International Journal of Operations Research

International Journal of Operations Research Vol. 9, No. 1, 12–26 (2012)

Continuous Min-Max Programs with Semi-Infinite Constraints

Jen-Yen Lin[1,*] **, Soon-Yi Wu**[2] **and Ruey-Lin Sheu**[2]

[1]Department of Applied Mathematics, NCYU, Chiayi, Taiwan

[2]Department of Mathematics, NCKU, Tainan, Taiwan

Received November 2011; *Revised December 2011*; *Accepted January 2012*

Abstract—In this paper, we propose an algorithm for solving a kind of nonlinear programming where the objective is the maximal function of a family of continuous functions and the feasible domain is explicitly made of infinitely many constraints. Our algorithm combines the entropic regularization and the cutting plane method (the Remez-type) to deal with the non-differentiability of the maximal function and the infinitely many constraints respectively. A finite relaxed version, which terminates within a finite number of iterations to give an approximate solution, is proposed to handle the computational issues, including the blow-up problem in the entropic regularization and the global optimization subproblems in the cutting plane method. To justify the efficiency of the inexact algorithm, we also analyze the theoretical error-bound and conduct numerical experiments.

Keywords— Min-max Programs, Semi-infinite Constraints, Entropic Regularization, Cutting Plane Methods.

1. INTRODUCTION

In this paper, we consider the following problem:

$$(P) \quad \begin{array}{ll} \min\limits_{x\in X} & F(x) \\ s.t. & g(s,x) \le 0, \ \forall s \in S \end{array} \tag{1}$$

where $F(x) = \max\limits_{t\in T} f(t,x)$; X is a compact subset of $\mathbb{R}^n$; T and S are compact subsets of $\mathbb{R}^p$; $f: T\times X \to \mathbb{R}$; $g: S\times X \to \mathbb{R}$ are continuous. When the cardinality of both T and S is finite, the program (P) is called the finite min-max programming. When the cardinality of T is uncountable, it is called the continuous min-max programming. If the cardinality of S is uncountable, then (P) is the semi-infinite programming. For convenience, we also let x^* be an optimal solution of the master problem (1) with α^* being the optimal value.

The finite min-max programming has been studied extensively in the literature, see for examples, Dem'yanov (1974), Dem' yanov & Vasilev (1985), Allen et al. (1987), Polak et al. (1982, 1991, 1922), Polak (1987,1994), Sheu & Wu (1999). Since the objective function $\max\limits_{1\le i\le n} f(t_i,x)$, (in this case, $T \triangleq \{t_1,t_2,\cdots,t_n\}$) is nonsmooth, the gradient-based algorithms can not be applied directly. Instead, the subgradients or the $\epsilon -$ subgradients are often used as the search direction in the line search. For the computation of subgradients and $\epsilon -$ subgradients, please see Dem'yanov (1974), Dem'yanov & Vasilev (1985), Bertsekas (2002), Bertsekas et al. (2003). On the other hand, the non-differentiable points can be regularized (approximated) by smooth functions so that the gradient-based methods become applicable. Such a method is called the *smoothing method*. In particular, Fang et al. (1997) proposed the following entropic regularization

$$F_p(x) = \frac{1}{p}\ln\sum_{i=1}^{n}\exp\{pf(t_i,x)\} \tag{2}$$

The approximate functions $F_p(x),\ p>0$ are smooth and due to the following uniform estimation

$$0 \le F_p(x) - F(x) \le \frac{\ln(m)}{p} \tag{3}$$

[*] Corresponding author's email: jylin@mail.ncyu.edu.tw

$F_p(x)$ converge to $\max_{1\le i\le n} f(t_i,x)$ uniformly as p approaches to ∞ . Therefore, minimizing $F_p(x)$ for each p creates a sequence of iterated points x_p in which there is a subsequence converging to an optimal solution of the original finite min-max problem.

A generalization of the finite min-max programming is to consider an infinite compact set T in the objective function. There are two classical approaches for solving it: the integral-type and the Remez-type. Polak et al. (1992) solved the subproblem

$$\min_{x\in C(\alpha_k)} p(\alpha_k,x)=\int_0^1 \frac{1}{\alpha_k - f(t,x)}dt \tag{4}$$

where α_k is the objective value at the k^{th} iteration and $C(\alpha_k)=\{x\in X \mid \max_{t\in T} f(t,x)<\alpha_k\}$. This is a kind of barrier function method. However, to ensure the barrier effect that $p(\alpha_k,x)$ should approach to infinity for points near the boundary of $C(\alpha_k)$, a uniform Lipschitz condition on the functions $f(t,x)$ is required. Another integral-type method is to generalize the idea of the entropic regularization (2) via integration:

$$\hat{F}_p(x)=\frac{1}{p}\ln\int_T \exp\{pf(t,x)\}dt. \tag{5}$$

However, the uniform estimation (3) for the finite min-max programming can not be similarly extended. In fact, only the right inequality in (3) remains valid:

$$\hat{F}_p(x)\le F(x)+\frac{\ln(m)}{p}, \tag{6}$$

so that the convergence of $\hat{F}_p(x)$ to $F(x)$ can be established merely under restrictive conditions. See Wu & Fang (1996). Then, the theoretical assumptions together with the numerical difficulty in treating the integral-type objective function (4) or (5) become a serious limitation on the capacity of applying the integral-type method for the continuous min-max programming.

In contrast to the integral-type method which deals with the entire set of functions $\{f(t,x)\}_{t\in T}$ at each iteration, the Remez-type method, also known as the method of outer approximations (Polak, 1994) or the exchange method in semi-infinite programming (Hettich & Kortanek, 1993), (Reemtsen & Gorner, 1998) is a cutting plane type of approach which considers only a finite subset T_m of T at a time and let T_m grow gradually to approximate T asymptotically. This reduces the continuous min-max problem into a sequence of finite ones, so that the entropy regularization (2) rather than (5) is feasible. See Sheu & Lin (2004) for the detail. With the help of the uniform estimation(3), the convergence of the Remez-type method can be proved with less restriction than that of the integral-type, but the numerical bottleneck appears (i) in the update for T_m which requires to solve a *global* optimization subproblem; and (ii) the possible blow-up caused by the exponential function in (2) when T grows. We shall take care of these issues later in the paper by proposing a relaxed version.

The program (P) is a semi-infinite programming(SIP) when the cardinality of S is uncountable. SIP aries in many areas: Mathematics, Engineering, Management Science, etc. Practical examples of SIP can be found in Chapter 2 of Hettich & Kortanek (1993), where the robot trajectory planning, Chebyshev approximation, minimal cost control of air pollution, statistical regression experimental design, and many others were discussed. Polak (1987) also had some SIP examples from the optimal control and the engineering design.

In these applications, we focus on the rational Chebyshev approximation which consider the problem of approximating a given real valued function $a(t)$, on an interval $[t_1,t_2]$, by a rational function $\frac{p(t,x)}{q(t,x)}$ where the integers m_1 and m_2 are given, $p(t,x)=x_{m_1+1}t^{m_1}+x_{m_1}t^{m_1-1}+\cdots+x_1$ and $q(t,x)=x_{m_1+m_2+2}t^{m_2}+x_{m_1+m_2+1}t^{m_2-1}+\cdots+x_{m_1+2}$. We wish to find an optimal solution of

$$\inf_{x\in X}\max_{t\in[t_1,t_2]}\left|a(t)-\frac{p(t,x)}{q(t,x)}\right| \tag{7}$$

where

$$X=\left\{x\in\mathbb{R}^{m_1+m_2+2}\left|\max_{1\le j\le m_2+1}\right| x_{m_1+1+j}\,|=1, q(t,x)>0,\forall t\in[t_1,t_2]\right\}.$$

The normalization condition $\max_{1\le j\le m_2+1} | x_{m_1+1+j} |= 1$ is necessary, because the expression $\frac{p(t,x)}{q(t,x)}$ is homogeneous in the required coefficients x_i , $i=1,2,\cdots,m_1+m_2+2$. The semi-infinite constraints $q(t,x)>0,\forall t\in[t_1,t_2]$ is used to avoid that the rational function $\dfrac{p(t,x)}{q(t,x)}$ has singularity. The program (7) is a kind of continuous min-max programs with semi-infinite constraints.

Numerical methods for solving SIP has been summarized in Reemtsen & Gorner (1998). These algorithms can be categorized into three types: Discretization, Exchange method/Remez type method, Methods based on the local reduction. The essence of the three types of methods is to approximate the feasible set

$g(s,x) \le 0, \forall s \in S$

of (P) by imposing only finitely many constraints

$g(s_i,x) \le 0, \forall s_i \in S_l = \{s_1, s_2, \cdots, s_l\}$.

The discretization method seems to be the simplest in the idea, but the number l used to be very large. Methods based on the local reduction requires only local searches. Its convergence is fast, but the requirement to find all local maxima $\overline{s}$ that $g(\overline{s},\overline{x}) = \max_{s \in S} g(s,\overline{x})$ is too expensive, where $\overline{x}$ is the previous iterate point. We are most interested in the exchange method/Remez-type approach as it can be used to incorporate with the Remez-type method that we have discussed for the min-max objective function above.

One way to solve the main program (1) is to move the function $\max_{t \in T} f(t,x)$ to the constraints as the following form

$$\begin{aligned} \min_{(\alpha,x)} \quad & \alpha \\ s.t. \quad & g(s,x) \le 0, \forall s \in S \\ & f(t,x) \le \alpha, \forall t \in T \\ & \alpha \in \mathbb{R}, x \in X \end{aligned} \tag{8}$$

and solve it by SIP methods. But sometimes it is not a good transformation. For example, if we use Remez-type method to solve the program(8), then, in the subproblem of each iteration, we need to consider finite subset T_m of T and the constraints

$f(t,x) \le \alpha, \forall t \in T_m$.

When the cardinal number of T_m is increased, the number of constraints of next subproblem is increased. Hence the next subproblem is harder to be solved. On the contrary, if we use entropy regularization

$$\frac{1}{p} \ln \sum_{t \in T_m} \exp\{pf(t,x)\} \tag{9}$$

to approximate the function $\max_{t \in T_m} f(t,x)$, then the cardinal number of T_m is increased by 1 only contribute to add one more term to the function(9). This is another benefit of entropy regularization.

This paper is aimed to solve (P) by applying the Remez-type method to both the continuous min-max objective function and also the infinitely many constraints. In other words, at each iteration we only face a finite min-max objective function (specified by the set T_m) with a finite number of constraints (specified by the set S_l) and solve it via the entropic regularization (by parameters p). Normally in such an approach, the next update for S_l has to wait until the completion of all necessary updates for T_m and p, but our algorithm allows to update T_m, S_l, and p in any order, depending upon the progress of the algorithm. We shall show in Section 2 that the idea not only increases the best efficiency, but also retains the convergence. In Section 3, we propose a finite inexact version which replaces the *most violated index* by *any violated one* to update the T_m and the S_l, and thus avoids the necessity to solve the global optimization subproblems. As a result, this relaxation has a nice finite convergence property with a controllable error. Finally in Section 4, we implement the algorithm and give some numerical examples to show the practice of our algorithm.

2. AN INEXACT ALGORITHM AND ITS CONVERGENCE PROOF

Let $T_m = \{t_1, t_2, \ldots, t_m\}$ be a finite subset of T and $S_l = \{s_1, s_2, \cdots, s_l\}$ be a finite subset of S. Based on T_m and S_l, we consider the following finite min-max programming problem with a finite number of inequality constraints:

$$(P_{m,l}) \quad \begin{aligned} \min_{x \in X} \quad & \max_{t \in T_m} f(t,x) \\ s.t. \quad & g(s,x) \le 0, \ \forall s \in S_l \end{aligned} \tag{10}$$

Define the objective function of (10) as

$$F^m(x) \triangleq \max_{t\in T_m} f(t,x) \tag{11}$$

Suppose (10) is solved with $x^*_{m,l}$ being an optimal solution and $\alpha^*_{m,l} = \max_{t\in T_m} f(t,x^*_{m,l})$ being the optimal value. To find the minimum of $F^m(x)$ one must face its non-differentiability and the way we handle it is to approximate $F^m(x)$ by the smooth functions:

$$F^m_p(x) = \frac{1}{p}\ln\sum_{t\in T_m}\exp\{pf(t,x)\} \tag{12}$$

The uniform estimation (3) under the current notation is expressed as

$$0 \le F^m_p(x) - F^m(x) \le \frac{\ln(m)}{p}, \forall x \in X \tag{13}$$

Then we may consider the following entropic-regularized version:

$$(P_{m,l,p}) \begin{array}{ll} \min\limits_{x\in X} & F^m_p(x) \\ s.t. & g(s,x) \le 0,\ \forall s \in S_l \end{array} \tag{14}$$

with the definition of a $(\gamma - \eta)$ *optimal solution* to measure the quality of the entropic approximate solution.

Definition 1.

We call x is a $(\gamma,\eta) -$ optimal solution of (1), if

$$|\max_{t\in T} f(t,x) - \alpha^*| \le \gamma$$

and

$$\max_{s\in S} g(s,x) \le \eta.$$

Proposition 2.

Let $\mathrm{x}^{m,l}_p$ be an optimal solution of (14) satisfying $\max_{s\in S} g(s,\mathrm{x}^{m,l}_p) \le 0$ and $F^m_p(\mathrm{x}^{m,l}_p) \ge F(\mathrm{x}^{m,l}_p)$, then $F(\mathrm{x}^{m,l}_p) \le F(x^*) + \frac{\ln m}{p}$. In other words, $\mathrm{x}^{m,l}_p$ is a $\left(\frac{\ln(m)}{p},0\right) -$ optimal solution of (1).

Proof.

Since x^* is feasible to (1), it must also be feasible to (14). This implies that

$$F^m_p(\mathrm{x}^{m,l}_p) \le F^m_p(\mathrm{x}^*). \tag{15}$$

Using the uniform estimation (13), we have

$$0 \le F^m_p(\mathrm{x}^*) - F^m(\mathrm{x}^*) \le \frac{\ln(m)}{p}. \tag{16}$$

Combining (15) with (16), we get

$$F(\mathrm{x}^{m,l}_p) \le F^m(\mathrm{x}^*) + \frac{\ln(m)}{p} \le F(\mathrm{x}^*) + \frac{\ln(m)}{p}.$$

By $\max_{s\in S} g(s,\mathrm{x}^{m,l}_p) \le 0$, $\mathrm{x}^{m,l}_p$ is a feasible solution of (1). Therefore, $\mathrm{x}^{m,l}_p$ is a $\left(\frac{\ln(m)}{p},0\right) -$ -optimal solution of (1). ■

However, if $\mathrm{x}^{m,l}_p$ is not a good approximation in the sense of $(\gamma,0)$-optimality, it is either not feasible or violates the condition $F^m_p(\mathrm{x}^{m,l}_p) \ge F(\mathrm{x}^{m,l}_p)$. In the former case that $\max_{s\in S} g(s,\mathrm{x}^{m,l}_p) > 0$, there is an $s_{l+1} \in S$ such that $g(s_{l+1},\mathrm{x}^{m,l}_p) = \max_{s\in S} g(s,\mathrm{x}^{m,l}_p)$ and $s_{l+1} \notin S_l$. It is thus reasonable to add the s_{l+1} to S_l, which forms S_{l+1}. In the latter case that $F^m_p(\mathrm{x}^{m,l}_p) < F(\mathrm{x}^{m,l}_p)$, the following inequality

$$F(\mathrm{x}^{m,l}_p) > F^m_p(\mathrm{x}^{m,l}_p) \ge F^m(\mathrm{x}^{m,l}_p)$$

implies that there is a $t_{m+1} \in T$ such that $f(t_{m+1},x^{m,l}_p) = F(x^{m,l}_p) > F^m(x^{m,l}_p)$ and $t_{m+1} \notin T_m$. We then add t_{m+1} to T_m and obtain T_{m+1}. The updates for S_l and T_m constitute the backbone of our algorithm as follows.

Algorithm 1

Initial Given prescribed numbers $\delta > 0$, $\epsilon > 1$, $p_1 \geq 0$ and two finite subsets $T_1 = \{t_1\} \subset T$ and $S_1 = \{s_1\} \subset S$. Let $m = 1$, $l = 1$ and $p = p_1$.

Step 1. Solve the following subproblem $(P_{m,l,p})$

$$(P_{m,l,p}) \begin{array}{ll} \min\limits_{x \in X} & F_p^m(x) \\ s.t. & g(s,x) \leq 0, \ \forall s \in S_l \end{array}$$

to get an optimal solution $\mathrm{x}_p^{m,l}$.

Step 2. If $F(x_p^{m,l}) \leq F_p^m(x_p^{m,l})$, $\max\limits_{s \in S} g(s, x_p^{m,l}) \leq 0$ and $\dfrac{1}{p} \leq \delta$, then set $\overline{m} = m$, $\overline{l} = l$ and $\overline{p} = p$; return $x_p^{m,l}$ as a $\left(\dfrac{\ln(\overline{m})}{\overline{p}}, 0\right)$-optimal solution and STOP.

Step 3. If $F(x_p^{m,l}) > F_p^m(x_p^{m,l})$, find a $t_{m+1} \in T$ such that $f(t_{m+1}, x_p^{m,l}) = F(x_p^{m,l})$ and update $T_{m+1} = T_m \cup \{t_{m+1}\}$. Set $m = m + 1$ and choose $p \geq (\ln m)^2$.

Step 4. If $\max\limits_{s \in S} g(s, x_p^{m,l}) > 0$, find a $s_{l+1} \in S$ such that $g(s_{l+1}, x_p^{m,l}) = \max\limits_{s \in S} g(s, x_p^{m,l})$ and $S_{l+1} = S_l \cup \{s_{l+1}\}$. Set $l = l + 1$.

Step 5. If $\dfrac{1}{p} \geq \delta$, then increase p by $p = p * \epsilon$.

Step 6. Goto Step 1.

According to Proposition 2, when Algorithm 1 stops, we have obtained a $\left(\dfrac{\ln(\overline{m})}{\overline{p}}, 0\right)$-optimal solution with $\dfrac{1}{\overline{p}} \leq \delta$. However, it may not always stop within a finite number of iterations and one of the following situations must happen:

1. The index l is eventually fixed at some $\overline{l}$, but the condition in Step 3 is never violated thereafter.
2. The index m is eventually fixed at some $\overline{m}$ and p is fixed at $\overline{p}$ with $\dfrac{1}{\overline{p}} \leq \delta$, but the condition in Step 4 is never violated thereafter.
3. Algorithm 1 keeps adding points to either T_m or S_l infinitely

In the first case, we use the following theorem to ensure that every cluster point of the sequence generated by Algorithm 1 is an optimal point of the problem (1).

Theorem 3.

Suppose that l is eventually fixed at some $\overline{l}$ and that the condition in Step 3 is never violated thereafter. Then Algorithm 1 generates an infinite sequence $\{x_p^{m,\overline{l}}\}$ $(m, p \to \infty)$ in X of which every cluster point is an optimal point of (1).

Proof.

Under the condition of Theorem 3, the points generated after $\overline{l}$ by Algorithm 1 are feasible to (1) but the index m is increased without a bound by Step 3. In this case, p is increased to infinity at a speed at least $(\ln m)^2$. Algorithm 1 then behaves like the IER Algorithm in Sheu & Lin (2004) which solves the continuous min-max problem over a fixed domain. The convergence proof then follows from Theorem 2.1 in Sheu & Lin (2004). ■

In the second case, we prove that every cluster point of the sequence generated by Algorithm 1 is a near-optimal point of the problem (1) with a known error estimation.

Theorem 4.

Suppose that m is eventually fixed at some $\overline{m}$ and p is fixed at $\overline{p}$ with $\dfrac{1}{\overline{p}} \leq \delta$, but the condition in Step 4 is never violated thereafter. Then Algorithm 1 generates an infinite sequence $\{x_{\overline{p}}^{\overline{m},l}\}_{l=1}^{\infty}$ in X of which every cluster point of $\{x_{\overline{p}}^{\overline{m},l}\}_{l=1}^{\infty}$ is a $\left(\dfrac{\ln(\overline{m})}{\overline{p}}, 0\right)$-optimal solution of (1).

Proof.

Let $\bar{x}$ be any cluster point of $\{x_{\bar{p}}^{\bar{m},l}\}$. Since the condition in Step 4 is never violated from a certain iteration on, the index l is increased to infinity. As S is compact, there exist convergent subsequence $\{x_{\bar{p}}^{\bar{m},l_k}\}$ and $\{s_{l_k+1}\}$ such that $\lim_{k\to\infty} x_{\bar{p}}^{\bar{m},l_k} = \bar{x}$ and $\lim_{k\to\infty} s_{l_k+1} = \bar{s}$ where $\bar{s}$ is a cluster point of $\{s_{l_k+1}\}$. Since, for all sufficiently large l_k,

$$F(x_{\bar{p}}^{\bar{m},l_k}) \leq F_{\bar{p}}^{\bar{m}}(x_{\bar{p}}^{\bar{m},l_k}),$$

we have, by Proposition 2, that

$$F(x_{\bar{p}}^{\bar{m},l_k}) \leq F(x^*) + \frac{\ln(\bar{m})}{\bar{p}}. \tag{17}$$

Taking the limit on both sides of inequality (17) and by the continuity of F, we get

$$F(\bar{x}) \leq F(x^*) + \frac{\ln(\bar{m})}{\bar{p}}.$$

Now we show that $\bar{x}$ is feasible, or equivalently, we have to show that $g(\tilde{s},\bar{x}) \leq 0$ where $\tilde{s} \in \arg\max_{s\in S} g(s,\bar{x})$. Since $s_{l_k+1} \in \arg\max_{s\in S} g(s, x_{\bar{p}}^{\bar{m},l_k})$, the following inequality holds:

$$g(\tilde{s}, x_{\bar{p}}^{\bar{m},l_k}) \leq g(s_{l_k+1}, x_{\bar{p}}^{\bar{m},l_k}),$$

which, when taking $l_k \to \infty$, leads to

$$g(\tilde{s},\bar{x}) \leq g(\bar{s},\bar{x}).$$

We claim that $g(\bar{s},\bar{x}) \leq 0$. Suppose, on the contrary, that $g(\bar{s},\bar{x}) > 0$. There exists some positive integer N such that $g(s_{l_N},\bar{x}) > 0$. Since $x_{\bar{p}}^{\bar{m},l_k}$ converges to $\bar{x}$ as $l_k \to \infty$, there is another positive integer $\bar{N} > N$ such that $g(s_{l_N}, x_{\bar{p}}^{\bar{m},l_{\bar{N}}}) > 0$. However, this is a contradiction as $s_{l_N} \in S_{l_{\bar{N}}}$ and $x_{\bar{p}}^{\bar{m},l_{\bar{N}}}$ satisfies that $g(s, x_{\bar{p}}^{\bar{m},l_{\bar{N}}}) \leq 0$ for all $s \in S_{l_{\bar{N}}}$. ■

Finally in the third case, we have the following theorem.

Theorem 5.

Suppose Algorithm 1 generates an infinite sequence $\{x_p^{m,l}\}(l \to \infty, m \to \infty, p \geq (\ln(m))^2)$ *in* X*, then every cluster point of* $\{x_p^{m,l}\}$ *is a global minimizer of (1).*

Proof.

Since $m \to \infty$ and $p \geq (\log(m))^2)$, we have $p \to \infty$ and $\frac{\ln m}{p} \to 0$. Let $\bar{x}$ be any cluster point of $\{x_p^{m,l}\}$. Then, by the compactness of X, T and S, there exist convergent subsequences $\{x_{p_k}^{m_k,l_k}\}$, $\{t_{m_k+1}\}$ and $\{s_{l_k+1}\}$ which converge to $\bar{x}$, $\bar{t}$ and $\bar{s}$, respectively, as $l_k \to \infty$, $m_k \to \infty$. Also (13) implies that

$$F^{m_k}(x_{p_k}^{m_k,l_k}) \leq F_{p_k}^{m_k}(x_{p_k}^{m_k,l_k})$$

and

$$F_{p_k}^{m_k}(x) \leq F^{m_k}(x) + \frac{\ln(m_k)}{p_k}, \quad \forall x \in \mathrm{X}.$$

Since $x_{p_k}^{m_k,l_k}$ is an optimal solution of (P_{m_k,l_k,p_k}), we have

$$\begin{aligned} F^{m_k}(x_{p_k}^{m_k,l_k}) &\leq F_{p_k}^{m_k}(x_{p_k}^{m_k,l_k}) \\ &\leq F_{p_k}^{m_k}(x), \forall x \in X \\ &\leq F^{m_k}(x) + \frac{\ln(m_k)}{p_k} \\ &\leq F(x) + \frac{\ln(m_k)}{p_k}, \quad \forall x \in X \end{aligned}. \tag{18}$$

Moreover,

$$f(t_{m_k+1}, x_{p_k}^{m_k,l_k}) = F(x_{p_k}^{m_k,l_k}) \geq F^{m_k}(x_{p_k}^{m_k,l_k}) \geq f(t_{m_k}, x_{p_k}^{m_k,l_k}). \tag{19}$$

By the continuity of $f(t,x)$ on $T \times X$, both sides of the inequality (19) tend to $f(\overline{t},\overline{x})$, which implies $F^{m_k}(x_{p_k}^{m_k,l_k}) - F(x_{p_k}^{m_k,l_k}) \to 0$ as $k \to \infty$. As $F(x_{p_k}^{m_k,l_k})$ converges to $F(\overline{x})$, we have

$$F^{m_k}(x_{p_k}^{m_k,l_k}) \to F(\overline{x}) . \tag{20}$$

Combining (20) with (18), we obtain

$$F(\overline{x}) \le F(x), \ \forall x \in X .$$

Finally, that $\overline{x}$ is feasible to (1) follows the same proof of Theorem 4. ■

3. A RELAXED VERSION

The purpose of this section is to enhance the effectiveness of Algorithm 1 by resolving three numerical issues in Algorithm 1. They are

1. a possible blow-up caused by the exponential term in the objective function of $(P_{m,l,p})$;
2. finding $t_{m+1} \in \arg\max_{t\in T} f(t, x_p^{m,l})$, a global optimization subproblem;
3. finding $s_{l+1} \in \arg\max_{s\in S} g(s, x_p^{m,l})$, again a global optimization subproblem.

Notice that the core of Algorithm 1 is to minimize $F_p^m(x)$, which requires to compute the function values of $F_p^m(x)$. As $F_p^m(x)$ involves an exponential term which is sensitive to the values of p and $f(t,x)$, when $f(t,x)$ is positive and p is large, $F_p^m(x)$ can be easily numerically unbounded. However, the difficulty can be eased by a proper shift of

$$\begin{array}{lll} \widetilde{(P_{m,l,p})} & \min_{x\in X} & \frac{1}{p}\ln \sum_{t\in T_m} \exp\{p[f(t,x) - \tilde{\alpha}_m]\} \\ & s.t. & g(s,x) \le 0, \ \forall s \in S_l \end{array} \tag{21}$$

where $\tilde{\alpha}_m = \max_{t\in T_m} f(t, x_p^{m-1,l})$ and let us define

$$\tilde{F}_p^m(x) = \frac{1}{p}\ln\{\sum_{t\in T_m} \exp\{p[f(t,x) - \tilde{\alpha}_m]\}\}$$

A simple computation shows that

$$\tilde{F}_p^m(x) = F_p^m(x) - \tilde{\alpha}_m . \tag{22}$$

so that minimizing $\tilde{F}_p^m(x)$ and minimizing $F_p^m(x)$ make the same optimal solution. However, by the shift of $\tilde{\alpha}_m$, the previous iterate point $x_p^{m-1,l}$ becomes a ready-to-use initial point for computing its next iterate point $x_p^{m,l}$ since the computation of $\tilde{F}_p^m(x_p^{m-1,l})$ then consists of only the exponentials of negative values. In practice, the sequence $\{x_p^{m,l}\}$ normally converges to a limit $\overline{x}$ in the compact set X as m and p tend to the infinity. In the early stage of the convergence when m and p are small, the difficulty for computing the exponentials are generally not too serious. When p gets bigger, since at the same time $x_p^{m-1,l}$ is also close to $x_p^{m,l}$, we can expect that the computational concerns for $\exp\{p(f(t,x) - \tilde{\alpha}_m)\}$ should be relaxed due to $\tilde{\alpha}_m$. This observation will be confirmed by our experiments in Section 4.

The other two computational issues involve the global optimization subproblems for finding t_{m+1} and s_{l+1}. We have seen from Algorithm 1 that, if either $F(x_p^{m,l}) > F_p^m(x_p^{m,l})$ or $\max_{s\in S} g(s, x_p^{m,l}) > 0$, the algorithm can not stop since, respectively, either the objective value $F(x_p^{m,l})$ is not small enough or the point $x_p^{m,l}$ is not feasible yet. By Theorem 5, it guarantees the global convergence of $\{x_p^{m,l}\}$ in the limit provided the *most violated indices* (or the *deepest cut* in the cutting plane slogan) $t_{m+1} \in \arg\max_{t\in T} f(t, x_p^{m,l})$ and $s_{l+1} \in \arg\max_{s\in S} g(s, x_p^{m,l})$ are generated to update T_m and S_l respectively. This creates two computational problems. First, the global optimization problems are in general difficult to solve. Secondly, generating t_{m+1} and s_{l+1} might become an infinite process. Though the convergence to an optimal solution holds eventually in the limit by Theorem 5, yet in a practical situation only a finite execution time is allowed.

We then propose a natural alternative for finding t_{m+1} and s_{l+1}. The idea is to find *any violated index* rather than *the most violated one*. Specifically, given two positive values γ and η as the tolerances, we suggest to find any $\tilde{t}_{m+1}$ such that

$$f(\tilde{t}_{m+1}, x_p^{m,l}) > F_p^m(x_p^{m,l}) + \gamma \tag{23}$$

and to find any $\tilde{s}_{l+1}$ such that

$$g(\tilde{s}_{l+1}, x_p^{m,l}) > \eta \,. \tag{24}$$

If there are no such $\tilde{t}_{m+1}$ and $\tilde{s}_{l+1}$, we should stop the algorithm and report the current iterate point as a near optimal solution for (1). It is going to be shown that, generating $\tilde{t}_{m+1}$ and $\tilde{s}_{l+1}$ must be a finite process regardless the points selected in (23) and (24). Moreover, the quality of the final approximate solution is controllable. For this purpose, we have the following propositions and theorems:

Algorithm 2

Initial Given two prescribed numbers $\delta > 0$, $\gamma > 0$, $\eta > 0$, $\epsilon > 1$, $p_1 \geq 0$ and two finite subsets $\tilde{T}_1 = \{\tilde{t}_1\} \subset T$, $\tilde{S}_1 = \{\tilde{s}_1\} \subset S$. Let $m = 1$, $l = 1$, $p = p_1$ and $\tilde{\alpha}_1 = \max_{t \in \tilde{T}_1} f(t, x)$.

Step 1. Solve

$$\begin{array}{lll} (\widetilde{P_{m,l,p}}) & \min_{x \in X} & \frac{1}{p} \ln \sum_{t \in \tilde{T}_m} \exp\{p[f(t,x) - \tilde{\alpha}_m]\} \\ & s.t. & g(s,x) \leq 0, \ \forall s \in \widetilde{S}_l \end{array}$$

to get an optimal solution $x_p^{m,l}$.

Step 2. If $F(x_p^{m,l}) \leq \tilde{F}_p^m(x_p^{m,l}) + \tilde{\alpha}_m + \gamma$, $\max_{s \in S} g(s, x_p^{m,l}) \leq \eta$ and $\frac{1}{p} \leq \delta$, then set $\overline{m} = m$, $\overline{l} = l$ and $\overline{p} = p$; return $x_{\overline{p}}^{\overline{m},\overline{l}}$ as a $(\frac{\ln \overline{m}}{\overline{p}} + \gamma, \eta) -$ optimal solution and STOP.

Step 3. Find $\tilde{t}_{m+1} \in T$ such that $f(\tilde{t}_{m+1}, x_p^{m,l}) > \tilde{F}_p^m(x_p^{m,l}) + \tilde{\alpha}_m + \gamma$. Set $\tilde{T}_{m+1} = \tilde{T}_m \cup \{\tilde{t}_{m+1}\}$; $\tilde{\alpha}_{m+1} = \max_{t \in \tilde{T}_{m+1}} f(t, x_p^{m,l})$; $m = m + 1$, and choose $p \geq (\ln m)^2$.

Step 4. Find $\tilde{s}_{l+1} \in S$ such that $g(\tilde{s}_{l+1}, x_p^{m,l}) > \eta$. Set $\tilde{S}_{l+1} = \tilde{S}_l \cup \{\tilde{s}_{l+1}\}$ and $l = l + 1$.

Step 5. If $\frac{1}{p} \geq \delta$, increase p by $p = p * \epsilon$.

Step 6. Go to Step 1.

In the remaining of this section, we prove that Algorithm 2 always stops in a finite number of steps. First, we discuss that Algorithm 2 will run Step 3 for at most a finite number of times.

Proposition 6.

Algorithm 2 can only run Step 3 for a finite number of steps.

Proof.

Suppose, on the contrary, that Algorithm 2 ran Step 3 infinitely often. Then an infinite sequence $\{x_p^{m,l}\}(m \to \infty, p \geq (\ln m)^2)$ would have been generated. Let $\{x_{p_k}^{m_k,l_k}\}$ and $\{\tilde{t}_{m_k+1}\}$ be convergent subsequences such that $x_{p_k}^{m_k,l_k} \to \overline{x}$ and $\tilde{t}_{m_k+1} \to \overline{t}$ as $k \to \infty$ and $f(\tilde{t}_{m_k+1}, x_{p_k}^{m_k,l_k}) > \tilde{F}_{p_k}^{m_k}(x_{p_k}^{m_k,l_k}) + \tilde{\alpha}_{m_k} + \gamma$. Moreover, $\{F^{m_k}(x)\}_{k=1}^{\infty}$ forms an increasing sequence of functions bounded above by $F(x)$ so that

$$\overline{F}(x) = \lim_{k \to \infty} F^{m_k}(x), \ \forall x \in X$$

is well-defined. Notice that the function $\overline{F}(x)$ depends on $\{\tilde{t}_{m_k+1}\}$. A different choice of $\{\tilde{t}_{m_k+1}\}$ in general gives a distinct $\overline{F}(x)$. Nevertheless, by the Dini theorem (Dieudonne, 1969), $\{F^{m_k}(x)\}_{k=1}^{\infty}$ always converges uniformly to its corresponding continuous function $\overline{F}(x)$ on X. By(22),

$$\begin{aligned} F^{m_k}(x_{p_k}^{m_k,l_k}) &= \tilde{F}^{m_k}(x_{p_k}^{m_k,l_k}) + \tilde{\alpha}_{m_k} \\ &\leq \tilde{F}_{p_k}^{m_k}(x_{p_k}^{m_k,l_k}) + \tilde{\alpha}_{m_k} \\ &\leq \tilde{F}_{p_k}^{m_k}(x) + \tilde{\alpha}_{m_k} \\ &\leq F^{m_k}(x) + \frac{\ln m_k}{p_k} \\ &\leq \overline{F}(x) + \frac{\ln m_k}{p_k}. \end{aligned} \tag{25}$$

Moreover, by the uniform convergence of $\{F^{m_k}(x)\}$ to $\overline{F}(x)$ and by the continuity of $\overline{F}(x)$, the following triangular inequality

$$| F^{m_k}(x_{p_k}^{m_k,l_k}) - \overline{F}(\overline{x}) | \leq | F^{m}(x_{p_k}^{m_k,l_k}) - \overline{F}(x_{p_k}^{m_k,l_k}) | + | \overline{F}(x_{p_k}^{m_k,l_k}) - \overline{F}(\overline{x}) |$$

shows that

$$F^{m_k}(x_{p_k}^{m_k,l_k}) \to \overline{F}(\overline{x}), \text{ as } m,p \to \infty$$

Passing the limit to both sides of (25), we obtain $\overline{F}(\overline{x}) \leq \overline{F}(x)$, indicating that $\overline{x}$ minimizes $\overline{F}(x)$ over X. Although $\overline{x}$ is in general not the global minimize of $F(x)$, the fact that $f(\overline{t},\overline{x}) = \overline{F}(\overline{x})$ as shown below, and also the way we choose $\tilde{t}_{m_k+1}$ in Step 3 make Algorithm 2 a finite convergence procedure. To see this specifically, we first notice that

$$\begin{aligned} f(\tilde{t}_{m_k+1}, x_{p_k}^{m_k,l_k}) &> \tilde{F}_{p_k}^{m_k}(x_{p_k}^{m_k,l_k}) + \tilde{\alpha}_{m_k} + \gamma \\ &\geq F^{m_k}(x_{p_k}^{m_k,l_k}) + \gamma \\ &\geq f(\tilde{t}_{m_{k-1}+1}, x_{p_k}^{m_k,l_k}) + \gamma. \end{aligned} \tag{26}$$

In the limit, since $f(t,x)$ is continuous, we have both $f(\tilde{t}_{m_k+1}, x_{p_k}^{m_k,l_k})$ and $f(\tilde{t}_{m_{k-1}+1}, x_{p_k}^{m_k,l_k})$ tend to $f(\overline{t},\overline{x})$ such that (26) becomes

$$f(\overline{t},\overline{x}) \geq \overline{F}(\overline{x}) + \gamma \geq f(\overline{t},\overline{x}) + \gamma$$

which is a contradiction. Therefore, Algorithm 2 will only run Step 3 for a finite number of steps. ■

In a similar fashion, we have

Proposition 7.

Algorithm 2 can only run Step 4 for a finite number of steps.

Proof.

Suppose on the contrary that Algorithm 2 ran Step 4 for infinitely many times and generated an infinite sequence $\{x_p^{m,l}\}(l \to \infty)$. Let $\{x_{p_k}^{m_k,l_k}\}$ and $\{\tilde{s}_{l_k+1}\}$ be convergent subsequences such that $x_{p_k}^{m_k,l_k} \to \overline{x}$ and $\tilde{s}_{l_k+1} \to \overline{s}$ as $l_k \to \infty$ and $g(\tilde{s}_{l_k+1}, x_{p_k}^{m_k,l_k}) > \eta$. Moreover,

$$g(\tilde{s}_{l_k+1}, x_{p_k}^{m_k,l_k}) > \eta > 0 \geq g(\tilde{s}_{l_{k-1}+1}, x_{p_k}^{m_k,l_k}) \tag{27}$$

since $l_{k-1} + 1 \leq l_k$ so that $\tilde{s}_{l_{k-1}+1} \in \tilde{S}_{l_k}$. Taking the limit at both sides of (27) and by the continuity of $g(s,x)$ on $S \times X$, we again obtain the following contradiction:

$$g(\overline{s},\overline{x}) \geq \eta > 0 \geq g(\overline{s},\overline{x}),$$

which proves the theorem. ■

Combining Proposition 6 and Proposition 7, we have the following theorem.

Theorem 8.

Algorithm 2 must stop in Step 2 after a finite number of executions and obtain $x_{\overline{p}}^{\overline{m},\overline{l}}$ *as a* $\left(\dfrac{\ln \overline{m}}{\overline{p}} + \gamma, \eta\right) -$ *optimal solution of (1).*

Proof.

Suppose, after a sufficiently large number of executions, $\dfrac{1}{p} \leq \delta$ and Algorithm 2 can neither generate $\tilde{t}_{m+1}$ nor $\tilde{s}_{l+1}$ with the final index of p, m and l being set to $\overline{p}$, $\overline{m}$ and $\overline{l}$, respectively. We must have, at this moment,

$$f(t, x_{\overline{p}}^{\overline{m},\overline{l}}) \leq \tilde{F}_{\overline{p}}^{\overline{m}}(x_{\overline{p}}^{\overline{m},\overline{l}}) + \tilde{\alpha}_{\overline{m}} + \gamma, \quad \forall t \in T$$

and

$$g(s, x_{\overline{p}}^{\overline{m},\overline{l}}) \le \eta, \ \ \forall s \in S$$

which implies that $F(x_{\overline{p}}^{\overline{m},\overline{l}}) \le \tilde{F}_{\overline{p}}^{\overline{m}}(x_{\overline{p}}^{\overline{m},\overline{l}}) + \tilde{\alpha}_{\overline{m}} + \gamma$ and $\max_{s\in S} g(s, x_{\overline{p}}^{\overline{m},\overline{l}}) \le \eta$ so that the stopping criterion at Step 2 of Algorithm 2 is certainly satisfied no later than $p = \overline{p}$, $m = \overline{m}$, and $l = \overline{l}$. Without loss of generality, we may assume that $\overline{p}, \overline{m}, \overline{l}$ be the earliest such indices. Then, Algorithm 2 stops at Step 2 with

$$\begin{aligned}
F(x^*) &\le F(x_{\overline{p}}^{\overline{m},\overline{l}}) \\
&\le \tilde{F}_{\overline{p}}^{\overline{m}}(x_{\overline{p}}^{\overline{m},\overline{l}}) + \tilde{\alpha}_{\overline{m}} + \gamma \\
&= F_{\overline{p}}^{\overline{m}}(x_{\overline{p}}^{\overline{m},\overline{l}}) + \gamma \\
&\le F_{\overline{p}}^{\overline{m}}(x^*) + \gamma \\
&\le F^{\overline{m}}(x^*) + \frac{\ln(\overline{m})}{\overline{p}} + \gamma \\
&\le F(x^*) + \frac{\ln(\overline{m})}{\overline{p}} + \gamma.
\end{aligned}$$

In other words,

$$0 \le F(x_{\overline{p}}^{\overline{m},\overline{l}}) - F(x^*) \le \frac{\ln \overline{m}}{\overline{p}} + \gamma\,,$$

which shows that $x_{p}^{\overline{m},\overline{l}}$ is a $\left(\dfrac{\ln \overline{m}}{\overline{p}} + \gamma, \eta\right)$ – optimal solution. ■

4. NUMERICAL EXAMPLES

In this section, we demonstrate some numerical experiments for Algorithm 2. The results were obtained on a PC equipped with Pentium(R) D CPU 3.0GHz/3.0GHz, 2 Gigabyte RAM and MATLAB 2007B. At each iteration, Algorithm 2 needs to

1. solve the subproblem $(\widetilde{P_{m,l,p}})$;
2. find $\tilde{t}_{m+1} \in T$ such that $f(\tilde{t}_{m+1}, x_p^{m,l}) > \tilde{F}_p^m(x_p^{m,l}) + \tilde{\alpha}_m + \gamma$; and
3. find $\tilde{s}_{l+1} \in S$ such that $g(\tilde{s}_{l+1}, x_p^{m,l}) > \eta$.

For the first item, we choose the MATLAB subroutine (*fmincon* in Optimization Toolbox) to solve $(\widetilde{P_{m,l,p}})$. For the second and third items, they are equivalent to the following *feasibility* problem: given a fixed $\tilde{x} \in X$ and $\xi > 0$,
Determine and find whether there is a $\tilde{y} \in Y$ such that $h(\tilde{y}, \tilde{x}) > \xi$. (28)

If $Y = T$, $h(y,x) = f(y,x) - \tilde{F}_p^m(x) - \tilde{\alpha}_m$, $\tilde{x} = x_p^{m,l}$ and $\xi = \gamma$, then the solutions of (28) satisfy the condition in Step 3 of Algorithm 2. Similarly, when $Y = S$, $h(y,x) = g(y,x)$, $\tilde{x} = x_p^{m,l}$ and $\xi = \eta$, the solutions of (28) meet the requirement in Step 4 of Algorithm 2. However, if there is no such $\tilde{y}$, we must be able to determine that problem (28) is infeasible.

Theoretically, a feasibility decision problem is as hard as an optimization problem. In practice, however, when the answer to the feasibility decision problem is ``yes", finding only a feasible solution in general saves a lot of work compared with finding an optimal one. We shall implement a discretization-based Algorithm 3 below to solve (28).

Let us first define some notations. Suppose that $\mathrm{Y} = [a,b]$ is an interval on $\mathbb{R}$ and for a given positive integer N, we define the partition Y_N of Y by

$$Y_N = \left\{ \mathrm{y}_i \mid \mathrm{y}_i = a + i\frac{b-a}{N}, i = 0,1,\cdots,N \right\}$$

and also the distance from Y_N to Y by

$$dist(\mathrm{Y}, \mathrm{Y}_N) = \max_{\mathrm{y}\in \mathrm{Y}} \min_{y_i \in \mathrm{Y}_N} \| y - y_i \| \ .$$

The idea is to select a feasible point of (28) repeatedly from a sequence of finer and finer partitions Y_N until either a feasible solution is detected or the distance of Y_N to Y becomes small enough.

Algorithm 3

Initial Given $\tilde{x} \in X$ and three prescribed numbers $\xi > 0$, $\epsilon_0 > 0$ and $\epsilon > \epsilon_0$. Select $r \in (0,1)$ and a finite subset Y_N of Y such that $dist(Y, Y_N) \le \epsilon$.

Step 1. If $h(\tilde{\mathrm{y}}, \tilde{x}) = \max_{y_i \in Y_N} h(y_i, \tilde{x}) > \xi$, report $\tilde{y}$ as a feasible solution to (28) and stop.

Step 2. If $\epsilon > \epsilon_0$, set $\epsilon = r\epsilon$ and reproduce Y_N such that $dist(Y, Y_N) \le \epsilon$. Go to step 1.

Step 3. If $\max_{y_i \in Y_N} h(y_i, \overline{x}) \le \xi$ and $\epsilon \le \epsilon_0$, stop and report that no feasible solution to problem (28) was found.

In the following examples, we incorporate Algorithm 2 with Algorithm 3 for finding approximate solutions to (1). Without sacrificing too much solution quality, the combined algorithm usually obtained a satisfactory solution in less execution time. Moreover, we also compare the numerical results with another approach which treats (1) as a pure semi-infinite programming:

$$\begin{aligned} \min_{(\alpha,x)\in\mathbb{R}\times X} \quad & \alpha \\ s.t. \quad & g(s,x) \le 0,\ \forall s \in S_l \\ & f(t,x) \le \alpha,\ \forall t \in T_m \end{aligned} \tag{29}$$

and solves the problem (28) by the following exchange method:

Algorithm 4

Initial Given two prescribed numbers $\delta > 0$, $\gamma > 0$, $\eta > 0$, $\epsilon > 1$, $p_1 \ge 0$ and two finite subsets $T_1 \subset T$, $S_1 \subset S$. Let $m = 1$, $l = 1$ and $p = p_1$.

Step 1. Solve the finite constrained subproblem

$$\begin{aligned} \min_{(\alpha,x)\in\mathbb{R}\times X} \quad & \alpha \\ s.t. \quad & g(s,x) \le 0,\ \forall s \in S_l \\ & f(t,x) \le \alpha,\ \forall t \in T_m \end{aligned}$$

and get an optimal solution $(\alpha_{m,l}, x^*_{m,l})$.

Step 2. If, for all $t \in T$ and $s \in S$, $x^*_{m,l}$ satisfy $f(t, x^*_{m,l}) \le \alpha_{m,l} + \gamma$, $g(s, x^*_{m,l}) \le \eta$, then return $x^*_{m,l}$ as a (γ,η) -optimal solution and STOP.

Step 3. If there exists $\tilde{t}_{m+1} \in T$ such that $f(\tilde{t}_{m+1}, x^*_{m,l}) > \alpha_{m,l} + \gamma$, let $T_{m+1} = T_m \cup \{\tilde{t}_{m+1}\}$ and $m = m + 1$.

Step 4. If there exists $\tilde{s}_{l+1} \in S$ such that $g(\tilde{s}_{l+1}, x^*_{m,l}) > \eta$, let $S_{l+1} = S_l \cup \{\tilde{s}_{l+1}\}$. Set $l = l + 1$.

Step 5. Go to Step 1.

More details about the relation between (1) and (29) can be found in Polak (1994).

Problem 9.

Define

$$X = [-3,3]^n, T = [0,20], S = [0,200]$$

$$f(t,x) = \sum_{i=1}^{n} t(x_i - \sin t)^2$$

$$g(s,x) = \frac{1}{1+s^2} - \sum_{i=1}^{n} x_i s^{i-1}$$

where the dimension n *is a given postive integer.*

Notice that, the objective function $\max_{t\in[0,20]} f(t,x)$ is convex on $\mathbb{R}^n$. When s is fixed, $g(s,x) \le 0$ is a linear constraint. In this example, we choose $n = 5$ and $n = 6$ as the problem dimension. Table 1 shows the two initial solutions for the two problems with $n = 5$ and $n = 6$, respectively. They both are infeasible.

To run the combined algorithm of Algorithm 2 and Algorithm 3, we set initially that $p_0 = 0.5$, $\delta = 1$, $\gamma = 1$, $\eta = 0.01$, $\epsilon = 2.5$, $T_1 = \{0,5,10,15,20\}$ and $S_1 = \{0,50,100,150,200\}$. The solution obtained, according to the parameters, is a $(1, 0.01)$ -optimal solution. Together with the run time, the results are documented in Table 2 for both $n = 5$ and $n = 6$. To know how good the approximate solutions are, we ran Algorithm 4 as a benchmark. In the first two rows of Table 3, we have very tiny tolerances $\gamma = 10^{-5}, \eta = 0$, so that the results with which can be thought to be a *true* optimal solution. As we can see, the solution quality for the combined algorithm is acceptable, but the run time is just 85% and 51% of that used by Algorithm 4 for $n=5$ and $n=6$ respectively. We expect the trend will continue to grow should the dimension of the problem become much larger. Another interesting result is listed at row 3 and row 4 of Table 3 where the parameters γ and η were set the same for the combined algorithm and Algorithm 4. In this case, the two algorithms differ only by the entropic regularization. The results show that the combined algorithm with the entropic regularization is more economical in both cases. Finally, it is interesting to notice that, when the algorithm stops, there provides by Theorem 8 the theoretical error bound $\dfrac{\ln(\overline{m})}{\overline{p}}+\gamma$, which is 1.2293 in this example. However, the actual errors are 0.0357 for $n=5$ and 0.2456 for $n=6$, much smaller than the estimations.

Table 1. Starting points in Example 9

n	x_0	$\max\limits_{s\in[0,200]} g(s,x_0)$	$\max\limits_{t\in[0,200]} f(t,x_0)$
5	(-0.1,-0.1,-0.1,-0.1,-0.1)	$1.60*10^{8}$	101.7207
6	(-0.1,-0.1,-0.1,-0.1,-0.1,-0.1)	$3.21*10^{10}$	122.0648

Table 2. The results of Algorithm 2 on Example 9 with $p_1=0.5$, $\delta=1$, $\gamma=1$, $\eta=0.01$, $\epsilon=2.5$, $T_1=\{0,5,10,15,20\}$ and $S_1=\{0,50,100,150,200\}$

n	k	$\dfrac{\ln(m)}{p}$	$\max\limits_{s\in[0,200]} g(s,x_0)$	$\max\limits_{t\in[0,200]} f(t,x_0)$	cpu(s)
5	21	0.2293	0.0048	125.3946	2.6250
6	24	0.2293	0.0087	141.3999	3.7344

Table 3. The results of Algorithm 4 on Example 9 with $T_1=\{0,5,10,15,20\}$ and $S_1=\{0,50,100,150,200\}$

n	parameters	$\max\limits_{s\in[0,200]} g(s,x_0)$	$\max\limits_{t\in[0,200]} f(t,x_0)$	cpu(s)
5	$\gamma=10^{-5},\eta=0$	$-2.8108*10^{-17}$	125.4303	3.0625
6	$\gamma=10^{-5},\eta=0$	$-1.1102*10^{-16}$	141.6455	7.3281
5	$\gamma=1,\eta=0.01$	0.0048	125.3946	2.7344
6	$\gamma=1,\eta=0.01$	0.0087	141.3999	4.1563

Problem 10.

The rational Chebyshev approximation is to approximate a given continuous function $h(t)$ *on an interval* T *by a rational function* $\dfrac{p(t,\overline{x})}{q(t,\overline{x})}$ *which satisfies*

$$\max_{t\in T}\left|h(t)-\frac{p(t,\overline{x})}{q(t,\overline{x})}\right|=\min_{x\in X}\max_{t\in T}\left|h(t)-\frac{p(t,x)}{q(t,x)}\right| \tag{30}$$

where $\overline{x}\in X$ *and*

$$X=\left\{x\in\mathbb{R}^{m_1+m_2+2}\,\middle|\,\max_{1\le j\le m_2+1}|\,x_{m_1+1+j}\,|=1, q(t,x)>0,\forall t\in[t_1,t_2]\right\}.$$

Let $h(t)=\exp(-t^2)$, $T=[-2,2]$, $m_1=3$ and $m_2=3$. Suppose the solution $\overline{x}=(-0.018420,0.0575825,0.191241,1.070369,0.148322,2.239045,0.275264,1)$ is given, then it produces a rational function

$$\frac{p(t,\overline{x})}{q(t,\overline{x})}=\frac{-0.018420t^3+0.0575825t^2+0.191241t+1.070369}{0.148322t^3+2.239045t^2+0.275264t+1} \tag{31}$$

and its corresponding objective value $\overline{\alpha}$ is

$$\overline{\alpha}=\max_{t\in T}\left|h(t)-\frac{p(t,\overline{x})}{q(t,\overline{x})}\right|=0.112591. \tag{32}$$

With the compact assumption of X and T, by the properties of generalized fractional programs in Crouzeix et al. (1985), α^* is the optimal value of (30) if and only if $G(\alpha^*)=0$ where

$$G(\alpha) \triangleq \min_{x\in X}\max_{t\in T}\left\{\left|h(t)q(t,x)-p(t,x)\right|-\alpha q(t,x)\right\}.$$

Hence we use Algorithm 2 to compute $G(\bar{\alpha})$ and obtain $G(\bar{\alpha})\in\left[\tilde{F}_p^m(x_p^{m,l})-\dfrac{\log(m)}{p},\tilde{F}_p^m(x_p^{m,l})\right]$. Since Algorithm 2 report $\tilde{F}_p^m(x_p^{m,l})=-0.184756\times10^{-3}$ and $\dfrac{\log(m)}{p}=9.520535\times10^{-7}$, the value $G(\bar{\alpha})$ satisfies

$G(\bar{\alpha})\in[-0.184756\times10^{-3}-9.520535\times10^{-7},-0.184756\times10^{-3}]$.

The value $G(\bar{\alpha})$ point out that $\bar{x}$ is very close to the optimal solution of (30), hence the function (31) is a good approximate function to e^{-t^2} on T.

In Figure 1, we draw three functions: e^{-t^2}, 7th order Taylor polynomial $P_7(t)$ and the function (31). Although $P_7(t)$ is very close to the function e^{-t^2} on $[-1,1]$, the distance between P_7 and e^{-t^2} increases quickly. When $t=2$, the value $|P_7(t)-e^{-t^2}|\approx-48$ which is larger than the objective value (32). Similar results also occur on the 15th order Taylor polynomial $P_{15}(t)$. When the Taylor polynomial is chosen, the Taylor polynomial only can approximate the function e^{-t^2} in an interval well. If t is not in this interval, then the error increases quickly. When we observe the function (31), the function (31) not only approximates the function e^{-t^2} on the requested interval $[-2,2]$ well, but also describes the behavior of e^{-t^2} outside the interval $[-2,2]$.

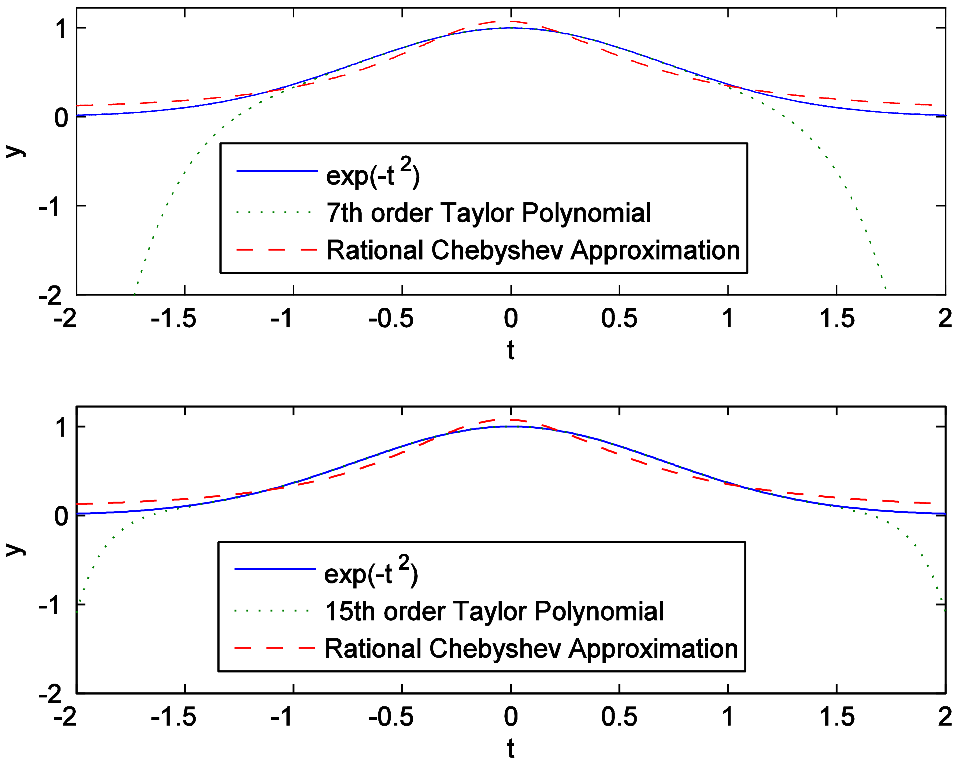

Figure 1. The graph of e^{-t^2}, 7th order Taylor polynomial, 15th order Taylor polynomial and the function (31)

Problem 11.

In this example, we consider the following problem

$$\begin{aligned}
&\min_{x}\max_{t\in[0,2\pi]} && \frac{1}{2}x^tLD(t)L^tx+b^tx+\frac{1}{1+t}\\
&s.t. && \frac{1}{2}x^tU^t\bar{D}(s)Ux+e^tx+\frac{1}{(1+s)^2}\le 0, s\in[0,2\pi]\\
& && \sum_{i=1}^{n}x_i=1, x_i\ge 0
\end{aligned}$$

where the given coefficients $n=20$, $D(t)=(D_{ij}(t))$, $\bar{D}(s)=(\bar{D}_{ij}(s))$, $L=(L_{ij})$ and $U=(U_{ij})$ are defined by

$$\begin{array}{ll}
D_{ii}(t) \sim U(0,0.01), & i \in \{1,2,\cdots,n\}, \\
\bar{D}_{ii}(t) \sim \text{Uniform Distribution}(0,4), & i \in \{1,2,\cdots,n\}, \\
L_{ij} \sim \text{Uniform Distribution}(0,0.01), & i \in \{1,2,\cdots,n\}, j < i, \\
L_{ij} = 0, & i \in \{1,2,\cdots,n\}, j > i, \\
L_{ii} = 1, & i \in \{1,2,\cdots,n\} \\
U_{ij} = 0, & i \in \{1,2,\cdots,n\}, j < i, \\
U_{ij} \sim \text{Uniform Distribution}(0,1), & i \in \{1,2,\cdots,n\}, j > i, \\
U_{ii} = 1, & i \in \{1,2,\cdots,n\}, \\
b_i \sim \text{Uniform Distribution}(0,0.1), & i \in \{1,2,\cdots,n\}, \\
e_i \sim \text{Uniform Distribution}(-60,0), & i \in \{1,2,\cdots,n\}, \\
D_{1,1}(t) = t + \sin t, & \\
\bar{D}_{1,1}(s) = s + \cos s. &
\end{array}$$

Table 4. The results of Algorithm 2 on Example 11 with $T_1 = \{\pi / 2\}$ and $S_1 = \{\pi / 2\}$

	Running Time	Optimal Value	Feasibility
MATLAB subroutine: fseminf	38.6102	1.2628	3.9902×10^{-4}
Algorithm 2	4.4304	1.2628	8.3518×10^{-9}

First, we transfer it into a convex semi-infinite programs by (29) and solve it by the MATLAB subroutine: fseminf. Second, we directly solve Example 11 by Algorithm 2.

Both of these two methods report feasible solutions which have the same objective value, but the running time of Algorithm 2 is less than ones of fseminf by Table 4.

5. CONCLUSION

In this paper, we proposed an inexact algorithm for solving the continuous min-max programming with semi-infinite constraints. The algorithm is based on a two-layer exchange method, reducing everything to just solving a finite min-max programming with a finite number of constraints. However, it is known that the traditional exchange method is very cumbersome by requiring to solve global optimization subproblems, which, in practical implementation, is often relaxed by finding inexact approximations $\tilde{t}_{m+1}$ and $\tilde{s}_{l+1}$. The best contribution of the paper is to provide a theoretical background for such a relaxation that it is indeed a finite convergence algorithm with analyzable error bounds. The primitive numerical results showed that it is very efficient, but we also believe that, based on its inexact nature, the true application of the algorithm lies in some large scale problems.

REFERENCES

1. Allen, E., Helgason, R., Kennington, J. and Shetty, B. (1987). A generalization of Polyak's convergence result for subgradient optimization. *Mathematical Programming*, 37(3): 309-317.
2. Bertsekas, D.P. (2002). *Convexity, Duality and Lagrange Multipliers*, to be published, available from http://www.dim.uchile.cl/~ceimat/sea/apuntes/.
3. Bertsekas, D.P., Nedi'c, A. and Ozdaglar, A.E. (2003). *Convex analysis and optimization*, Athena Scientific, Belmont, MA.
4. Chen, C.-H. and Mangasarian, O.L. (1995). Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 71(1): 51-69.
5. Crouzeix, J.P., Ferland, J.A. and Schaible, S. (1985). An algorithm for generalized fractional programs. *Journal of Optimization Theory and Applications*, 47(1): 35-49.
6. Dem'yanov, V.F. and Malozemov, V.N. (1974). *Introduction to minimax.* John Wiley & Sons, New York.
7. Dem'yanov, V.F. and Vasil'ev, L.V. (1985). *Nondifferentiable optimization.* Optimization Software, Inc..
8. Dieudonne, J. (1969). *Foundations of modern analysis 10-1*, Academic Press, New York.
9. Fang, S.-C., Rajasekera, J.R. and Tsao, H.S.J. (1997). *Entropy optimization and mathematical programming*, Kluwer Academic Publishers, Boston, MA.
10. Fang, S.-C. and Wu, S.-Y. (1996). Solving min-max problems and linear semi-infinite programs. *Computers and Mathematics with Applications*, 32(6): 87-93.
11. Hettich, R. and Kortanek, K.O. (1993). Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35: 380-429.
12. Polak, E. (1987). On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Review*, 29(1): 21-89.

13. Polak, E. (1997). *Optimization: algorithms and consistent approximations*, Springer-Verlag, New York.
14. Polak, E., Higgins, J.E. and Mayne, D.Q. (1992). A barrier function method for minimax problems. *Mathematical Programming*, 54(2): 155-176.
15. Polak, E., Mayne, D.Q. and Higgins, J.E. (1991). Superlinearly convergent algorithm for min-max problems. *Journal of Optimization Theory and Applications*, 69(3): 407-439.
16. Polak, E., Qi, L. and Sun, D. (2001). Second-order algorithms for generalized finite and semi-infinite min-max problems. *SIAM Journal on Optimization*, 11(4): 937-961.
17. Polak, E. and Royset, J.O. (2005). On the use of augmented Lagrangians in the solution of generalized semi-infinite min-max problems. *Computational Optimization and Applications*, 31(2): 173-192.
18. Polak, E. and Tits, A.L. (1982). A recursive quadratic programming algorithm for semi-infinite optimization problems. *Applied Mathematics and Optimization*, 8(4): 325-349.
19. Qi, L. (1993). Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18: 227-244.
20. Reemtsen, R. and Goerner, S. (1998). Numerical methods for semi-infinite programming: A survey. In Reemtsen R. and R'uckmann, J.-J. editors, *Semi-Infinite Programming*, Kluwer Academic publishers.
21. Sheu, R.-L. and Wu, S.-Y. (1999). Combined entropic regularization and path-following method for solving finite convex min-max problems subject to infinitely many linear constraints. *Journal of Optimization Theory and Applications*, 101(1): 167-190.
22. Sheu, R.-L. and Lin, J.-Y. (2004). Solving continuous min-max problems by an iterative entropic regularization method. *Journal of Optimization Theory and Applications*, 121(3): 597-612.
23. Wu, S.-Y., Li, D.-H., Qi, L. and Zhou, G. (2005). An iterative method for solving KKT system of the semi-infinite programming. *Optimization Methods & Software*, 20(6): 629-643.