

An improved partial bundle method for linearly constrained minimax problems

Chunming Tang¹, Huangyue Chen¹, Jinbao Jian^{2,*}

¹College of Mathematics and Information Science, Guangxi University, Nanning 530004, P. R. China.

²College of Mathematics and Information Science, Yulin Normal University; Guangxi Colleges and Universities Key Lab of Complex System Optimization and Big Data Processing, Yulin 537000, P. R. China.

(Received: 21 February 2016; Accepted: 26 February 2016)

Abstract In this paper, we propose an improved partial bundle method for solving linearly constrained minimax problems. In order to reduce the number of component function evaluations, we utilize a partial cutting-planes model to substitute for the traditional one. At each iteration, only one quadratic programming subproblem needs to be solved to obtain a new trial point. An improved descent test criterion is introduced to simplify the algorithm. The method 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 established. Moreover, we utilize the subgradient aggregation strategy to control the size of the bundle and therefore overcome the difficulty of computation and storage. Finally, some preliminary numerical results show that the proposed method is effective.

Keywords Minimax problems, Bundle method, Partial cutting-planes model, Global convergence, Subgradient aggregation

Mathematics subject classifications (2000) 90C30, 49M30

DOI: 10.19139/soic.v4i1.205

1. Introduction

We consider the linearly constrained minimax problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \langle a_i, x \rangle \leq b_i, \quad i \in I = \{1, \dots, p\}, \end{aligned} \quad (1)$$

where the objective function $f(x) = \max\{f_j(x), j \in J\}$ with $J = \{1, \dots, m\}$, the component functions f_j ($j \in J$): $\mathbb{R}^n \rightarrow \mathbb{R}$ are convex but not necessarily differentiable, and a_i ($i \in I$) $\in \mathbb{R}^n$, b_i ($i \in I$) $\in \mathbb{R}$. Denote by $\langle x, y \rangle = x^T y$ the inner product of vectors x and y . The feasible set of problem (1) is denoted by $X = \{x \in \mathbb{R}^n : \langle a_i, x \rangle \leq b_i, i \in I\}$.

Minimax problems are a special and important class of nonsmooth optimization problems, whose essence is to make an “optimal” decision under the “worst” case. They widely appear in many fields, such as location problems [1], portfolio strategy [23], optimizing energy consumption [2], etc. Furthermore, many mathematical problems can be transformed to minimax problems, such as ℓ_1 and ℓ_∞ approximation [21], system of nonlinear equations [26].

*Correspondence to: Jinbao Jian (Email: jianjb@gxu.edu.cn). College of Mathematics and Information Science, Yulin Normal University, 299 Education Middle Road, Yulin, Guangxi, P. R. China (537000).

The case where the component functions f_i ($i \in I$) are all continuously differentiable is well studied, see e.g., [17, 24, 25]. In what follows, we particularly concern the case where the component functions are not necessarily differentiable. In order to reduce the number of component function evaluations, Gaudioso et al. [6] proposed an incremental bundle method for solving convex unconstrained minimax problems, in which a partial cutting-planes model of the objective function f is introduced. Hare & Macklem [7] and Hare & Nutini [8] proposed a derivative-free gradient sampling method for solving unconstrained minimax problems by applying the gradient sampling idea of Burke et al. [4]. Liuzzi et al. [18] presented a derivative-free method for linearly constrained minimax problems by using a smoothing technique based on an exponential penalty function. Jian et al. [9] proposed a feasible descent bundle method for general inequality constrained minimax problems by using the partial cutting-planes model of [6]. Tang et al. [22] proposed a proximal-projection partial bundle method for convex constrained minimax problems by combining the partial cutting-planes model of [6] with the proximal-projection idea of [13, 15].

In this paper, we propose an improved partial bundle method for solving linearly constrained minimax problem (1). Its main interesting features are summarized as follows.

(i) Our method not only extends the method of [6] to linearly constrained case, but also improves the descent test criterion used in [6] and [9] (see Remark 1 for details).

(ii) Compared to [9], our method can easily deal with linear constraints, and guarantee that all the trial points are feasible.

(iii) Compared to [22], at each iteration, only one quadratic programming (QP) subproblem needs to be solved to obtain a new trial point, while two subproblems are required to solve in [22].

(iv) By making use of the partial cutting-planes model, the number of component function evaluations may be reduced greatly. The objective function is monotonically decreasing on the sequence of stability centers.

(v) The subgradient aggregation strategy [11] is used to control the size of the bundle and therefore overcome the difficulty of computation and storage.

(vi) Global convergence of the algorithm is established, and some preliminary numerical results show that our method is effective.

The paper is organized as follows. In section 2, we review some previous methods which are closely related to our method. In section 3, we present the details of our algorithm and discuss its properties. Global convergence of the algorithm is established in section 4. Improvement of the algorithm by subgradient aggregation is given in Section 5. Preliminary numerical results are reported in sections 6. Conclusions are presented in section 7. We denote by $\|\cdot\|$ the Euclidean norm in \mathbb{R}^n . The subdifferential (in convex analysis) of a function f at any point x is denoted by $\partial f(x)$, and each element $g \in \partial f(x)$ is called a subgradient.

2. Preliminaries

In this section, we review briefly the cutting-planes method, partial cutting-planes model and bundle methods. For simplicity, we consider momentarily the following unconstrained problem

$$\min \{f(x) : x \in \mathbb{R}^n\},$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex but not necessarily differentiable.

2.1. Cutting-planes method and partial cutting-planes model

Let k be the current iteration index, and $y^l, l \in L^k := \{1, \dots, k\}$ be given points (generated in previous iterations) with subgradients $g^l \in \partial f(y^l)$. The linearizations of $f(x)$ at y^l are given by

$$\bar{f}^l(y) = f(y^l) + \langle g^l, y - y^l \rangle, \quad l \in L^k.$$

The *cutting-planes method* [5, 10] uses the following cutting-planes (piecewise affine) model of f at the k -th iteration:

$$\tilde{f}_{\text{cp}}^k(y) = \max_{l \in L^k} \{f(y^l) + \langle g^l, y - y^l \rangle\},$$

which is a lower approximation to f by convexity, i.e., $\tilde{f}_{\text{cp}}^k(x) \leq f(x)$.

The new point y^{k+1} is generated by solving

$$\min_{y \in \mathbb{R}^n} \tilde{f}_{\text{cp}}^k(y),$$

which is equivalent to the linear programming

$$\begin{aligned} \min_{v \in \mathbb{R}, y \in \mathbb{R}^n} \quad & v \\ \text{s.t.} \quad & f(y^l) + \langle g^l, y - y^l \rangle \leq v, \quad l \in L^k. \end{aligned}$$

If the cutting-planes method is applied directly to the objective function of problem (1), then at each point y^l , we need to calculate the objective value $f(y^l)$, and therefore we have to evaluate all the values of the component functions f_j ($j \in J$) at y^l . This is time-consuming if the number of the component functions is large.

In order to reduce the number of component function evaluations, Gaudioso et al. [6] proposed that at each point y^l just evaluate one of the component functions, say $f_{j_l}(y^l)$, for some $j_l \in J$, along with a subgradient $g_{j_l}^l \in \partial f_{j_l}(y^l)$, and then define the linearizations:

$$\bar{f}_{j_l}(y) = f_{j_l}(y^l) + \langle g_{j_l}^l, y - y^l \rangle, \quad l \in L^k. \quad (2)$$

Therefore, the so-called *partial cutting-planes model* is defined by

$$\tilde{f}_{\text{pcp}}^k(y) = \max_{l \in L^k} \{f_{j_l}(y^l) + \langle g_{j_l}^l, y - y^l \rangle\}. \quad (3)$$

It also holds that $\tilde{f}_{\text{pcp}}^k(x) \leq f(x)$.

2.2. Bundle methods

Bundle methods [16, 27, 3] can be considered stabilized variants of cutting-planes method. Let y^l , $l \in L^k$ be trial points generated in past iterations. Bundle methods keep memory of these points, their function values and subgradients in a bundle of information:

$$\mathcal{B}_k = \{(y^l, f(y^l), g^l), \quad l \in L^k\}, \quad (4)$$

and a point x^k (called *stability center*) which has the “best” objective value obtained so far.

Proximal bundle methods [12, 14] are a typical class of bundle methods, which generate a new trial point y^{k+1} by solving

$$\min_{l \in L^k} \tilde{f}_{\mathcal{B}}^k(y) + \frac{1}{2t_k} \|y - x^k\|^2,$$

where $\tilde{f}_{\mathcal{B}}^k$ built on the information in \mathcal{B}_k is a piecewise affine model of f , and $t_k > 0$ is proximal parameter that controls the distance from x^k to y^{k+1} .

If the function f achieves sufficient decrease at y^{k+1} compared to $f(x^k)$, then a *descent step* is declared, and the stability center is updated by $x^{k+1} = y^{k+1}$; otherwise a *null step* is declared, and set $x^{k+1} = x^k$.

3. The algorithm

In this section, combining the partial cutting-planes model and the idea of proximal bundle methods, we propose an improved partial bundle method for problem (1). The partial cutting-planes model (3) would be applied to the objective function of problem (1).

Let $x^k \in X$ be the current stability center. Define the linearization errors:

$$\alpha_l^k := f(x^k) - \bar{f}_{j_l}(x^k) = f(x^k) - [f_{j_l}(y^l) + \langle g_{j_l}^l, x^k - y^l \rangle], \quad l \in L^k, \quad (5)$$

which are nonnegative by convexity. The bundle set (4) can be rewritten as

$$\mathcal{B}_k := \{(y^l, f_{j_l}(y^l), g_{j_l}^l, \alpha_l^k), l \in L^k\}.$$

From (3) and (5), we have

$$\begin{aligned} \tilde{f}_{\text{pcp}}^k(y) &= \max_{l \in L^k} \{f_{j_l}(y^l) + \langle g_{j_l}^l, y - y^l \rangle\} \\ &= \max_{l \in L^k} \{f(x^k) - \alpha_l^k - \langle g_{j_l}^l, x^k - y^l \rangle + \langle g_{j_l}^l, y - y^l \rangle\} \\ &= f(x^k) + \max_{l \in L^k} \{-\alpha_l^k + \langle g_{j_l}^l, y - x^k \rangle\}. \end{aligned} \tag{6}$$

We consider the following subproblem

$$y^{k+1} := \arg \min \left\{ \tilde{f}_{\text{pcp}}^k(y) + \frac{1}{2t_k} \|y - x^k\|^2, y \in X \right\}, \tag{7}$$

where $t_k > 0$ is proximal parameter. From (6), we know that the solution y^{k+1} of (7) is also the solution to

$$\min \left\{ \max_{l \in L^k} \{-\alpha_l^k + \langle g_{j_l}^l, y - x^k \rangle\} + \frac{1}{2t_k} \|y - x^k\|^2, y \in X \right\}.$$

Letting $d = y - x^k$, we can equivalently consider the following QP subproblem

$$\begin{aligned} \text{QP}(\mathcal{B}_k) \quad & \min_{v \in \mathbb{R}, d \in \mathbb{R}^n} \quad v + \frac{1}{2t_k} \|d\|^2 \\ & \text{s.t.} \quad -\alpha_l^k + \langle g_{j_l}^l, d \rangle \leq v, \quad l \in L^k, \\ & \quad \quad \langle a_i, x^k + d \rangle \leq b_i, \quad i \in I. \end{aligned}$$

It is obvious that $(z, d) = (0, 0)$ is a feasible solution of $\text{QP}(\mathcal{B}_k)$. Moreover, $\text{QP}(\mathcal{B}_k)$ has a unique optimal solution, denoted by (v_k, d^k) . Therefore, we have $y^{k+1} = x^k + d^k$.

The Lagrangian dual problem of $\text{QP}(\mathcal{B}_k)$ has the form

$$\begin{aligned} \text{DP}(\mathcal{B}_k) \quad & \min_{\lambda \in \mathbb{R}^{|L^k|}, \mu \in \mathbb{R}^{|I|}} \quad \frac{1}{2} t_k \left\| \sum_{l \in L^k} \lambda_l g_{j_l}^l + \sum_{i \in I} \mu_i a_i \right\|^2 + \sum_{l \in L^k} \lambda_l \alpha_l^k - \sum_{i \in I} \mu_i (\langle a_i, x^k \rangle - b_i), \\ & \text{s.t.} \quad \lambda_l \geq 0, \quad l \in L^k, \quad \mu_i \geq 0, \quad i \in I, \quad \sum_{l \in L^k} \lambda_l = 1. \end{aligned}$$

Let (λ^k, μ^k) be the optimal solution of $\text{DP}(\mathcal{B}_k)$. It is not difficult to verify the following relations:

$$d^k = -t_k g^k,$$

$$v_k = -t_k \|g^k\|^2 - \epsilon_k \leq 0, \tag{8}$$

where

$$g^k := \sum_{l \in L^k} \lambda_l^k g_{j_l}^l + \sum_{i \in I} \mu_i^k a_i, \tag{9}$$

$$\epsilon_k := \sum_{l \in L^k} \lambda_l^k \alpha_l^k - \sum_{i \in I} \mu_i^k (\langle a_i, x^k \rangle - b_i). \tag{10}$$

Now we present the details of our algorithm.

Algorithm 1

Step 0. Initialization. Select $y^1 \in X$, and set $x^1 = y^1$, $L^1 = \{1\}$, the initial bundle $\mathcal{B}_1 = \{(y^1, f(y^1), g^1, 0)\}$ with $g^1 \in \partial f(y^1)$. Select $\epsilon > 0$, $\sigma > 1$, $\eta > 0$, $\beta \in (0, 1)$, $\bar{t} > 0$. Set $t_1 = \bar{t}$, $k := 1$.

Step 1. Proximal point finding. Solve QP(\mathcal{B}_k) (or DP(\mathcal{B}_k)) to obtain (v_k, d^k) and (λ^k, μ^k) . Set $y^{k+1} = x^k + d^k$, and calculate g^k and ϵ_k by (9) and (10), respectively.

Step 2. Stopping criterion.

If $\|g^k\| \leq \eta$, $\epsilon_k \leq \epsilon$, STOP (approximate optimality achieved);

else if $\|g^k\| \leq \eta$, $\epsilon_k > \epsilon$, go to Step 3;

else $\|g^k\| > \eta$, go to Step 4.

Step 3. Bundle reset. Set $x^{k+1} := x^k$, $y^{k+1} := x^k$, $L^{k+1} = \{k+1\}$, and make a bundle reset

$$\mathcal{B}_{k+1} = \{(y^{k+1}, f(y^{k+1}), g^{k+1}, 0)\},$$

where $g^{k+1} \in \partial f(y^{k+1})$. Set $t_{k+1} = t_k/\sigma$, $k := k+1$, and return to Step 1.

Step 4. Descent test. Extract any index h from the function index set J . If

$$f_h(y^{k+1}) - f(x^k) > \beta v_k, \quad (11)$$

then set $x^{k+1} := x^k$ (null step), $L^{k+1} = L^k \cup \{k+1\}$, and update the bundle by

$$\mathcal{B}_{k+1} = \mathcal{B}_k \cup \{(y^{k+1}, f_h(y^{k+1}), g_h^{k+1}, \alpha_{k+1}^{k+1})\},$$

where

$$g_h^{k+1} \in \partial f_h(y^{k+1}), \quad \alpha_{k+1}^{k+1} = f(x^{k+1}) - [f_h(y^{k+1}) + \langle g_h^{k+1}, x^{k+1} - y^{k+1} \rangle].$$

Restore the function index set by setting $J = \{1, \dots, m\}$. Let $t_{k+1} = t_k$, $k := k+1$, and return to Step 1.

Step 5. Index removing. Set $J := J \setminus \{h\}$. If $J \neq \emptyset$, then return to Step 4.

Step 6. Updating. Set $x^{k+1} := y^{k+1}$ (descent step), $L^{k+1} = L^k \cup \{k+1\}$, and update the bundle by

$$\mathcal{B}_{k+1} = \{(y^l, f_{j_l}(y^l), g_{j_l}^l, \alpha_l^{k+1}), l \in L^k\} \cup \{(y^{k+1}, f(y^{k+1}), g^{k+1}, 0)\},$$

where $g^{k+1} \in \partial f(y^{k+1})$ and

$$\alpha_l^{k+1} = \alpha_l^k + f(x^{k+1}) - f(x^k) - \langle g_{j_l}^l, x^{k+1} - x^k \rangle, \quad l \in L^k.$$

Restore the function index set by setting $J = \{1, \dots, m\}$, let $t_{k+1} = \bar{t}$, $k := k+1$, and return to Step 1.

Before giving some comments on Algorithm 1, we first present three lemmas as follows.

Lemma 1

Let x^k , d^k , g^k and ϵ_k be generated by Algorithm 1. Then

$$f(x^k) \leq f(x) + \|g^k\| \|x - x^k\| + \epsilon_k, \quad \forall x \in X.$$

Proof

From (2) and (5), we have

$$\begin{aligned} f(x) &\geq \bar{f}_{j_l}(x) = f_{j_l}(y^l) + \langle g_{j_l}^l, x - y^l \rangle \\ &= f_{j_l}(y^l) + f(x^k) - f(x^k) + \langle g_{j_l}^l, x^k - y^l \rangle + \langle g_{j_l}^l, x - x^k \rangle \\ &= -\alpha_l^k + f(x^k) + \langle g_{j_l}^l, x - x^k \rangle. \end{aligned} \quad (12)$$

On the other hand, for $x \in X$, we have

$$0 \geq \langle a_i, x \rangle - b_i = \langle a_i, x - x^k \rangle + \langle a_i, x^k \rangle - b_i. \quad (13)$$

Multiplying both sides of (12) and (13) by λ_l^k and μ_i^k , respectively, and summing up for $l \in L^k$ and $i \in I$, we have

$$\begin{aligned} f(x) &\geq - \sum_{l \in L^k} \lambda_l^k \alpha_l^k + f(x^k) + \sum_{l \in L^k} \lambda_l^k \langle g_{j_l}^l, x - x^k \rangle + \sum_{i \in I} \mu_i^k \langle a_i, x - x^k \rangle + \sum_{i \in I} \mu_i^k (\langle a_i, x^k \rangle - b_i) \\ &= f(x^k) + \langle g^k, x - x^k \rangle - \epsilon_k, \quad \forall x \in X. \end{aligned}$$

This together with Cauchy-Schwarz inequality proves the lemma. \square

Lemma 2

(Cut Property). Suppose that $y^{k+1} = x^k + d^k$ and $x^{k+1} = x^k$. If there exists an index $h \in J$ such that

$$f_h(y^{k+1}) - f(x^k) > \beta v_k, \quad (14)$$

and the bundle is modified by

$$\mathcal{B}_{k+1} = \mathcal{B}_k \cup \{(y^{k+1}, f_h(y^{k+1}), g_h^{k+1}, \alpha_{k+1}^{k+1})\}, \quad L^{k+1} = L^k \cup \{k+1\},$$

where

$$g_h^{k+1} \in \partial f_h(y^{k+1}), \quad \alpha_{k+1}^{k+1} = f(x^{k+1}) - [f_h(y^{k+1}) + \langle g_h^{k+1}, x^{k+1} - y^{k+1} \rangle],$$

then (v_k, d^k) is not feasible for subproblem QP(\mathcal{B}_{k+1}).

Proof

From the construction of \mathcal{B}_{k+1} , we know that subproblem QP(\mathcal{B}_{k+1}) has a constraint of the form

$$-\alpha_{k+1}^{k+1} + \langle g_h^{k+1}, d \rangle \leq v.$$

However, from (14), it follows

$$\begin{aligned} -\alpha_{k+1}^{k+1} + \langle g_h^{k+1}, d^k \rangle &= [f_h(y^{k+1}) + \langle g_h^{k+1}, x^{k+1} - y^{k+1} \rangle] - f(x^{k+1}) + \langle g_h^{k+1}, d^k \rangle \\ &= f_h(y^{k+1}) - f(x^k) > \beta v_k > v_k. \end{aligned}$$

So (v_k, d^k) is not feasible for QP(\mathcal{B}_{k+1}), and the proof is complete. \square

Lemma 3

(Feasible descent property). If Algorithm 1 enters Step 6, then $x^{k+1} \in X$ and

$$f(x^{k+1}) - f(x^k) \leq -\beta t_k \eta^2 < 0.$$

Proof

If Algorithm 1 enters Step 6, then

$$\|g^k\| > \eta, \quad (15)$$

$$f_j(x^{k+1}) - f(x^k) \leq \beta v_k, \quad \forall j \in J. \quad (16)$$

From (16) and (8), we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &= \max_{j \in J} \{f_j(x^{k+1}) - f(x^k)\} \leq \beta v_k \\ &= -\beta(\epsilon_k + t_k \|g^k\|^2) \\ &\leq -\beta t_k \|g^k\|^2. \end{aligned} \quad (17)$$

This together with (15) shows the lemma. \square

Based on the lemmas above, we give some comments on Algorithm 1.

Remark 1

(1) In Step 2, if both $\|g^k\|$ and ϵ_k are “sufficiently small”, then from Lemma 1, Algorithm 1 can stop, and we accept the current stability center x^k as an approximate optimal solution. If $\|g^k\|$ is small but ϵ_k is large, then we consider that the partial cutting-planes model is “bad”, and therefore a bundle reset is made. If $\|g^k\|$ is not small, then we accept y^{k+1} as a new trial point.

(2) In Step 3, a bundle reset implies that the model is not reliable enough, so we decrease the proximal parameter such that a “closer” trial point is obtained.

(3) In Step 4, the rule of selecting the index $h \in J$ does not affect the theoretical analysis, but suitable rules are helpful to improve the numerical performance (see [6, 22] for more detailed discussion).

(4) The descent test criterion (11) is proposed in [22], which is simpler than the one of [6] in the sense that it does not contain a problem-data-independent parameter. In fact, [6] (essentially) used the following criterion

$$f_i(y^{k+1}) - f(x^k) > v_k + \theta_k, \quad (18)$$

where $\theta_k > 0$ is a problem-data-independent parameter satisfying $\theta_k/t_k < \eta^2$. This is somewhat restricted such that it may be not easy to choose the parameters θ_k and t_k numerically. Furthermore, the constant β in (11) is very helpful to improve numerical performance. The numerical results in [22] show that the criterion (11) performs better than (18).

(5) In Step 4, if there exists an index h such that (11) holds, then a new cutting-plane built on the component function f_h at y^{k+1} is added, which can improve the model significantly (see Lemma 2).

(6) In Step 5, if $J = \emptyset$, then Algorithm 1 enters Step 6. From Lemma 3, we know that the new trial point y^{k+1} satisfies the feasible descent property, so the stability center is updated by this point.

4. Global convergence

In this section, we establish the global convergence of Algorithm 1. The following basic assumption is required.

Assumption 1

The level set $\Gamma := \{x \in X : f(x) \leq f(x^1)\}$ for problem (1) is bounded.

From Lemma 3, we know that the sequence $\{f(x^k)\}$ is monotonically decreasing. So the sequence $\{x^k\}$ of stability centers belongs to the level set Γ .

Algorithm 1 must take only one of the following two cases.

- (1) The algorithm loops between Step 1 and Step 5, generates null steps, and the stability center does not change;
- (2) The algorithm enters Step 6, generates a descent step, and the stability center is updated.

In what follows, we will show that Algorithm 1 is well defined, i.e., the number of loops between Step 1 and Step 5 is finite, and therefore Algorithm 1 must enter Step 6 after a finite number of iterations. In particular, we will show that the following claims hold when the stability center does not change.

- (a) The algorithm passes finitely many times through Step 3;
- (b) The algorithm passes finitely many times through Step 5;
- (c) The number of loops between Step 1 and Step 4 is finite.

The following lemma provides an upper bound for the number of bundle resets, and therefore claim (a) holds.

Lemma 4

Suppose that Algorithm 1 reaches a certain stability center $x^{\bar{k}}$ which remains unchanged. Then Algorithm 1 passes finitely many times through Step 3. In particular, let $N_{\bar{k}}$ be the number of times passing through Step 3, then there

exists a positive constant M such that

$$N_{\bar{k}} \leq \left\lceil \frac{\ln \frac{\bar{t}M^2}{\epsilon}}{\ln \sigma} \right\rceil, \quad (19)$$

where notation $\lceil A \rceil$ rounds A to the nearest integer greater than or equal to A .

Proof

From the statement of the lemma, we know that $x^k \equiv x^{\bar{k}}, \forall k \geq \bar{k}$. Denote $k_r (\geq \bar{k})$ the r -th time the algorithm enters Step 3. After entering Step 3, the bundle is reset by

$$\mathcal{B}_{k_r+1} = \{(x^{\bar{k}}, f(x^{\bar{k}}), g^{\bar{k}}, 0)\},$$

where $g^{\bar{k}} \in \partial f(x^{\bar{k}})$, and the proximal parameter is updated by

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

For $k_r + 1 \leq k < k_{r+1}$, from (8) we have

$$v_k = -t_k \|g^k\|^2 - \epsilon_k \leq -t_k \|g^k\|^2 = -\|d^k\|^2/t_k = -\sigma^r \|d^k\|^2/\bar{t}. \quad (20)$$

On the other hand, by the constraints of $\text{QP}(\mathcal{B}_k)$, we have

$$v_k \geq \langle g^{\bar{k}}, d^k \rangle \geq -\|d^k\| \|g^{\bar{k}}\|,$$

which together with (20) shows that

$$-\|d^k\| \|g^{\bar{k}}\| \leq v_k \leq -\sigma^r \|d^k\|^2/\bar{t}.$$

Thus

$$\|d^k\| \leq \frac{\bar{t}}{\sigma^r} \|g^{\bar{k}}\|. \quad (21)$$

Combining (20) with (21), we have

$$v_k \geq -\frac{\bar{t}}{\sigma^r} \|g^{\bar{k}}\|^2. \quad (22)$$

In addition, by Assumption 1, there exists a positive constant M such that $\|g^{\bar{k}}\| \leq M$, so from (8) and (22), we have

$$\epsilon_k \leq -v_k \leq \frac{\bar{t}M^2}{\sigma^r}.$$

Hence, if

$$r \geq \frac{\ln \frac{\bar{t}M^2}{\epsilon}}{\ln \sigma},$$

then

$$\epsilon_k \leq \epsilon.$$

This means that (19) holds. \square

In order to prove (b) and (c), we first prove the following lemma. From Lemma 4, in what follows, we may assume that the bundle reset does not occur.

Lemma 5

Suppose that Algorithm 1 reaches a certain stability center $x^{\bar{k}}$ which keeps unchanged, and the bundle reset does not occur. Let p, q be two iteration indices with $p > q \geq \bar{k}$. Let (v_p, d^p) , w_p and (v_q, d^q) , w_q be the optimal solutions and optimal values of $\text{QP}(\mathcal{B}_p)$ and $\text{QP}(\mathcal{B}_q)$, respectively, then

- (i) $w_p \geq w_q + \|d^p - d^q\|^2/2\bar{t}$;
- (ii) $\|d^p - d^q\| \rightarrow 0, v_p - v_q \rightarrow 0$, as $q \rightarrow \infty$.

Proof

(i) Since (v_p, d^p) is feasible for subproblem QP(\mathcal{B}_q), we have

$$-\alpha_l^q + \langle g_{j_l}^l, d^p \rangle \leq v_p, \quad l \in L^q; \quad (23)$$

$$\langle a_i, x^{\bar{k}} + d^p \rangle \leq b_i, \quad i \in I. \quad (24)$$

Multiplying both sides of (23) and (24) by λ_l^q and μ_i^q , respectively, and summing up, we have

$$-\sum_{l \in L^q} \lambda_l^q \alpha_l^q + \sum_{l \in L^q} \lambda_l^q \langle g_{j_l}^l, d^p \rangle + \sum_{i \in I} \mu_i^q \langle a_i, x^{\bar{k}} + d^p \rangle \leq \sum_{l \in L^q} \lambda_l^q v_p + \sum_{i \in I} \mu_i^q b_i,$$

thus

$$-\epsilon_q + \langle g^q, d^p \rangle \leq v_p. \quad (25)$$

By (8), we have

$$-\epsilon_q + \langle g^q, d^q \rangle = v_q. \quad (26)$$

Combining (25) and (26), we have

$$\langle g^q, d^p - d^q \rangle \leq v_p - v_q. \quad (27)$$

Since the bundle reset does not occur, then it follows $t_q = t_p = \bar{t}$. This together with (27) shows that

$$\begin{aligned} w_p &= v_p + \|d^p\|^2/2\bar{t} = v_p + v_q - v_q + \|d^p + d^q - d^q\|^2/2\bar{t} \\ &= v_q + \|d^q\|^2/2\bar{t} + v_p - v_q + \|d^p - d^q\|^2/2\bar{t} + \langle d^p - d^q, d^q \rangle/\bar{t} \\ &= w_q + \|d^p - d^q\|^2/2\bar{t} + v_p - v_q - \langle g^q, d^p - d^q \rangle \\ &\geq w_q + \|d^p - d^q\|^2/2\bar{t}. \end{aligned}$$

Hence, part (i) holds.

(ii) From (i) and $w_k \leq 0$, we know that the sequence $\{w_k\}_{k \geq \bar{k}}$ is monotonic nondecreasing and bounded above, so it converges, and therefore

$$\|d^p - d^q\| \rightarrow 0, \quad \text{as } q \rightarrow \infty.$$

Moreover, we have

$$\|d^p\|^2 - \|d^q\|^2 \rightarrow 0, \quad \text{as } q \rightarrow \infty.$$

Since

$$w_p - w_q = \frac{1}{2\bar{t}}(\|d^p\|^2 - \|d^q\|^2) + v_p - v_q,$$

we obtain

$$v_p - v_q \rightarrow 0, \quad \text{as } q \rightarrow \infty.$$

□

Based on Lemma 5, we now prove the claims (b) and (c).

Lemma 6

Suppose that Algorithm 1 reaches a certain stability center $x^{\bar{k}}$ which keeps unchanged. Then Algorithm 1 passes finitely many times through Step 5, and the number of loops between Step 1 and Step 4 is finite.

Proof

Suppose by contradiction that one of the following cases occur:

- (i) infinitely many times through Step 5;
- (ii) the number of loops between Step 1 and Step 4 is infinite.

Notice that both cases imply that

$$\|g^k\| > \eta, \quad \forall k \geq \bar{k}, \quad (28)$$

but at the same time there exists an index $j_{k+1} \in J$ generated in Step 4 such that

$$f_{j_{k+1}}(y^{k+1}) - f(x^k) > \beta v_k, \quad \forall k \geq \bar{k}.$$

Hence, for any $p > q \geq \bar{k}$, it follows

$$\begin{aligned} v_p &\geq \langle g_{j_{q+1}}^{q+1}, d^p \rangle - \alpha_{q+1}^{q+1} \\ &= \langle g_{j_{q+1}}^{q+1}, d^p \rangle - f(x^{q+1}) + f_{j_{q+1}}(y^{q+1}) + \langle g_{j_{q+1}}^{q+1}, x^{q+1} - y^{q+1} \rangle \\ &= \langle g_{j_{q+1}}^{q+1}, d^p \rangle - f(x^q) + f_{j_{q+1}}(y^{q+1}) + \langle g_{j_{q+1}}^{q+1}, x^q - y^{q+1} \rangle \\ &= \langle g_{j_{q+1}}^{q+1}, d^p - d^q \rangle + f_{j_{q+1}}(y^{q+1}) - f(x^q) \\ &> \langle g_{j_{q+1}}^{q+1}, d^p - d^q \rangle + \beta v_q. \end{aligned}$$

This further implies

$$v_p - v_q > \langle g_{j_{q+1}}^{q+1}, d^p - d^q \rangle + (\beta - 1)v_q.$$

Passing to the limit as $q \rightarrow \infty$, and combining Lemma 5 and Assumption 1, we have

$$v_q \rightarrow 0,$$

which contradicts (28), so the lemma holds. \square

From Lemma 4 and Lemma 6, the following theorem holds immediately.

Theorem 1

Algorithm 1 is well defined, i.e., the stability center must be updated after a finite number of iterations.

Now we prove the global convergence of Algorithm 1.

Theorem 2

For any $\epsilon > 0$ and $\eta > 0$, Algorithm 1 stops after a finite number of iterations at a point satisfying the approximate optimality condition

$$\|g^{k_*}\| \leq \eta \quad \text{and} \quad \epsilon_{k_*} \leq \epsilon,$$

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

Proof

Suppose by contradiction that Algorithm 1 cannot stop finitely. Then from Theorem 1, we know that the stability centers are updated infinitely many times. Let k_s ($s = 1, 2, \dots$) be the indices of stability centers. From Lemma 3, we have

$$f(x^{k_{s+1}}) - f(x^{k_s}) = f(x^{k_{s+1}}) - f(x^{k_{s+1}-1}) \leq -\beta t_{k_{s+1}-1} \eta^2. \quad (29)$$

Let N_{k_s} be the number of bundle resets from x^{k_s} to $x^{k_{s+1}}$, then it follows

$$t_{k_{s+1}-1} = \frac{\bar{t}}{\sigma^{N_{k_s}}}. \quad (30)$$

Combining (29) and (30), we have

$$f(x^{k_{s+1}}) - f(x^{k_s}) \leq \frac{1}{\sigma^{N_{k_s}}} (-\beta \bar{t} \eta^2),$$

which further implies

$$f(x^{k_{s+1}}) - f(x^{k_1}) \leq -\tau \sum_{l=1}^s \frac{1}{\sigma^{N_{k_l}}}, \quad (31)$$

where $\tau = \beta \bar{t} \eta^2 > 0$.

On the other hand, from Lemma 4, we know that

$$N_{k_l} \leq \bar{N} := \left\lceil \frac{\ln \frac{\bar{t} M^2}{\epsilon}}{\ln \sigma} \right\rceil.$$

This together with (31) shows that

$$f(x^{k_{s+1}}) - f(x^{k_1}) \leq -\tau \sum_{l=1}^s \frac{1}{\sigma^{\bar{N}}}.$$

Passing to the limit as $s \rightarrow \infty$, we have

$$\lim_{s \rightarrow \infty} (f(x^{k_{s+1}}) - f(x^{k_1})) \leq -\infty,$$

which contradicts Assumption 1, so the theorem holds. \square

5. Improvement by subgradient aggregation

In order to control the bundle size and meanwhile keep the theoretical convergence, we utilize the subgradient aggregation strategy [11] to improve Algorithm 1.

Algorithm 2

(Improved version of Algorithm 1)

Step 0. Initialization. Select $y^1 \in X$, and set $x^1 = y^1$, $L^1 = \{1\}$, the initial bundle $\mathcal{B}_1 = \{(y^1, f(y^1), g^1, 0)\}$ with $g^1 \in \partial f(y^1)$, and $\alpha_P^1 = 0$, $P^0 = g^1$. Select $\epsilon > 0$, $\sigma > 1$, $\eta > 0$, $\beta \in (0, 1)$, $\bar{t} > 0$. Set $t_1 = \bar{t}$, $BR = \emptyset$, $k := 1$.

Step 1. Proximal point finding. Solve the subproblem

$$\text{AQP}(\mathcal{B}_k) \quad \begin{array}{ll} \min_{v \in \mathbb{R}, d \in \mathbb{R}^n} & v + \frac{1}{2t_k} \|d\|^2 \\ \text{s.t.} & -\alpha_l^k + \langle g_{j_l}^l, d \rangle \leq v, \quad l \in L^k, \\ & -\alpha_P^k + \langle P^{k-1}, d \rangle \leq v, \\ & \langle a_i, x^k + d \rangle \leq b_i, \quad i \in I \end{array}$$

to obtain the optimal solution (v_k, d^k) and its multipliers $(\lambda^k, \lambda_P^k, \mu^k)$. Let $y^{k+1} = x^k + d^k$, and compute

$$\begin{aligned} g^k &= \sum_{l \in L^k} \lambda_l^k g_{j_l}^l + \lambda_P^k P^{k-1} + \sum_{i \in I} \mu_i^k a_i, \\ \epsilon_k &= \sum_{l \in L^k} \lambda_l^k \alpha_l^k + \lambda_P^k \alpha_P^k - \sum_{i \in I} \mu_i^k (\langle a_i, x^k \rangle - b_i), \\ (P^k, \tilde{\alpha}_P^k) &= \sum_{l \in L^k} \lambda_l^k (g_{j_l}^l, \alpha_l^k) + \lambda_P^k (P^{k-1}, \alpha_P^k). \end{aligned}$$

Step 2. Stopping criterion.

If $\|g^k\| \leq \eta$, $\epsilon_k \leq \epsilon$, STOP (approximate optimality achieved);

else if $\|g^k\| \leq \eta$, $\epsilon_k > \epsilon$, go to Step 3;

else $\|g^k\| > \eta$, go to Step 4.

Step 3. Bundle reset. Set $x^{k+1} := x^k$, $y^{k+1} := x^k$, $L^{k+1} = \{k+1\}$, $BR = \{k+1\}$, and make a bundle reset

$$\mathcal{B}_{k+1} = \{(y^{k+1}, f(y^{k+1}), g^{k+1}, 0)\} \text{ with } g^{k+1} \in \partial f(y^{k+1}).$$

Set $\alpha_P^{k+1} = 0$, $P^k = g^{k+1}$, $t_{k+1} = t_k/\sigma$, $k := k + 1$, and return to Step 1.

Step 4. Descent test. Extract any index h from the function index set J . If

$$f_h(y^{k+1}) - f(x^k) > \beta v_k, \quad (1)$$

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

$$\mathcal{B}_{k+1} = \hat{\mathcal{B}}_k \cup \{(y^{k+1}, f_h(y^{k+1}), g_h^{k+1}, \alpha_{k+1}^{k+1})\},$$

where

$$g_h^{k+1} \in \partial f_h(y^{k+1}), \quad \alpha_{k+1}^{k+1} = f(x^{k+1}) - [f_h(y^{k+1}) + \langle g_h^{k+1}, x^{k+1} - y^{k+1} \rangle].$$

Set $\alpha_P^{k+1} = \tilde{\alpha}_P^k$. Restore the function index set by setting $J = \{1, \dots, m\}$, and let $t_{k+1} := t_k$, $k := k + 1$, and return to Step 1.

Step 5. Index removing. Set $J := J \setminus \{h\}$. If $J \neq \emptyset$, then return to Step 4.

Step 6. Updating. Set $x^{k+1} := y^{k+1}$ (*serious step*). Select $\hat{L}^k \subseteq L^k$, generate its corresponding bundle $\hat{\mathcal{B}}_k \subseteq \mathcal{B}_k$, and update the elements α_l^k ($l \in \hat{L}^k$) in $\hat{\mathcal{B}}_k$ by

$$\alpha_l^k := \alpha_l^k + f(x^{k+1}) - f(x^k) - \langle g_{j_l}^k, x^{k+1} - x^k \rangle, \quad l \in \hat{L}^k.$$

Set $L^{k+1} = \hat{L}^k \cup \{k + 1\}$, and update the bundle by

$$\mathcal{B}_{k+1} = \hat{\mathcal{B}}_k \cup \{(y^{k+1}, f(y^{k+1}), g^{k+1}, 0)\} \quad \text{with } g^{k+1} \in \partial f(y^{k+1}).$$

Set

$$\alpha_P^{k+1} = \tilde{\alpha}_P^k + f(x^{k+1}) - f(x^k) - \langle P^k, x^{k+1} - x^k \rangle.$$

Restore the function index set by setting $J = \{1, \dots, m\}$, let $t_{k+1} := \bar{t}$, $BR = \emptyset$, $k := k + 1$, and return to Step 1.

Remark 2

The choice of subset \hat{L}^k is very flexible, since theoretically speaking, \hat{L}^k can be a singleton or even an empty set. In practice, we can delete some elements from the bundle when its size reaches a preset maximum value, see e.g. [3].

6. Numerical results

In this section, we aim to test the practical effectiveness of Algorithm 2. We tested a set of 15 constrained minimax problems, in which problems Wong 2 and Wong 3 are taken from [19], and the other 13 problems are modifications of the corresponding problems in [19] by imposing box constraints of the form $\ell \leq x \leq u$ (see Table I for detailed data, and see [22] for detailed explanations of the data). For simplicity, we may use some MATLAB notations: `mod(x, y)` finds the remainder after division of x by y ; `ones(p, q)` and `zeros(p, q)` are p -by- q matrices of ones and zeros, respectively.

All numerical experiments were implemented by using MATLAB R2011b, and on a PC with Intel 3.2GHz CPU, 4GB RAM, Windows 7 platform. The subproblem $\text{AQP}(\mathcal{B}_k)$ is solved by the well-known software MOSEK [20].

The parameters are selected as: $\epsilon = 0.001$, $\sigma = 4$, $\eta = 0.01$; $\bar{t} = 1$ and $\beta = 0.3$ for the first 8 problems in Table I and Max1; $\bar{t} = 0.001$ and $\beta = 0.5$ for Wong 2 and Maxquad; $\bar{t} = 100/11$ and $\beta = 0.4$ for Wong 3; $\bar{t} = 10/3$ and $\beta = 0.3$ for Maxq, Goffin and MXHILB. The update of t_k is slightly more sophisticated than stated in the algorithm, since a heuristic procedure similar to the one in [12] is applied. The index h in Step 4 is selected basically in order $1, \dots, m$, but we first test the indices that violate (i.e. (1) holds) in previous iterations, since they are more likely to violate at this time. In addition, we set a maximum value $M = \min\{10n, 50\}$ for the bundle sizes.

The numerical results for problems Wong 2 and Wong 3 are reported in Table II, and the other results are reported in Table III. The notations are: the number of iterations ‘‘NI’’; the number of descent steps ‘‘ND’’; the number of component function evaluations ‘‘N f_i ’’; the approximate optimal value ‘‘ f^* ’’; the number of equivalent objective function evaluations ‘‘N f_f^{eq} ’’, i.e., $N_f^{eq} = Nf_i/m$.

Table I. The data for 13 test problems

Problem	starting point y^1	box constraints
CB2	$(3, 3)^T$	$\ell = (2, 2)^T, u = (4, 4)^T$
CB3	$(3, 3)^T$	$\ell = (2, 0)^T, u = (4, 3)^T$
DEM	$(0.5, -2.5)^T$	$\ell = (0.1, -3)^T, u = (1.1, -2)^T$
QL	$(2, 3)^T$	$\ell = (1.3, 2.5)^T, u = (2.3, 3.5)^T$
LQ	$(1, 1)^T$	$\ell = (\frac{1}{\sqrt{2}} + 0.1, \frac{1}{\sqrt{2}} + 0.1)^T,$ $u = (\frac{1}{\sqrt{2}} + 1.1, \frac{1}{\sqrt{2}} + 1.1)^T$
Mifflin1	$(1.5, 0.5)^T$	$\ell = (1.1, 0.1)^T, u = (2.1, 1.1)^T$
Rosen-Suzuki	$(1, 2.1, -3, -0.9)^T$	$\ell = (-\infty, 1.1, -\infty, -0.9)^T,$ $u = (+\infty, 2.1, +\infty, 0.1)^T$
Shor	2ones (5, 1)	$\ell = (-\infty, 1.1, -\infty, 1.1, -\infty)^T,$ $u = (+\infty, 2.1, +\infty, 2.1, +\infty)^T$
Maxquad	zeros (10, 1)	$\ell = -2\text{ones}(10, 1), u = 2\text{ones}(10, 1)$
Maxq	for $j = 1, \dots, 10$, if $\text{mod}(j, 2)=0$, $y_j^1 = 1.1$, else $y_j^1 = j$; for $j = 11, \dots, 20$, if $\text{mod}(j, 2)=0$, $y_j^1 = 0.1$, else $y_j^1 = -j$	for $j = 1, \dots, 20$, if $\text{mod}(j, 2)=0$, $\ell_j = 0.1, u_j = 1.1$, else $\ell_j = -\infty, u_j = +\infty$
Maxl	the same as Maxq	the same as Maxq
Goffin	for $j = 1, \dots, 25, y_j^1 = 0$, for $j = 26, \dots, 50, y_j^1 = 3$	for $j = 1, \dots, 25, \ell_j = -1, u_j = 1$ for $j = 26, \dots, 50, \ell_j = 2, u_j = 4$
MXHILB	0.1ones (50, 1)	for $j = 1, \dots, 50$, if $\text{mod}(j, 2)=0$, $\ell_j = 0.1, u_j = 1.1$, else $\ell_j = -\infty, u_j = +\infty$

Table II. Numerical results for Wong 2 and Wong 3

Problem	n	m	p	NI	ND	Nf_i	f^*	N_f^{eq}
Wong 2	10	6	3	27	10	102	24.306268	17
Wong 3	20	14	4	83	16	664	92.764791	47

7. Conclusions

In this paper, we have presented an improved partial bundle method for linearly constrained minimax problems. By introducing the partial cutting-planes model and an improved descent test criterion, the number of component function evaluations may be reduced greatly. At each iteration, only one QP subproblem is solved. The proposed method produces a sequence of feasible trial points, and ensures that the objective function is monotonically decreasing on the sequence of stability centers. Global convergence is proved, and limited numerical results show that our method is promising.

Table III. Numerical results for 13 problems

Problem	n	m	p	NI	ND	Nf_i	f^*	N_f^{eq}
CB2	2	3	4	2	2	6	20.000000	2
CB3	2	3	4	2	2	6	16.000000	2
DEM	2	3	4	2	2	6	-2.500000	2
QL	2	3	4	2	2	6	7.940000	2
LQ	2	2	4	2	2	4	-1.311371	2
Mifflin1	2	2	4	2	2	4	3.300000	2
Rosen_Suzulki	4	4	4	23	6	54	-43.853040	14
Shor	5	10	4	40	11	189	23.418955	19
Maxquad	10	5	20	33	9	121	-0.841170	24
Maxq	20	20	20	51	14	496	0.010176	25
Maxl	20	20	20	18	11	269	0.100361	13
Goffin	50	50	100	27	3	1074	25.000000	21
MXHILB	50	50	50	34	11	567	0.001698	11

Acknowledgement

This work was supported by the National Natural Science Foundation (11301095, 11271086), Guangxi Natural Science Foundation (2013GXNSFAA019013, 2014GXNSFFA118001) of China and Innovation Group of Talents Highland of Guangxi Higher School.

REFERENCES

1. I. Averbakh and O. Berman, *Minimax p -traveling salemen location problems on a tree*, Annals of Operations Research, vol. 110, pp. 55-68, 2002.
2. A. Baums, *Minimax method in optimizing energy consumption in real-time embedded systems*, Automatic Control and Computer Sciences, vol. 43, pp. 57-62, 2009.
3. J. F. Bonnans, J. C. Lemaréchal and C. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects, Second ed*, Springer-Verlag, 2006.
4. J. V. Burke, A. S. Lewis and M. L. Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM Journal on Optimization, vol. 15, pp. 751-779, 2005.
5. E. W. Cheney and A. A. Goldstein, *Newton's method for convex programming and Tchebycheff approximations*, Numerische Mathematik, vol. 1, pp. 253-268, 1959.
6. M. Gaudioso, G. Giallombardo and G. Miglionico, *An incremental method for solving convex finite min-max problems*, Mathematics of Operations Research, vol. 31, pp. 173-187, 2006.
7. W. Hare and M. Macklem, *Derivative-free optimization methods for finite minimax problems*, Optimization Methods and Software, vol. 28, pp. 300-312, 2013.
8. W. Hare and J. Nutini, *A derivative-free approximate gradient sampling algorithm for finite minimax problems*, Computational Optimization and Applications, vol. 56, pp. 1-38, 2013.
9. J. B. Jian, C. M. Tang and F. Tang, *A feasible descent bundle method for inequality constrained minimax problems (in Chinese)*, Science China: Mathematics, vol. 45, pp. 2001-2024, 2015.
10. J. E. Kelley, *The cutting-plane method for solving convex programs*, Journal of the society for Industrial and Applied Mathematics, vol. 8, pp. 703-712, 1960.
11. K. C. Kiwiel, *Method of Descent for Nondifferentiable Optimization*, Lecture Note in Mathematics 1133, Springer-Verlag, 1985.
12. K. C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Mathematical Programming, vol. 46, pp. 105-122, 1990.
13. K. C. Kiwiel, *A projection-proximal bundle method for convex nondifferentiable minimization*, In M. Théra et al. (Eds), Ill-posed Variational Problems and Regularization Techniques, Lecture Notes in Econom. Math. Systems 477, pp. 137-150, Berlin: Springer-Verlag, 1999.
14. K. C. Kiwiel, *A proximal bundle method with approximate subgradient linearizations*, SIAM Journal on Optimization, vol. 16, pp. 1007-1023, 2006.
15. K. C. Kiwiel, *A proximal-projection bundle method for Lagrangian relaxation, including semidefinite programming*, SIAM Journal on Optimization, vol. 17, pp. 1015-1034, 2006.

16. C. Lemaréchal, *An extension of Davidon methods to nondifferentiable problems*, Mathematical Programming Study, vol. 3, pp. 95-109, 1975.
17. X. S. Li and S. C. Fang, *On the entropic regularization method for solving min-max problems with applications*, Mathematical Method of Operations Research, vol. 46, pp. 119-130, 1997.
18. G. Liuzzi, S. Lucidi and M. Sciandrone, *A derivative-free algorithm for linearly constrained finite minimax problems*, SIAM Journal on Optimization, vol. 16, pp. 1054-1075, 2006.
19. L. Lukšan and J. Vlček, *Test problems for nonsmooth unconstrained and linearly constrained optimization*, Technical Report No.798, Institute of Computer Science, Academy of Science of the Czech Republic, Prague, 2000.
20. MOSEK *The MOSEK optimization toolbox for MATLAB manual, Version 7.1 (Revision 30)* MOSEK ApS, Denmark, <http://www.mosek.com>, 2015.
21. M. L. Overton, *Algorithms for nonlinear ℓ_1 and ℓ_∞ fitting*, In M.J.D.Powell (Ed.), Nonlinear Optimization, pp. 213-221, 1982.
22. C. M. Tang, J. B. Jian and G. Y. Li, *A proximal-projection partial bundle method for convex constrained minimax problems*, Manuscript, 2015.
23. F. Y. Wang, Y. Yamamoto and M. Yu, *A minimax rule for portfolio selection in frictional markets*, Mathematical Methods of Operations Research, vol. 57, pp. 141-155, 2003.
24. F. S. Wang and K. C. Zhang, *A hybrid algorithm for nonlinear minimax problems*, Annals of Operations Research, vol. 164, pp. 167-191, 2008.
25. F. S. Wang and K. C. Zhang, *A hybrid algorithm for linearly constrained minimax problems*, Annals of Operations Research, vol. 206, pp. 501-525, 2013.
26. G. A. Waston, *The minimax solution of an overdetermined system of nonlinear equations*, Journal of the Institute of Mathematics and its Applications, vol. 23, pp. 167-180, 1979.
27. P. Wolf, *A method of conjugate subgradient for minimizing nondifferentiable function*, Mathematical Programming Study, vol. 3, pp. 145-173, 1975.