, Guangxi University
Nanning 530004, China

JINBAO JIAN*
College of Science, Guangxi University of Nationalities
Nanning 530007, China
School of Mathematics and Statistics, Yulin Normal University
Yulin 537000, China

GUOYIN LI
Department of Applied Mathematics, University of New South Wales
Kensington, 2052, Sydney, Australia

(Communicated by Jinyan Fan)

ABSTRACT. In this paper, we propose a partial bundle method for a convex constrained minimax problem where the objective function is expressed as maximum of finitely many convex (not necessarily differentiable) functions. To avoid complete evaluation of all component functions of the objective, a partial cutting-planes model is adopted instead of the traditional one. Based on the proximal-projection idea, at each iteration, an unconstrained proximal subproblem is solved first to generate an aggregate linear model of the objective function, and then another subproblem based on this model is solved to obtain a trial point. Moreover, a new descent test criterion is proposed, which can not only simplify the presentation of the algorithm, but also improve the numerical performance significantly. An explicit upper bound for the number of bundle resets is also derived. Global convergence of the algorithm is established, and some preliminary numerical results show that our method is very encouraging.

1. Introduction. We consider the following convex constrained minimax problem

\[
\min \{ f(x) : x \in C \},
\]

where \( f(x) = \max \{ f_i(x), i \in I \} \) with \( I = \{1, \ldots, m\} \), and the component functions \( f_i (i \in I) : \mathbb{R}^n \to \mathbb{R} \) are convex but not necessarily differentiable, and \( C \) is a nonempty closed convex set in \( \mathbb{R}^n \).

Minimax problems are a typical and special class of nonsmooth optimization problems, which aims to make an “optimal” decision under the “worst” cost. They

2010 Mathematics Subject Classification. Primary: 90C47, 49K35.

Key words and phrases. Minimax problem, partial bundle method, proximal-projection, descent test criterion, global convergence.

* Corresponding author: Jinbao Jian. Project supported by the National Natural Science Foundation (11761013, 11771383) and Guangxi Natural Science Foundation (2013GXNSFAA019013, 2014GXNSFFA118001, 2016GXNSFDA380019) of China. This work was essentially done while the first author was visiting The University of New South Wales, Australia.
are widely used in practical applications, such as multi-facility location [28], portfolio selection [40], optimal control [4], energy consumption [1], waste management [24], etc. Due to the special structure of $f(x)$, general-purpose nonsmooth methods ([33, 22, 3]) may not be efficient to solve them. So, by exploiting their peculiar structure, a great deal of effort has been devoted to developing efficient methods for minimax problems, see, e.g., [37, 6, 31, 23, 41, 25, 38, 39, 9, 14, 15, 10, 11, 7, 12].

In particular, for the case where $f_i (i \in I)$ are continuously differentiable, one common approach for solving the original problem is the smoothing method. There are mainly two smoothing approaches: (i) by introducing a smoothing function, the original problem is approximated by a sequence of smooth optimization problems, and the solution can be obtained from the smoothing function, see e.g., [23, 41]; (ii) by introducing an auxiliary variable, the minimax problem can be equivalently reformulated as a smooth problem, and then smooth methods can be designed to solve this special smooth problem, such as penalty methods [6], trust region methods [37, 38, 39], and sequential (quadratically constrained) quadratic programming [31, 14, 15].

For the case where $f_i (i \in I)$ are not necessarily differentiable, or differentiable but their gradients are difficult to calculate, Gaudioso et al. [8] proposed an incremental bundle method for solving convex unconstrained minimax problems, which uses a partial cutting-planes model to avoid complete evaluation of the objective function $f$. Later, Gaudioso et al. [9] applied the method of [8] to solve the Lagrangian dual of integer programs, and Fuduli et al. [7] further extended it to solve convex minimax problems with infinite linear constraints. Jian et al. [16] proposed a feasible descent bundle method for inequality constrained minimax problems by using the partial cutting-planes idea of [8]. Liuzzi et al. [25] presented a derivative-free method for linearly constrained finite minimax problems by using a smoothing technique. Hare & Nutini [10] and Hare & Macklem [11] proposed a derivative-free gradient sampling method for solving unconstrained minimax problems.

We note that the above methods in the latter case (except for [16]) can only deal with unconstrained or linearly constrained problems. In this paper, we are interested in minimizing a max-function over a convex set with the form (1), and particularly concern that $f_i (i \in I)$ are not necessarily differentiable.

Our method falls within the well-known class of bundle methods for nonsmooth optimization. In particular, in order to reduce the number of component function evaluations, we adopt the partial cutting-planes model of [8] instead of the traditional one. As a result, at each iteration, only one component function is newly used to enrich the model. Moreover, in order to deal with the constraint set $C$ efficiently and make the proposed method more practical, we utilize the “proximal-projection” idea of [19, 20], which is different from the traditional projection technique. Therefore, at each iteration, an unconstrained proximal subproblem is solved first to generate an aggregate linear model of the objective function, and then another subproblem based on this model is solved to obtain a trial point. We note that solving the latter subproblem is equivalent to projecting a certain point onto the feasible set $C$, which can be easily calculated or even has a closed-form solution if $C$ has some special structure (e.g., boxes, simplices, balls, and special polyhedrons [5]).

Our algorithm produces a sequence of feasible trial points, and ensures that the objective function is monotonically decreasing on the sequence of stability centers. Global convergence of the algorithm is proved. Finally, the algorithm is improved
by the subgradient aggregation strategy, and some preliminary numerical results show that our method is quite efficient.

An important and interesting feature of this paper is that a new descent test criterion different from that of [8] is proposed. Our criterion does not contain a problem-data-independent parameter which may be not easy to choose numerically, and therefore the presentation of our algorithm and the theoretical analysis are simplified. Moreover, from the numerical comparisons reported in Section 5, our criterion performs better than that of [8]. That is to say, our method not only extends the method of [8] to constrained case, but also improves its descent test criterion significantly. Based on the new criterion, Tang et al. [34] presented an improved partial bundle method for solving linearly constrained minimax problems. In addition, we derive an explicit upper bound for the number of bundle resets.

Compared to [19, 20], we present a partial bundle variation to the proximal-projection bundle method for the purpose of reducing the number of component function evaluations. What is more, there is a major difference of theoretical analysis between our method and that of [19, 20], since except for the proximal-projection technique, our algorithmic framework is more close to that of [8]. We also note that our method is very different from that in [16] due to the proximal-projection strategy and the new descent test criterion. Some other recent methods for general-purpose nonsmooth constrained minimization problems can be found in, e.g., [13, 32, 17, 35, 36].

The paper is organized as follows. In Section 2, we present the details of our algorithm and discuss its properties. In Section 3, the global convergence is proved. Improvement of the algorithm by subgradient aggregation is given in Section 4. Preliminary numerical results and conclusion are given in Sections 5 and 6, respectively.

2. The algorithm. Let $k$ be the current iteration index, and $y^\ell, \ell \in J^k := \{1, \cdots, k\}$ be given points with subgradients $g^\ell \in \partial f(y^\ell)$. The classic bundle methods [2] uses the following cutting-planes model for $f$ at the $k$-th iteration:

$$f^k_{\text{CP}}(x) := \max_{\ell \in J^k} \{f(y^\ell) + \langle g^\ell, x - y^\ell \rangle\}.$$  

However, if the model $f^k_{\text{CP}}(x)$ is directly applied to the objective function (a max-function) of problem (1), then at each point $y^\ell$, we need to calculate the objective value $f(y^\ell)$ and a certain subgradient $g^\ell \in \partial f(y^\ell)$, which in turn implies that we have to evaluate the values of all the component functions $f_i (i \in I)$ at $y^\ell$. This is time-consuming, especially when $m$, the number of the component functions is large. Moreover, for some practical applications (see e.g., [9, 7]), the full knowledge of all component functions may be impossible.

In order to avoid complete evaluation of $f$, Gaudioso et al. [8] proposed that at each point $y^\ell$ just evaluate one of the component functions, say $f_{i_\ell}(y^\ell)$, for some $i_\ell \in I$, along with a corresponding subgradient $g^\ell_{i_\ell} \in \partial f_{i_\ell}(y^\ell)$, and then define the
partial cutting-planes (PCP) model:
\[ \bar{f}_{pcp}^k(x) := \max_{i \in J^k} \{ f_i(y^i) + \langle g_i^0, x - y^i \rangle \}, \]
which is also a lower polyhedral model for \( f \), i.e., \( \bar{f}_{pcp}^k(x) \leq f(x) \). This model has been proved to be very efficient in some cases, such as the Lagrangian dual of integer programs [9], and convex semi-infinite minimax problems [7].

In this section, we present a proximal-projection partial bundle method for solving problem (1), which well combines the PCP model (2) and the proximal-projection idea of [19, 20].

It is obvious that problem (1) is equivalent to the unconstrained problem
\[ \min f_C(x) := f(x) + \delta_C(x), \tag{3} \]
where \( \delta_C(x) \) is the indicator function of \( C \), i.e., \( \delta_C(x) = 0 \) if \( x \in C \) and \( \delta_C(x) = \infty \) otherwise. In what follows, we aim to design an algorithm based on the model problem (3). Similar to the traditional proximal bundle methods, we may consider the proximal partial bundle subproblem for problem (3) as follows
\[ \min \bar{f}_{pcp}^k(\cdot) + \delta_C(\cdot) + \frac{1}{2t_k} \| -x^k \|^2, \tag{4} \]
where \( t_k > 0 \) is the proximal parameter, and \( x^k \) (called stability center) is the “best” point obtained so far. However, the subproblem (4) is usually not easy to solve, so we “decompose” it into two subproblems. One is an unconstrained proximal subproblem which is much easier to solve than the original problem and used to generate an aggregate linear model of \( f \). The other subproblem based on such a linear model is solved to obtain a new trial point. Note that solving the second subproblem is equivalent to projecting a certain point onto the feasible set \( C \).

Before giving the algorithm, we first provide a brief description of the sequences generated by the following algorithm:

\{-z^k\} — the sequence of proximal points, at which the aggregate linear models of \( f \) are generated;
\{-y^k\} — the sequence of trial points. If \( f \) achieves a sufficient decrease at a trial point, then declare a descent step, and accept it as a stability center;
\{-x^k\} — the sequence of stability centers. We emphasize that we only require a full knowledge of \( f \) at the stability centers. This seems to be the minimum requirement for global convergence. Moreover, from the mechanism of our algorithm, there is no extra computational cost. In fact, once a new stability center is generated, the values of all the component functions are already known. Moreover, from our numerical experience, the number of stability centers is generally a very small proportion of the whole iterations.

Algorithm 2.1. (Proximal-projection partial bundle method)

**Step 0. Initialization.** Select \( y^1, x_0 \in C \), set \( x^1 = y^1 \), \( J^1 = \{1\} \), the initial bundle \( B_1 = \{(y^1, f(y^1), g^1)\} \) with \( g^1 \in \partial f(y^1) \). Set \( \delta_C^0(\cdot) := \langle p_C^0, \cdot \rangle \) with \( p_C^0 := 0 \).

Select \( \epsilon > 0, \eta > 0, \sigma > 1, t_0 > 0, \alpha \in (0, 1) \). Set \( t_1 = t, k := 1 \).

**Step 1. Proximal point finding.** Set
\[ z^{k+1} := \arg \min \left\{ \phi_{\delta_C^k}(\cdot) := \bar{f}_{pcp}^k(\cdot) + \delta_C^{k-1}(\cdot) + \frac{1}{2t_k} \| -x^k \|^2 \right\}, \tag{5} \]
\[ \bar{f}_{pcp}^k(\cdot) := \bar{f}_{pcp}^k(z^{k+1}) + \langle p_C^k, \cdot - z^{k+1} \rangle \] with \( p_C^k := -p_C^{k-1} - \frac{1}{t_k}(z^{k+1} - x^k) \). \tag{6}
Step 2. Projection. Set

\[ y^{k+1} := \arg \min \left\{ \phi_C^k(\cdot) := \tilde{f}^k(\cdot) + \delta_C(\cdot) + \frac{1}{2t_k} \| \cdot - x^k \|^2 \right\}, \quad (7) \]

\[ \tilde{\delta}_C^k(\cdot) := \langle p_C^k, - y^{k+1} \rangle \text{ with } p_C^k := -p_f - \frac{1}{t_k} (y^{k+1} - x^k). \quad (8) \]

Step 3. Stopping criterion. Compute

\[ p^k := \bar{p}_f^k + p_C^k = - \frac{1}{t_k} (y^{k+1} - x^k), \quad (9) \]

\[ v_k := \tilde{f}^k(y^{k+1}) - f(x^k), \quad (10) \]

\[ \epsilon_k := -v_k - t_k \| p^k \|^2. \quad (11) \]

If \( \| p^k \| \leq \eta \) and \( \epsilon_k \leq \epsilon \), STOP. Else if \( \| p^k \| > \eta \), go to Step 5. Else \( \| p^k \| \leq \eta \) and \( \epsilon_k > \epsilon \), go to Step 4.

Step 4. Bundle reset. Set \( x^{k+1} := x^k \) (null step), \( y^{k+1} := x^k \), \( J^{k+1} = \{ k + 1 \} \), and make a bundle reset

\[ B_{k+1} = \{ (y^{k+1}, f(y^{k+1}), g^{k+1}) \} \text{ with } g^{k+1} \in \partial f(y^{k+1}). \]

Set \( p_C^k := 0 \), \( \delta_C^j(\cdot) := 0 \), \( t_{k+1} := t_k / \sigma, k := k + 1 \), and go to Step 1.

Step 5. Descent test. Extract any index \( i \) from the index set \( I \). If

\[ f_i(y^{k+1}) - f(x^k) > \alpha v_k, \quad (12) \]

set \( x^{k+1} := x^k \) (null step), \( J^{k+1} = J^k \cup \{ k + 1 \} \), and update

\[ B_{k+1} = B_k \cup \{ (y^{k+1}, f_i(y^{k+1}), g_i^{k+1}) \} \text{ with } g_i^{k+1} \in \partial f_i(y^{k+1}). \quad (13) \]

Restore the index set by setting \( I = \{ 1, \ldots, m \} \), let \( t_{k+1} := t_k, k := k + 1 \), and return to Step 1.

Step 6. Index removing. Set \( I := I \setminus \{ i \} \). If \( I \neq \emptyset \), then return to Step 5.

Step 7. Updating. Set \( x^{k+1} := y^{k+1} \) (descent step), \( J^{k+1} = J^k \cup \{ k + 1 \} \), update the bundle by

\[ B_{k+1} = B_k \cup \{ (y^{k+1}, f(y^{k+1}), g^{k+1}) \} \text{ with } g^{k+1} \in \partial f(y^{k+1}). \quad (14) \]

Restore the index set by setting \( I = \{ 1, \ldots, m \} \), let \( t_{k+1} := t_i, k := k + 1 \), and return to Step 1.

A few comments on the algorithm are in order.

Remark 1.  
- (On Step 0) The algorithm needs to calculate the function value of \( f \) at the initial stability center \( x^1 \) (see (12)), so in Step 0 we simply use \( (y^1, f(y^1), g^1) \) to initialize the bundle. In addition, \( \delta_C(\cdot) \) is a linearization of \( \delta_C(\cdot) \) at \( y^1 \), since \( \delta_C(y^1) = 0 \) and \( p_C^0 = 0 \in \partial \delta_C(y^1) \).
- (On Step 1) In subproblem (5), \( f^k_{\text{pep}} \) and \( \delta_C^k \) approximate \( f \) and \( \delta_C \), respectively; the third term is a quadratic proximal term. This subproblem can be cast as a simple convex quadratic programming (QP), and therefore can be solved efficiently. By solving (5), a proximal point \( z^{k+1} \) is first found, and then a linearization \( \tilde{f}^k(\cdot) \) of \( f^k_{\text{pep}}(\cdot) \) at \( z^{k+1} \) is built due to the fact that \( p_f^k \in \partial f^k_{\text{pep}}(z^{k+1}) \). Moreover, there exist multipliers \( \lambda^k_\ell, \ell \in J^k \) such that

\[
\begin{align*}
    p_f^k &= \sum_{\ell \in J^k} \lambda^k_\ell g^k_\ell, \\
    \sum_{\ell \in J^k} \lambda^k_\ell &= 1, \\
    \lambda^k_\ell &\geq 0, \\
    \lambda^k_\ell [f^k_{\text{pep}}(z^{k+1}) - f^k_\ell(y^k) - (g^k_\ell, z^{k+1} - y^k)] &= 0, \quad \ell \in J^k.
\end{align*}
\]
which implies
\[
\hat{f}^k(\cdot) = \sum_{\ell \in I^k} \lambda^k_{\ell}[f_{\ell}(y^\ell) + \langle g_{\ell}, \cdot - y^\ell \rangle].
\]  
(16)

This shows that the linear function \(\hat{f}^k(\cdot)\) is an aggregation (convex combination) of the component functions of \(\tilde{f}^k_{\text{pcp}}(\cdot)\).

- **(On Step 2)** The solution of (7) can be obtained by the projection: \(y^{k+1} = P_C(x^k - t_k p^k)\), which is straightforward if \(C\) is simple. Moreover, \(\delta^k_C(\cdot)\) is a linearization of \(\delta_C\) at \(y^{k+1}\), since \(p^k \in \partial\delta_C(y^{k+1})\) and \(y^{k+1} \in C\).

- **(On Step 3)** Three cases are considered in Step 3. (a) If both \(\|p^k\|\) and \(\epsilon_k\) are “sufficiently small”, then the algorithm stops, and we accept the current stability center \(x^k\) as an approximate optimal solution. (b) If \(\|p^k\|\) is not small, then we accept \(y^{k+1}\) as a new trial point at which either sufficient decrease is achieved (descent step) or a new cutting plane is built to improve the model (null step). (c) The case “\(\|p^k\|\) is small but \(\epsilon_k\) is large” implies that the partial cutting-planes model is “bad”, so the algorithm makes a bundle reset, i.e., it restarts from the current stability center.

- **(On Step 4)** In the case of bundle reset, the model is not reliable enough, so we should become more “conservative”, and therefore decrease the proximal parameter by \(t_{k+1} = t_k / \sigma\) (see [8]). Although we need to know \(f\) and a subgradient at the current stability center, there is no extra computational cost, since they have been generated in the previous iterations.

- **(On Step 5)** The descent test criterion (12) is newly introduced in this paper, which has obvious advantages when compared to that of [8]. In fact, Gaudioso et al. [8] (essentially) used the following criterion

\[
f_i(y^{k+1}) - f(x^k) > v_k + \theta_k,
\]

(17)

where \(\theta_k > 0\) is a problem-data-independent parameter satisfying \(\theta_k / t_k < \eta^2\). However, as a stopping tolerance, \(\eta\) should be a suitably small positive constant (say \(\eta = 10^{-3}\) in our numerical tests), and then \(\eta^2\) becomes a much more smaller constant (say \(\eta = 10^{-6}\)). As a result, the numerical choice of the parameters \(\theta_k\) and \(t_k\) becomes very difficult, and therefore it is usually hard to obtain satisfactory numerical results.

In contrast, without the parameter \(\theta_k\) and by adding a coefficient \(\alpha \in (0, 1)\) to \(v_k\), our criterion (12) becomes more flexible and practical, since \(v_k\) itself represents a predicted decrease. Moreover, the removal of parameter \(\theta_k\) can simplify the presentation of the algorithm and theoretical analysis. From our numerical experience, the criterion (12) performs obviously better than (17). We note that the constant \(\alpha\) in (12) plays an important role not only in theoretical analysis but also in improving numerical results.

In Step 5, the selection strategy of index \(i \in I\) does not affect the theoretical analysis, but suitable rules are expected to be useful in practical performance, such as fixed sorted selection, randomized selection, and selection by exploiting knowledge of the active constraints from (15), see, e.g., [29]. If an index \(i\) is found such that (12) holds, then a new cutting plane built on the component function \(f_i\) at \(y^{k+1}\) is added, which can improve the model significantly [8, Lemma 2.4]. Otherwise, if no such index exists, the algorithm enters Step 7.
3. **Global convergence.** In this section, we prove the global convergence of Algorithm 2.1. We first make the following basic assumption.

**Assumption 1.** The level set \( \mathcal{L} := \{ x \in C : f(x) \leq f(x^1) \} \) for problem (1) is bounded.

Proof. The proof is the same as Lemma 3 of [34].
Between two consecutive descent steps, Algorithm 2.1 must take only one of the following two cases: (i) The algorithm loops between Steps 1 and 6, the stability center remains unchanged, and only null steps are made; (ii) The algorithm enters Step 7, the stability center is updated, and a descent step is generated.

In what follows, we first show that Algorithm 2.1 is well defined, i.e., the number of loops between Step 1 and Step 6 is finite, and therefore Algorithm 2.1 must enter Step 7 after a finite number of iterations. In particular, we will show that the following claims hold if the stability center remains unchanged:

(a) The algorithm get through Step 4 finitely many times;
(b) The algorithm get through Step 6 finitely many times;
(c) The number of loops between Step 1 and Step 5 is finite.

Before proving (a), we give some important relations about the optimal values of subproblems (5) and (7).

**Lemma 3.1.** Consider the following two subproblems:

\[
\min \phi_f^k(\cdot) := \tilde{f}_f^k(\cdot) + \delta_{f}^{k-1}(\cdot) + \frac{1}{2t_k} \| \cdot - x^k \|^2, \tag{22}
\]

\[
\min \phi_C^k(\cdot) := \tilde{f}_C^k(\cdot) + \delta_{C}^k(\cdot) + \frac{1}{2t_k} \| \cdot - x^k \|^2. \tag{23}
\]

Then it follows that \( z^{k+1} \) and \( y^{k+1} \) are solutions of subproblems (22) and (23), respectively. Furthermore, the following relations hold:

\[
\phi_f^k(z^{k+1}) + \frac{1}{2t_k} \| z^{k+1} - x^k \|^2 = \tilde{\phi}_f^k(x^k) \leq f(x^k), \tag{26}
\]

\[
\phi_C^k(y^{k+1}) + \frac{1}{2t_k} \| y^{k+1} - x^k \|^2 = \tilde{\phi}_C^k(x^k) \leq f(x^k). \tag{27}
\]

**Proof.** From (6), we have

\[
\nabla \tilde{\phi}_f^k(z^{k+1}) = p^k_f + p_{C}^{k-1} + \frac{1}{t_k} (z^{k+1} - x^k) = 0,
\]

which implies that \( z^{k+1} \) is the solution of (22). Similarly, it follows from (8) that \( \nabla \tilde{\phi}_C^k(y^{k+1}) = 0 \), so \( y^{k+1} \) is the solution of (23).

Since \( \tilde{f}_f^k(z^{k+1}) = \tilde{f}_C^k(z^{k+1}) \), we have \( \phi_f^k(z^{k+1}) = \phi_C^k(z^{k+1}) \). So by Taylor’s expansion and the fact that \( \tilde{\phi}_f^k \) is quadratic, we have (24). Equality (25) holds similarly from \( \tilde{\phi}_C^k(y^{k+1}) = \phi_C^k(y^{k+1}) \).

Relations (26) and (27) hold easily from (24) and (25), respectively. \( \square \)

**Lemma 3.2.** If a null step takes place from \( k \) to \( k + 1 \) without bundle reset, then

\[
\phi_f^k(z^{k+1}) + \frac{1}{2t_k} \| z^{k+1} - z^{k+1} \|^2 = \tilde{\phi}_f^k(y^{k+1}) \leq \phi_C^k(y^{k+1}), \tag{28}
\]

\[
\phi_C^k(y^{k+1}) + \frac{1}{2t_k} \| y^{k+1} - y^{k+1} \|^2 = \tilde{\phi}_C^k(z^{k+2}) \leq \phi_f^k(z^{k+2}), \tag{29}
\]

\[
\phi_f^k(z^{k+1}) \leq \phi_C^k(y^{k+1}) \leq \phi_f^k(z^{k+2}) \leq \phi_C^k(y^{k+2}). \tag{30}
\]
Lemma 3.3. Suppose that Algorithm 2.1 reaches a certain stability center $x^\bar{k}$ for some iterative index $\bar{k}$, and it remains unchanged after that, i.e., Step 7 is omitted and only null steps are made. Then Algorithm 2.1 get through Step 4 finitely many times. In particular, let $N_{\bar{k}}$ be the number of times get through Step 4, then there exists a positive constant $L$ such that

$$N_{\bar{k}} \leq \left\lfloor \frac{\ln 3\sqrt{2A^2\varepsilon}}{\ln \sigma} \right\rfloor,$$

where $[A]$ is the smallest integer greater than or equal to $A$.

Proof. We first note that $x^k = x^\bar{k}$, for all $k \geq \bar{k}$. Denote $k_r (\geq \bar{k})$ the index corresponding to the $r$-th time the algorithm enters Step 4. After entering Step 4, the bundle is reset by $B_{k_r+1} = \{(x^k, f(x^\bar{k}), g^\bar{k})\}$ with $g^\bar{k} \in \partial f(x^\bar{k})$, and the parameter is updated by

$$t_{k_r+1} = \bar{t}/\sigma^r.$$

The model is updated as

$$\tilde{f}_{\text{p}}^{k_r+1}(\cdot) = f(x^\bar{k}) + \langle g^\bar{k}, \cdot - x^\bar{k} \rangle, \quad \tilde{\delta}_C^{k_r}(\cdot) = 0.$$  \hfill (33)

Replacing $k$ by $k_r + 1$ in (26), we have

$$\phi_f^{k_r+1}(z^{k_r+2}) + \frac{1}{2t_{k_r+1}} \|z^{k_r+2} - x^\bar{k}\|^2 \leq f(x^\bar{k}).$$  \hfill (34)

On the other hand, it follows from (33) that

$$\phi_f^{k_r+1}(z^{k_r+2}) = \tilde{f}_{\text{p}}^{k_r+1}(z^{k_r+2}) + \tilde{\delta}_C^{k_r}(z^{k_r+2}) + \frac{1}{2t_{k_r+1}} \|z^{k_r+2} - x^\bar{k}\|^2$$

$$= f(x^\bar{k}) + \langle g^\bar{k}, z^{k_r+2} - x^\bar{k} \rangle + \frac{1}{2t_{k_r+1}} \|z^{k_r+2} - x^\bar{k}\|^2.$$  \hfill (35)

Combining (34) and (35), we have

$$\frac{1}{t_{k_r+1}} \|z^{k_r+2} - x^\bar{k}\|^2 \leq -\langle g^\bar{k}, z^{k_r+2} - x^\bar{k} \rangle \leq \|g^\bar{k}\| \|z^{k_r+2} - x^\bar{k}\|,$$

which along with (32) implies

$$\|z^{k_r+2} - x^\bar{k}\| \leq t_{k_r+1} \|g^\bar{k}\| = \bar{t} \|g^\bar{k}\| / \sigma^r.$$  \hfill (36)

For $k_r + 1 \leq k \leq k_{r+1}$, from (24), (26) and (30), we have

$$\|z^{k+1} - x^\bar{k}\|^2 = 2t_k [\phi_f^{k+1}(x^\bar{k}) - \phi_f^k(z^{k+1})] \leq 2t_k [f(x^\bar{k}) - \phi_f^{k+1}(z^{k+2})].$$  \hfill (37)

This together with (35) and (36) shows that

$$\|z^{k+1} - x^\bar{k}\|^2 = 2t_k [f(x^\bar{k}) - f(x^\bar{k}) + \langle g^\bar{k}, z^{k_r+2} - x^\bar{k} \rangle + \frac{1}{2t_{k_r+1}} \|z^{k_r+2} - x^\bar{k}\|^2]$$

$$\leq -2t_k \langle g^\bar{k}, z^{k_r+2} - x^\bar{k} \rangle \leq 2t_k \|g^\bar{k}\| \|z^{k_r+2} - x^\bar{k}\| \leq 2(\tilde{t}/\sigma^r)^2.$$  \hfill (38)
Similarly, from (25), (27) and (30), we have
\[
\|y^{k+1} - x^k\|^2 = 2t_k[f_C(x^k) - \phi_C(y^{k+1})] \leq 2t_k[f_C(x^k) - \phi_f(z^{k+1})] \\
\leq 2t_k[f(x^k) - \phi_f(z^{k+1} - z^{k+2})] \leq 2(\frac{\|x^k\|^2}{\sigma}),
\]
where the last inequality holds equally from (38).
Thus, for \(k_r + 1 \leq k \leq k_{r+1}\), from (33), we obtain
\[
\hat{f}_{pcp}^k(\cdot) \geq f(x^k) + \langle g^k, \cdot - x^k \rangle,
\]
which together with (6), (38) and (39) shows that
\[
v_k = \hat{f}_{pcp}^k(z^{k+1}) + (p_f, y^{k+1} - z^{k+1}) - f(x^k) \\
\geq f(x^k) + (g^k, z^{k+1} - x^k) + (p_f, y^{k+1} - x^k) - f(x^k) \\
= (g^k, z^{k+1} - x^k) + (p_f, y^{k+1} - x^k) + (p_f, x^k - z^{k+1}) \\
\geq -\|g^k\|\|z^{k+1} - x^k\| - \|p_f\|\|y^{k+1} - x^k\| - \|p_f\|\|z^{k+1} - x^k\| \\
\geq -\sqrt{2t_l}\|g^k\|^2 - \sqrt{2t_l}\|g^k\|^2 - \sqrt{2t_l}\|g^k\|^2.
\]

On the other hand, from Assumption 1, (15) and (39), there exists a positive constant \(L\) such that \(\|g^k\| \leq L\) and \(\|p_f\| \leq L\). This together with (40) shows that
\[
v_k \geq -\frac{\sqrt{2t_l}L^2}{\sigma},
\]
which along with (11) in turn implies
\[
\epsilon_k \leq -\epsilon_k - 3\sqrt{2t_l}L^2, \quad \forall k_r + 1 \leq k \leq k_{r+1}.
\]
Thus, if \(r \geq \frac{\ln 2}{\ln \sigma}\), then \(\epsilon_k \leq c\). This means that \(N_k\) satisfies (31). \(\square\)

Remark 2. In the preceding lemma, the constant \(L\) can be chosen as the common Lipschitz modulus of \(f_i\) over the bounded level set \(L\).

We proceed to prove (b) and (c). From Lemma 3.3 we may assume that the bundle reset does not occur.

Lemma 3.4. Suppose that Algorithm 2.1 reaches a certain stability center \(x^k\), bundle reset does not occur, and infinitely many null steps are made. Then \(\limsup_{k \to \infty} \epsilon_k \leq 0\), where \(\epsilon_k := f_{i_{k+1}}(y^{k+1}) - \bar{f}_k(y^{k+1})\), and \(i_{k+1}\) is the index generated in Step 5 satisfying \(f_{i_{k+1}}(y^{k+1}) - f(x^k) > \alpha v_k\).

Proof. First, from the statement of the lemma, it follows that
\[
x^k \equiv x^k, \quad t_k \equiv t_k, \quad \forall k \geq \tilde{k},
\]
Let \(g_{i_{k+1}}^{k+1} \in \partial f_{i_{k+1}}(y^{k+1})\) be generated in Step 5. Then from (13), we have
\[
\hat{f}_{pcp}^{k+1}(\cdot) \geq f_{i_{k+1}}(y^{k+1}) + \langle g_{i_{k+1}}^{k+1}, \cdot - y^{k+1} \rangle.
\]
This together with (41) and the objective functions of (5) and (7) shows that
\[
\epsilon_k = f_{i_{k+1}}(y^{k+1}) + \langle g_{i_{k+1}}^{k+1}, z^{k+2} - y^{k+1} \rangle + \langle g_{i_{k+1}}^{k+1}, y^{k+1} - z^{k+2} \rangle - \bar{f}_k(y^{k+1}) \\
\leq \hat{f}_{pcp}^{k+1}(z^{k+2}) - \bar{f}_k(y^{k+1}) + \|g_{i_{k+1}}^{k+1}\|\|y^{k+1} - z^{k+2}\| \\
= [\phi_{C}^{k+1}(z^{k+2}) - \bar{f}_k(y^{k+1}) - \frac{1}{T_k}\|z^{k+2} - x^k\|^2] \\
- [\phi_{C}(y^{k+1}) - \frac{1}{T_k}\|y^{k+1} - x^k\|^2] + \|g_{i_{k+1}}^{k+1}\|\|y^{k+1} - z^{k+2}\| \\
= \phi_{C}(y^{k+1}) - \phi_{C}(y^{k+1}) + \frac{1}{T_k}\|y^{k+1} - x^k\|^2 - \|z^{k+2} - x^k\|^2 \\
- \bar{f}_k(y^{k+1}) + \|g_{i_{k+1}}^{k+1}\|\|y^{k+1} - z^{k+2}\|.\]

(42)
From (26), (27) and (30), the sequences \( \{ \phi_k(z^{k+1}) \}_{k \geq \bar{k}} \) and \( \{ \phi_C(y^{k+1}) \}_{k \geq \bar{k}} \) are nondecreasing and bounded above by \( f(x^\bar{k}) \), so they are both convergent. By using (30) again, their limits are equal, i.e.,

\[
\lim_{k \to \infty} \phi_k(z^{k+1}) = \lim_{k \to \infty} \phi_C(y^{k+1}),
\]

which together with (29) and (41) shows that

\[
\lim_{k \to \infty} \| z^{k+2} - y^{k+1} \| = 0.
\]

Moreover, (27) implies \( \{ y^{k+1} \} \) is bounded, so from (8) and (15) we have

the sequences \( \{ y_{ik+1}^{k+1} \} \) and \( \{ p^k_C \} \) are bounded.

Since

\[
\| z^{k+2} - x^\bar{k} \|^2 = \| y^{k+1} - x^\bar{k} \|^2 + 2 \langle z^{k+2} - y^{k+1}, y^{k+1} - x^\bar{k} \rangle + \| z^{k+2} - y^{k+1} \|^2,
\]

we have from (44) that

\[
\lim_{k \to \infty} \frac{1}{2v_k} (\| y^{k+1} - x^\bar{k} \|^2 - \| z^{k+2} - x^\bar{k} \|^2) = 0.
\]

On the other hand, from (8), (44) and (45), it follows

\[
\lim_{k \to \infty} \bar{\delta}_k(z^{k+2}) = 0.
\]

Summarizing (43), (46) and (47), we know that the right-hand side of (42) converges to zero, so the lemma holds.

**Lemma 3.5.** Suppose that Algorithm 2.1 reaches a certain stability center \( x^\bar{k} \) which remains unchanged. Then Algorithm 2.1 passes finitely many times through Step 6, and the number of loops between Step 1 and Step 5 is finite.

**Proof.** Suppose by contradiction that one of the following cases occurs: (i) infinitely many times through Step 6; (ii) the loop between Step 1 and Step 5 is infinite. In both cases, it follows that

\[
\| p^k \| > \eta, \quad k \geq \bar{k},
\]

but at the same time the descent property \( f(y^{k+1}) - f(x^\bar{k}) \leq \alpha v_k \), \( k \geq \bar{k} \) is not satisfied. Thus, for each \( k \geq \bar{k} \), there exists an index \( i_{k+1} \) generated in Step 5 such that \( f_{i_{k+1}}(y^{k+1}) - f(x^\bar{k}) > \alpha v_k \). This together with (10) implies

\[
\epsilon_k = f_{i_{k+1}}(y^{k+1}) - f_k^{k_{y^{k+1}}} > \alpha v_k + f(x^\bar{k}) - f_k^{y^{k+1}} = (\alpha - 1)v_k \geq 0.
\]

So from Lemma 3.4, we have \( v_k \to 0 \), which contradicts (48).

From Lemmas 3.3 and 3.5, the following theorem holds immediately.

**Theorem 3.6.** Algorithm 2.1 is well defined, i.e., the stability center must be updated after a finite number of iterations.

Now we present the global convergence result of Algorithm 2.1.

**Theorem 3.7.** For any \( \epsilon > 0 \) and \( \eta > 0 \), Algorithm 2.1 stops after a finite number of iterations at a point satisfying the approximate optimality condition

\[
\| p^{k^*} \| \leq \eta \quad \text{and} \quad \epsilon_{k^*} \leq \epsilon,
\]

where \( k^* \) is the index for the last iteration. Furthermore, \( x^{k^*} \in C \) can serve as an approximately optimal solution of problems (1).

**Proof.** The proof is similar to that of Theorem 2 in [34].
4. Improvement by subgradient aggregation. In this section, similar to [34], we will improve our Algorithm 2.1 by introducing the subgradient aggregation (SA) strategy of [21]. As iterations go along, the number of elements in the bundle may increase infinitely. This could present serious problems with storage and computation after a large number of iterations, so we need to control the size of the bundle and meanwhile keep the theoretical convergence. The subgradient aggregation strategy [21] is the most popular and efficient way to do that. Hence, we utilize it to improve Algorithm 2.1 such that the storage requirements are bounded. To save space, we omit the same steps as in Algorithm 2.1.

Algorithm 4.1. (Proximal-projection partial bundle method with SA)

Step 0. Initialization. Select $y^1 \in C$, set $x^1 = y^1$, $J^1 = \{1\}$, the initial bundle $\mathcal{B}_1 = \{(y^1, f(y^1), g^1)\}$, with $g^1 \in \partial f(y^1)$. Set $f^0(\cdot) := f(y^1) + \langle g^1, \cdot - y^1 \rangle$, and $\delta^0_C(\cdot) := (p_C^0, \cdot - y^1)$ with $p_C^0 := 0$. Select $\epsilon > 0$, $\eta > 0$, $\sigma > 1$, $\bar{t} > 0$, $\alpha \in (0, 1)$. Set $t_1 = \bar{t}$, $BR = \emptyset$, $k := 1$.

Step 1. Proximal point finding. Set

$$f^{k}_{sa}(\cdot) = \max \{\tilde{f}^{k-1}_{sa}(\cdot), \tilde{f}^{k}_{pcp}(\cdot)\},$$

(49)

$$z^{k+1} := \arg \min \left\{ \phi^k(\cdot) := f^{k}_{sa}(\cdot) + \delta^{k-1}_C(\cdot) + \frac{1}{2t_k} \| \cdot - x^k \|^2 \right\},$$

(50)

$$\tilde{f}^k(\cdot) := f^{k}_{sa}(z^{k+1}) + \langle p^k, \cdot - z^{k+1} \rangle$$

with $p^k := -p^{k-1}_C - \frac{1}{t_k}(z^{k+1} - x^k)$.

Step 4. Bundle reset. Set $x^{k+1} := x^k$ (null step), $y^{k+1} := x^k$, $J^{k+1} = \{k + 1\}$, $BR = \{k + 1\}$, and make a bundle reset

$$\mathcal{B}_{k+1} = \{(y^{k+1}, f(y^{k+1}), g^{k+1})\}$$

with $g^{k+1} \in \partial f(y^{k+1})$. Set $\tilde{f}^k(\cdot) := f(y^{k+1}) + \langle g^{k+1}, \cdot - y^{k+1} \rangle$, $p^k_C := 0$, $t_{k+1} := t_k/\sigma$, $k := k + 1$, and return to Step 1.

Step 5. Descent test. Extract any index $i$ from the index set $I$. If

$$f_i(y^{k+1}) - f(x^k) > \alpha v_k,$$

(51)

then set $x^{k+1} := x^k$ (null step). Select subset $\tilde{J}^k$ satisfying $BR \subseteq \tilde{J}^k \subseteq J^k$ and generate its corresponding bundle $\tilde{\mathcal{B}}_k \subseteq \mathcal{B}_k$, set $J^{k+1} = \tilde{J}^k \cup \{k + 1\}$, and update the bundle by

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k \cup \{(y^{k+1}, f_i(y^{k+1}), g_i^{k+1})\}$$

with $g_i^{k+1} \in \partial f_i(y^{k+1})$. Restore the index set by setting $I = \{1, \ldots, m\}$, let $t_{k+1} := t_k$, $k := k + 1$, and return to Step 1.

Step 6. Updating. Set $x^{k+1} := y^{k+1}$ (descent step). Select $\tilde{J}^k \subseteq J^k$ and its corresponding bundle $\tilde{\mathcal{B}}_k \subseteq \mathcal{B}_k$, set $J^{k+1} = \tilde{J}^k \cup \{k + 1\}$, and update

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k \cup \{(y^{k+1}, f(y^{k+1}), g^{k+1})\}$$

with $g^{k+1} \in \partial f(y^{k+1})$. Restore the index set by setting $I = \{1, \ldots, m\}$, let $t_{k+1} := \bar{t}$, $BR = \emptyset$, $k := k + 1$, and return to Step 1.

Remark 3. (i) From (15), we know that $\tilde{f}^{k-1}$ is an aggregation (convex combination) of the past linearizations, which is now treated as a usual linearization in (49). $\tilde{f}^{k}_{pcp}$ at least contains the linearization at the current trial point $y^k$, and in the case of bundle reset, such linearization is $f^k := f(y^k) + \langle g^k, \cdot - y^k \rangle$; otherwise,
it is \( f^k := f_{i_k}(y^k) + \langle g^k_{i_k}, \cdot \rangle - y^k \) for some \( i_k \in I \) with \( g^k_{i_k} \in \partial f_{i_k}(y^k) \). Therefore, it follows that the model \( \tilde{f}_{sa} \) satisfies

\[
\max \{ f^{k-1}, f^k \} \leq \tilde{f}_{sa} \leq f^k.
\]  

(ii) The set \( BR \) is used to record the first index after a bundle reset, since the corresponding element should be included in the new bundle of Step 5 such that Lemma 3.3 (especially (40)) holds. This together with (52) guarantees that, after suitable modifications, the convergence analysis for Algorithm 2.1 also applies to Algorithm 4.1. Some similar discussions can be found in [20].

(iii) The choice of subset \( J_k \) is very flexible, since theoretically speaking, it can be a singleton or even an empty set. In practice, a popular way is to delete some elements from the bundle when its size reaches a preset maximum value, see [2, Ch. 10] for more details.

5. Numerical results. This section aims to test the practical effectiveness of Algorithm 4.1. We tested a set of 14 convex minimax problems. The first set of 13 problems are generalized from the unconstrained versions in [26] (a widely used collection of nonsmooth test problems) by imposing suitable constraints. In particular, two classes of constraints are tested here:

(i) box constraints: \( C = \{ x : l \leq x \leq u \} \) with \( l, u \in \mathbb{R}^n \);

(ii) ball constraints: \( C = \{ x : \| x - a \| \leq b \} \) with \( a \in \mathbb{R}^n, 0 < b \in \mathbb{R} \).

We add a superscript "□" to the problem name in Table 1 to indicate that the problem is subjected to a box, and a superscript "o" in Table 2 to indicate a ball constraint. The 14th problem is taken from [27] (Example 8).

The detailed data for the first 13 problems are listed below, where \( y^1 \) means the starting point for box constraints, and the starting points for ball constraints are always set to be \( a \). For simplicity, we may write the vectors as row vectors, and use some MATLAB notations: \( \text{mod}(x,y) \) finds the remainder after division of \( x \) by \( y \); \( \text{ones}(p,q) \) and \( \text{zeros}(p,q) \) are \( p \)-by-\( q \) matrices of ones and zeros, respectively.

- **CB2**: \( y^1 = (3.3), l = (2.2), u = (4.4), a = (0.0), b = 1; \)
- **CB3**: \( y^1 = (3.3), l = (2.0), u = (4.3), a = (3.3), b = 1; \)
- **DEM**: \( y^1 = (0.5, -2.5), l = (0.1, -3), u = (1.1, -2), a = (0, 0), b = 1.5; \)
- **QL**: \( y^1 = (2.3), l = (1.3, 2.5), u = (2.3, 3.5), a = (-1, -1), b = 3; \)
- **LQ**: \( y^1 = (1, 1), l = \left( \frac{1}{\sqrt{2}} + 0.1, \frac{1}{\sqrt{2}} + 0.1 \right), u = \left( \frac{1}{\sqrt{2}} + 1.1, \frac{1}{\sqrt{2}} + 1.1 \right), a = (1, -1), b = 1; \)
- **Mifflin**: \( y^1 = (1.5, 0.5), l = (1.1, 0.1), u = (2.1, 1.1), a = (-2, 2), b = 1; \)
- **Rosen-Suzuki**: \( y^1 = (1.2, 1.0, -3, -0.9), l = (-\infty, 1.1, -\infty, -0.9), u = (+\infty, 2.1, +\infty, 0.1), a = (1, 2, 3, 4), b = 2; \)
- **Shor**: \( y^1 = 2\text{zeros}(5,1), l = (-\infty, 1.1, -\infty, 1.1, -\infty), u = (+\infty, 2.1, +\infty, 2.1, +\infty), a = (1, -1, 1, -1, 1), b = 1.5; \)
- **Maxquad**: \( y^1 = \text{zeros}(10,1), l = -2\text{ones}(10,1), u = -2\text{ones}(10,1), a = \text{zeros}(10,1), b = 3; \)
- **Maxq**: for \( j = 1, \ldots, 10, if \mod(j,2) = 0, y^j_1 = 1.1, else y^j_1 = j; for j = 11, \ldots, 20, if \mod(j,2) = 0, y^j_1 = 0.1, else y^j_1 = -j; for j = 1, \ldots, 20, if \mod(j,2) = 0, l_j = 0.1, u_j = 1.1, else l_j = -\infty, u_j = +\infty; \)
- **Maxl**: \( y^j_1, l, u \) and \( a \) are the same as Maxq, \( b = 2; \)
- **Goffin**: for \( j = 1, \ldots, 25, y^j_1 = 0, l_j = -1, u_j = 1; for j = 26, \ldots, 50, y^j_1 = 3, l_j = 2, u_j = 4; a = (-\text{ones}(1,25), \text{ones}(1,25)), b = 3; \)
MXHILB: $y^1 = 0.1 \text{ones}(50,1)$; for $j = 1, \ldots, 50$, if $\text{mod}(j, 2) = 0$, $l_j = 0.1$, $u_j = 1.1$, else $l_j = -\infty$, $u_j = +\infty$; $a = (-\text{ones}(1,25), \text{ones}(1,25))$, $b = 2$.

Some explanations are necessary for the above data. The bounds and balls for the first 8 problems and Goffin are simply set such that the unconstrained optimal solutions are excluded. The data for Maxquad are set such that the unconstrained solution is included. The starting points for Maxq and Maxl, and the bounds for Maxq, Maxl and MXHILB are suggested by [18].

Three algorithms are compared: PPPBM$^+$ — Algorithm 4.1 of our paper; PPPBM$^-$ — Algorithm 4.1 with the descent test criterion proposed in [8], i.e., replace (51) by (17) in Algorithm 4.1; PPBM — the algorithm in [20]. The comparisons between PPPBM$^+$ and PPPBM$^-$ aim to see the advantage of our new criterion (51) when compared with the criterion (17) proposed in [8]. The comparisons between PPPBM$^+$ and PPBM aim to see the advantage of the partial bundle model when compared with the classic bundle model.

All numerical experiments were implemented by using MATLAB R2013b. Subproblem (50) is cast as a QP and solved by MOSEK [42]. The parameters are selected as $\epsilon = 10^{-4}$, $\eta = 10^{-3}$, $\sigma = 5$, $\alpha = 0.3$. As it is well known that the choice of the proximal parameter $t_k$ and the bundle size may have considerable influence on computational results, we choose a suitable $i$ for each test problems, and set a maximum value $M = \min\{10n, 50\}$ for the bundle sizes. The index $i$ in Step 5 is selected basically in order $1, \ldots, m$, but we first test the indices that violate (i.e. (51) holds) in previous iterations, since they are more likely to violate at this time.

The numerical results are reported in Tables 1, 2 and 3. The notations are: the number of iterations $N_I$; the number of descent steps $N_D$; the number of component function evaluations $N_{f}$; the approximate optimal value $f^*$; the number of equivalent objective function evaluations $N_{eq}$, i.e., $N_{eq} = N_f / m$.

The comparisons for the first 13 problems are listed in Table 1 (for box constraints) and Table 2 (for ball constraints). From these two tables, we see that PPPBM$^+$ needs obviously less evaluations of the component functions than PPPBM$^-$ and PPBM. The comparisons for the 14th problem are listed in Table 3. For simplicity, we only list the comparisons between PPPBM$^+$ and PPBM, since PPPBM$^+$ obviously outperforms PPPBM$^-$ from Tables 1-2 and our experience. From Table 3, we also see that PPPBM$^+$ needs obviously less evaluations of the component functions than PPBM.

6. Conclusions. In this paper, we have presented a proximal-projection partial bundle method for convex constrained minimax problems. Interesting features of the proposed method are: (i) we extend the partial bundle method of [8] to convex constrained case; (ii) the number of component function evaluations may be reduced greatly by using the partial cutting-planes model; (iii) the constraint set $C$ can be handled efficiently by the proximal-projection technique; (iv) a new descent test criterion is proposed, which has both theoretical and numerical advantages; (v) global convergence is proved, and preliminary numerical results show that our method is promising.

As future work, it would be interesting to test our method for large size problems or practical applications. It is also meaningful to consider the case where the component functions are not necessarily convex.

Acknowledgments. The authors wish to thank the two anonymous referees for their helpful comments, which greatly improved the quality of the paper.
Table 1. Numerical results for $C = \{ x : l \leq x \leq u \}$.

| Problem       | $n,m$ | Algorithm | NI | ND | $N_f$ | $f^*$ | $N_f^*$ |
|---------------|-------|-----------|----|----|-------|-------|---------|
| CB2$\square$  | 2,3   | PPPBM$^\dagger$ | 3  | 2  | 6     | 20.000000 | 2 |
|               |       | PPPBM$^*$  | 11 | 3  | 17    | 20.000000 | 6 |
|               |       | PPBM$^*$   | 6  | 5  | 18    | 20.000000 | 6 |
| CB3$\square$  | 2,3   | PPPBM$^\dagger$ | 3  | 2  | 6     | 16.000000 | 2 |
|               |       | PPPBM$^*$  | 38 | 7  | 51    | 16.000000 | 17 |
|               |       | PPBM$^*$   | 9  | 7  | 27    | 16.000000 | 9 |
| DEM$\square$  | 2,3   | PPPBM$^\dagger$ | 2  | 2  | 6     | -2.500000 | 2 |
|               |       | PPPBM$^*$  | 2  | 2  | 6     | -2.500000 | 2 |
|               |       | PPBM$^*$   | 2  | 1  | 6     | -2.500000 | 2 |
| QL$\square$   | 2,3   | PPPBM$^\dagger$ | 2  | 2  | 4     | 7.940000  | 2 |
|               |       | PPPBM$^*$  | 4  | 2  | 8     | 7.940000  | 3 |
|               |       | PPBM$^*$   | 2  | 1  | 6     | 7.940000  | 2 |
| LQ$\square$   | 2,2   | PPPBM$^\dagger$ | 2  | 2  | 4     | -1.311371 | 2 |
|               |       | PPPBM$^*$  | 4  | 2  | 6     | -1.311371 | 3 |
|               |       | PPBM$^*$   | 2  | 1  | 4     | -1.311371 | 2 |
| Mifflin1$\square$ | 2,2 | PPPBM$^\dagger$ | 3  | 2  | 4     | 3.300000  | 2 |
|               |       | PPPBM$^*$  | 10 | 2  | 12    | 3.300000  | 6 |
|               |       | PPBM$^*$   | 3  | 2  | 6     | 3.300000  | 3 |
| Rosen-Suzuki$\square$ | 4,4 | PPPBM$^\dagger$ | 46 | 14 | 101   | -43.853572 | 25 |
|               |       | PPPBM$^*$  | 177| 10 | 247   | -43.853353 | 62 |
|               |       | PPBM$^*$   | 87 | 14 | 348   | -43.853158 | 87 |
| Shor$\square$ | 5,10  | PPPBM$^\dagger$ | 43 | 16 | 241   | 23.418920 | 24 |
|               |       | PPPBM$^*$  | 197| 7  | 343   | 23.418952 | 34 |
|               |       | PPBM$^*$   | 46 | 16 | 460   | 23.418938 | 46 |
| Maxquad$\square$ | 10,5 | PPPBM$^\dagger$ | 38 | 15 | 145   | -0.841407 | 29 |
|               |       | PPPBM$^*$  | 149| 30 | 416   | -0.841408 | 83 |
|               |       | PPBM$^*$   | 69 | 31 | 345   | -0.841408 | 69 |
| Maxq$\square$ | 20,20 | PPPBM$^\dagger$ | 49 | 13 | 512   | 0.010000  | 26 |
|               |       | PPPBM$^*$  | 663| 16 | 2967  | 0.010000  | 148 |
|               |       | PPBM$^*$   | 45 | 9  | 900   | 0.010000  | 45 |
| Maxl$\square$ | 20,20 | PPPBM$^\dagger$ | 19 | 11 | 269   | 0.100000  | 13 |
|               |       | PPPBM$^*$  | 61 | 24 | 623   | 0.100000  | 31 |
|               |       | PPBM$^*$   | 46 | 18 | 920   | 0.100000  | 46 |
| Goffin$\square$ | 50,50 | PPPBM$^\dagger$ | 27 | 3  | 1074  | 25.000000 | 21 |
|               |       | PPPBM$^*$  | 28 | 4  | 1124  | 25.000000 | 22 |
|               |       | PPBM$^*$   | 29 | 4  | 1450  | 25.000000 | 29 |
| MXHILB$\square$ | 50,50 | PPPBM$^\dagger$ | 41 | 16 | 830   | 0.000038  | 17 |
|               |       | PPPBM$^*$  | 148| 8  | 858   | 0.000037  | 17 |
|               |       | PPBM$^*$   | 36 | 12 | 1800  | 0.000043  | 36 |

REFERENCES

[1] A. Baums, Minimax method in optimizing energy consumption in real-time embedded systems, Automatic Control and Computer Sciences, 43 (2009), 57–62.
[2] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal and C. Sagastizábal, Numerical Optimization: Theoretical and Practical Aspects, Second ed. Springer-Verlag, Berlin Heidelberg New York, 2006.
[3] J. V. Burke, A. S. Lewis and M. L. Overton, A robust gradient sampling algorithm for nonsmooth, nonconvex optimization, SIAM Journal on Optimization, 15 (2005), 751–779.
[4] F. L. Chernousko, Minimax control for a class of linear systems subject to disturbances, Journal of Optimization Theory and Applications, 127 (2005), 535–548.
Table 2. Numerical results for $C = \{ x : \| x - a \| \leq b \}$.

| Problem  | $n, m$ | Algorithm | NI | ND | $N_f$ | $f^*$ | $N_f^*$ |
|----------|-------|-----------|----|----|-------|------|--------|
| CB2°     | 2, 3  | PPPBM°    | 2  | 2  | 6     | 3.343146 | 2     |
|          |       | PPPBM°    | 4  | 2  | 8     | 3.343146 | 3     |
|          |       | PPPBM°    | 2  | 1  | 6     | 3.343146 | 2     |
| CB3°     | 2, 3  | PPPBM°    | 8  | 4  | 14    | 24.479797 | 5     |
|          |       | PPPBM°    | 16 | 7  | 30    | 24.479796 | 10    |
|          |       | PPPBM°    | 11 | 8  | 33    | 24.479795 | 11    |
| DEM°     | 2, 3  | PPPBM°    | 22 | 11 | 45    | −1.499974 | 15    |
|          |       | PPPBM°    | 36 | 2  | 41    | −1.499956 | 14    |
|          |       | PPPBM°    | 19 | 13 | 57    | −1.499950 | 19    |
| QL°      | 2, 2  | PPPBM°    | 5  | 5  | 15    | 25.820215 | 5     |
|          |       | PPPBM°    | 12 | 5  | 24    | 25.820216 | 8     |
|          |       | PPPBM°    | 12 | 9  | 36    | 25.820212 | 12    |
| Mifflin1°| 2, 2  | PPPBM°    | 3  | 2  | 4     | 48.153612 | 2     |
|          |       | PPPBM°    | 4  | 2  | 6     | 48.153612 | 3     |
|          |       | PPPBM°    | 2  | 1  | 4     | 48.153612 | 2     |
| Rosen-Suzuki° | 4, 4 | PPPBM°    | 10 | 6  | 25    | 39.715617 | 6     |
|          |       | PPPBM°    | 12 | 4  | 24    | 39.715617 | 6     |
|          |       | PPPBM°    | 9  | 8  | 36    | 39.715617 | 9     |
| Shor°    | 5, 10 | PPPBM°    | 3  | 2  | 20    | 50.250278 | 2     |
|          |       | PPPBM°    | 6  | 2  | 24    | 50.250278 | 2     |
|          |       | PPPBM°    | 4  | 1  | 40    | 50.250278 | 4     |
| Maxquad° | 10, 5 | PPPBM°    | 35 | 16 | 146   | −0.841408 | 29    |
|          |       | PPPBM°    | 146| 30 | 413   | −0.841406 | 83    |
|          |       | PPPBM°    | 84 | 20 | 420   | −0.841408 | 84    |
| Maxq°    | 20, 20| PPPBM°    | 78 | 8  | 780   | 0.011183  | 39    |
|          |       | PPPBM°    | 612| 18 | 4777  | 0.011193  | 239   |
|          |       | PPPBM°    | 84 | 18 | 1680  | 0.011197  | 84    |
| Maxl°    | 20, 20| PPPBM°    | 22 | 3  | 269   | 0.552786  | 13    |
|          |       | PPPBM°    | 24 | 5  | 309   | 0.552786  | 15    |
|          |       | PPPBM°    | 22 | 2  | 440   | 0.552786  | 22    |
| Goffin°  | 50, 50| PPPBM°    | 27 | 3  | 1074  | 28.786797 | 21    |
|          |       | PPPBM°    | 28 | 4  | 1124  | 28.786797 | 22    |
|          |       | PPPBM°    | 28 | 3  | 1400  | 28.786797 | 28    |
| MXHILB°  | 50, 50| PPPBM°    | 61 | 16 | 847   | 0.613446  | 17    |
|          |       | PPPBM°    | 250| 14 | 937   | 0.613479  | 19    |
|          |       | PPPBM°    | 41 | 21 | 2050  | 0.613444  | 41    |

[5] J. Dattorro, *Convex Optimization † Euclidean Distance Geometry*, second edn, Meboo, 2015.
[6] G. Di Pillo, L. Grippo and S. Lucidi, *A smooth method for the finite minimax problem*, *Mathematical Programming*, 60 (1993), 187–214.
[7] A. Fuduli, M. Gaudioso, G. Giallombardo and G. Miglionico, *A partially inexact bundle method for convex semi-infinite minimax problems*, *Communications in Nonlinear Science and Numerical Simulation*, 21 (2015), 172–180.
[8] M. Gaudioso, G. Giallombardo and G. Miglionico, *An incremental method for solving convex finite min-max problems*, *Mathematics of Operations Research*, 31 (2006), 173–187.
Table 3. Numerical results for Madsen-Schjær-Jacobsen’s problem [27].

| n, m   | Algorithm  | NI | ND | Nf | f∗ | Nf∗ |
|--------|------------|----|----|----|----|-----|
| 30, 58 | PPPBM⁺     | 99 | 15 | 3842 | −19.336156 | 66  |
|        | PFBM       | 99 | 18 | 5742 | −19.336155 | 99  |
| 50, 98 | PPPBM⁺     | 138| 22 | 10502| −62.867219 | 107 |
|        | PFBM       | 161| 25 | 15778| −62.867219 | 161 |
| 100, 198| PPPBM⁺    | 346| 27 | 42247| −281.077609| 275 |
|        | PFBM       | 358| 25 | 70884| −281.077569| 358 |
| 150, 298| PPPBM⁺   | 388| 31 | 81950| −655.540257| 275 |
|        | PFBM       | 510| 31 | 151980| −655.540202| 510 |

[9] M. Gaudioso, G. Giallombardo and G. Miglionico, On solving the Lagrangian dual of integer programs via an incremental approach, *Computational Optimization and Applications*, 44 (2009), 117–138.

[10] W. Hare and J. Nutini, A derivative-free approximate gradient sampling algorithm for finite minimax problems, *Computational Optimization and Applications*, 56 (2013), 1–38.

[11] W. Hare and M. Macklem, Derivative-free optimization methods for finite minimax problems, *Optimization Methods and Software*, 28 (2013), 300–312.

[12] S. X. He and Y. Y. Nie, A class of nonlinear Lagrangian algorithms for minimax problems, *Journal of Industrial and Management Optimization*, 9 (2013), 75–97.

[13] M. Huang, X. J. Liang, Y. Lu and L. P. Pang, The bundle scheme for solving arbitrary eigenvalue optimizations, *Journal of Industrial and Management Optimization*, 13 (2017), 659–680.

[14] J. B. Jian, X. L. Zhang, R. Quan and Q. Ma, Generalized monotone line search SQP algorithm for constrained minimax problems, *Optimization*, 58 (2009), 101–131.

[15] J. B. Jian, X. D. Mo, L. J. Qiu, S. M. Yang and F. S. Wang, Simple sequential quadratically constrained quadratic programming feasible algorithm with active identification sets for constrained minimax problems, *Journal of Optimization Theory and Applications*, 160 (2014), 158–188.

[16] J. B. Jian, C. M. Tang and F. Tang, A feasible descent bundle method for inequality constrained minimax problems (in Chinese), *Science China: Mathematics*, 45 (2015), 2001–2024.

[17] E. Karas, A. Ribeiro, C. Sagastizábal and M. Solodov, A bundle-filter method for nonsmooth convex constrained optimization, *Mathematical Programming*, 116 (2009), 297–320.

[18] N. Karmitsa, *Test Problems for Large-Scale Nonsmooth Minimization*, Tech. Rep. No. B. 4/2007, Department of Mathematical Information Technology, University of Jyväskylä, Finland, 2007.

[19] K. C. Kiwiel, A projection-proximal bundle method for convex nondifferentiable minimization, In: M. Théra, R. Tichatschke (eds.) Ill-posed Variational Problems and Regularization Techniques, Lecture Notes in Econ. Math. Systems, Springer-Verlag, Berlin, 477 (1999), 137–150.

[20] K. C. Kiwiel, A proximal-projection bundle method for Lagrangian relaxation, including semidefinite programming, *SIAM Journal on Optimization*, 17 (2006), 1015–1034.

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

[22] C. Lemaréchal, An extension of Davidon methods to nondifferentiable problems, *Mathematical Programming Study*, 3 (1975), 95–109.

[23] X. S. Li and S. C. Fang, On the entropic regularization method for solving min-max problems with applications, *Mathematical Methods of Operations Research*, 46 (1997), 119–130.

[24] Y. P. Li and G. H. Huang, Inexact minimax regret integer programming for long-term planning of municipal solid waste management – part a: Methodology development, *Environmental Engineering Science*, 26 (2009), 209–218.

[25] G. Liuzzi, S. Lucidi and M. Sciandrone, A derivative-free algorithm for linearly constrained finite minimax problems, *SIAM Journal on Optimization*, 16 (2006), 1054–1075.
[26] L. Lukšan and J. Vlček, Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization, Tech. Rep. No. 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2000.

[27] K. Madsen and H. Schjær-Jacobsen, Linearly constrained minimax optimization, Mathematical Programming, 14 (1978), 208–223.

[28] C. Michelot and F. Plastria, An extended multifacility minimax location problem revisited, Annals of Operations Research, 111 (2002), 167–179.

[29] A. Nedić and D. P. Bertsekas, Incremental subgradient methods for nondifferentiable optimization, SIAM Journal on Optimization, 12 (2001), 109–138.

[30] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, N.J., 1970.

[31] B. Rustem and Q. Nguyen, An algorithm for the inequality-constrained discrete min-max problem, SIAM Journal on Optimization, 8 (1998), 265–283.

[32] C. Sagastizábal and M. Solodov, An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter, SIAM Journal on Optimization, 16 (2005), 146–169.

[33] N. Z. Shor, Minimization Methods for Non-differentiable Functions, Springer-Verlag, Berlin, 1985.

[34] C. M. Tang, H. Y. Chen, J. B. Jian, An improved partial bundle method for linearly constrained minimax problems, Statistics, Optimization and Information Computing, 4 (2016), 84–98.

[35] C. M. Tang and J. B. Jian, Strongly sub-feasible direction method for constrained optimization problems with nonsmooth objective functions, European Journal of Operational Research, 218 (2012), 28–37.

[36] C. M. Tang, S. Liu, J. B. Jian and J. L. Li, A feasible SQP-GS algorithm for nonconvex, nonsmooth constrained optimization, Numerical Algorithms, 65 (2014), 1–22.

[37] A. Vardi, New minimax algorithm, Journal of Optimization Theory and Applications, 75 (1992), 613–634.

[38] F. S. Wang and K. C. Zhang, A hybrid algorithm for nonlinear minimax problems, Annals of Operations Research, 164 (2008), 167–191.

[39] F. S. Wang and K. C. Zhang, A hybrid algorithm for linearly constrained minimax problems, Annals of Operations Research, 206 (2013), 501–525.

[40] S. Y. Wang, Y. Yamamoto and M. Yu, A minimax rule for portfolio selection in frictional markets, Mathematical Methods of Operations Research, 57 (2003), 141–155.

[41] S. Xu, Smoothing method for minimax problems, Computational Optimization and Applications, 20 (2001), 267–279.

[42] MOSEK: The MOSEK optimization toolbox for MATLAB manual, Version 7.1 (2016). MOSEK ApS, Denmark, http://www.mosek.com.

Received September 2016; revised January 2018.

E-mail address: cmrtang@gxu.edu.cn
E-mail address: jianjb@gxu.edu.cn
E-mail address: g.li@unsw.edu.au