J Optim Theory Appl (2009) 141: 461–473
DOI 10.1007/s10957-008-9473-4

An Algorithm for the Global Optimization of a Class of Continuous Minimax Problems

P. Parpas · B. Rustem

Published online: 26 November 2008

Abstract We propose an algorithm for the global optimization of continuous minimax problems involving polynomials. The method can be described as a discretization approach to the well known semi-infinite formulation of the problem. We proceed by approximating the infinite number of constraints using tools and techniques from semidefinite programming. We then show that, under appropriate conditions, the SDP approximation converges to the globally optimal solution of the problem. We also discuss the numerical performance of the method on some test problems.

Keywords Worst case analysis · Continuous minimax algorithms · Semidefinite programming · Global optimization

1 Introduction

Many decision models can be formulated as continuous minimax problems. The minimax framework injects robustness into the model. It is a tool that one can use to perform worst–case analysis, and it can provide considerable insight into the decision process. It is frequently used alongside other methods such as expected value optimization in order to identify extreme scenarios and strategies that might provide cover under such scenarios. Despite its importance and usefulness, there are very few algorithms that can reliably solve continuous minimax problems. To the authors knowledge there are no algorithms that can compute the global optimum of such problems. The aim of this paper is to propose such an algorithm, analyze its convergence properties, and report on its numerical performance.

Communicated by P.M. Pardalos.

Financial support of EPSRC Grant GR/T02560/01 gratefully acknowledged.

P. Parpas · B. Rustem (✉)
Department of Computing, Imperial College, London, UK
e-mail: br@doc.ic.ac.uk

We will be concerned with the following problem:

$$\begin{aligned} &\min_x \max_y f(x,y), \\ &\text{s.t.} \qquad x \in X \subset \mathbb{R}^n, \\ &\qquad\quad y \in Y \subset \mathbb{R}^m, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ will be assumed to be a polynomial in both variables. Moreover, the sets X and Y will be assumed to be defined by polynomial inequalities as follows:

$$X = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0,\ i = 1, \ldots, n_x\},$$
$$Y = \{y \in \mathbb{R}^m \mid h_i(y) \geq 0,\ i = 1, \ldots, n_y\},$$

where $g : \mathbb{R}^n \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^m \to \mathbb{R}^{n_y}$, are polynomials.

Numerous models in finance, engineering, and economics can be formulated as minimax polynomial optimization problems. The interested reader can find applications in finance and economics in Ref. [1]. A general interpretation of the model in (1) is that the decision maker chooses a strategy from X as an optimal response to the worst case strategy. An opponent chooses the latter strategy from the set Y. When f is convex in X and concave in Y then the solution to (1) is a saddle point. When f satisfies convexity/concavity assumptions then very efficient algorithms are available for the solution of the problem, see for example Ref. [2]. Another special case of the problem in (1) is when the set Y is of finite cardinality. In this case the problem is called the discrete minimax problem. Various algorithms have been proposed for the solution of the discrete minimax problem (see e.g. Ref. [3]). When Y is not discrete and when f is not assumed to have any (known) convexity properties, then the problem in (1) is quite hard to solve in practice. To explain the numerical and theoretical difficulties for developing algorithms for (1) in the general case, we introduce the following function:

$$\begin{aligned} \Phi(x) = \max_y\ & f(x,y), \\ \text{s.t.}\quad & h(y) \geq 0. \end{aligned} \tag{2}$$

For obvious reasons, $\Phi(x)$ is called the max function. Thus, the problem in (1) is equivalent to

$$\begin{aligned} &\min_x \Phi(x), \\ &\text{s.t.}\quad g(x) \geq 0. \end{aligned} \tag{3}$$

One cannot directly use conventional algorithms to solve (3) because the max-function is in general non-differentiable. Moreover, function evaluations are extremely expensive because of the global optimization required over y to evaluate the max-function for some fixed x. For the unconstrained version of (3), some algorithms have been proposed in the literature; see Ref. [1] for a computational comparison between some of these algorithms. For the constrained version of the problem,

algorithms were proposed in [4]. Since we will be interested in the global solution of the problem, we will not make a complete review of the available methods for the solution of the problem. Instead, we refer the interested reader to [4] for a review.

Apart from the nondifferentiable view adopted in the works mentioned above, one can equivalently formulate (1) as the following semi-infinite programming problem:

$$\begin{aligned} &\min_{x,\theta} \theta, \\ &\text{s.t.} \quad \theta \geq f(x,y), \quad \forall y \in Y, \\ &\qquad g(x) \geq 0. \end{aligned} \tag{4}$$

The problem in (4) has an infinite number of constraints since the constraints need to be satisfied for all y in Y, and the latter set has infinite cardinality. Most of the available methods for the solution of (4) use some kind of discretization approach [5–7], see [8] for a review.

The algorithm that we propose in this paper is related to the discretization approach to the semi-infinite formulation. Our work differs from others in that we endeavor to compute the global minimum of (4). Moreover, we use tools from semidefinite programming to approximate the infinite number of constraints. To the authors knowledge this is the first paper to propose a numerical algorithm for the global optimization of the constrained continuous minimax problem. Moreover, we believe that the links we draw between semi-infinite and semidefinite programming will be useful in other problems too.

2 Algorithm

In this section, we reformulate the minimax problem so that its solution can be approximated using techniques from semidefinite programming. We will exploit the links between global optimization and semidefinite programming proposed recently by Lassere [9], and Parillo [10].

The crux of the proposed algorithm lies on the way the following constraint:

$$\theta - f(x,y) \geq 0, \quad \forall y \in Y, \tag{5}$$

is reformulated as an SDP problem. Let x and θ be fixed to x_k and θ_k respectively. Then if $\theta_k - f(x_k, y)$ can be written as a sum of squares of polynomials in y, it follows that (x_k, θ_k) satisfy the semi-infinite constraints in (5). Therefore, to check the feasibility of (x_k, θ_k), we need to establish whether or not there exist polynomials $r_i(y)$ such that

$$\theta_k - f(x_k, y) = \sum_i r_i(y)^2. \tag{6}$$

At first sight it may seem that we have not made any substantial progress. However, the problem of computing sum-of-squares representations of non-negative polynomials has a long and distinguished history. In fact, whether or not a nonnegative poly-

nomial can be written as a sum-of-squares of rational functions was Hilbert's 17th problem, in his famous list of problems. The question was answered positively by Artin in 1927. Before we explain how we apply the available results on nonnegative polynomials, it is necessary to introduce some notation.

We will denote polynomials as follows:

$$k(x) = \sum_{\alpha \in \mathcal{S}} k_\alpha x^\alpha,$$

where $x \in \mathbb{R}^n$. $\mathcal{S} \subset \mathbb{Z}^n$ will be used to denote the support of the polynomial. By $|\mathcal{S}|$ we will denote the cardinality of the support. $\mathbb{R}^{\mathcal{S}}$ will be used to denote an $|\mathcal{S}|$-dimensional Euclidean space indexed by $\alpha \in \mathcal{S}$. The coordinates will be assumed to be lexicographically ordered. The elements of $k \in \mathbb{R}^{\mathcal{S}}$ are indexed by k_α with $\alpha \in \mathcal{S}$. $\mathbb{R}[x_1, \ldots, x_n] = \mathbb{R}[X]$ will be used to denote the set of polynomials in n variables. $\sum \mathbb{R}[X]^2$ will denote the set of sum of squares of polynomials.

With sets defined by polynomial inequalities, we associate the set

$$M(X) \triangleq \sum \mathbb{R}[X]^2 + \sum_i \sum \mathbb{R}[X]^2 g_i.$$

The latter set is referred to as the quadratic module generated by X Ref. [11, 12]. We will use $S\mathbb{R}^{\mathcal{S}}$ to denote the set of $|\mathcal{S}| \times |\mathcal{S}|$ symmetric matrices with coordinates indexed by $\alpha \in \mathcal{S}$. Matrices in $S\mathbb{R}^{\mathcal{S}}$ satisfy $X_{\alpha\beta} = X_{\beta\alpha}$; for any $d \in S\mathbb{R}^{\mathcal{S}}_+$, we have

$$d^T X x = \sum_{\alpha\beta} d_\alpha d_\beta X_{\alpha\beta} \geq 0.$$

With the notation out of the way we can return to the main thread of our discussion. Before we delve into the issue on how the coefficients and degrees of the polynomials in (6) can be calculated, we discuss the delicate issue of the existence of such a representation. In fact, Hilbert proved in 1888 (and Motzkin found an example in 1967) that a nonnegative polynomial depending on two variables cannot in general be represented as a sum of squares. We refer the interested reader to Ref. [13] for an in depth discussion about Hilbert's 17th problem. We will be taking advantage of the following result due to Putinar Ref. [14].

Theorem 2.1 (Putinar Ref. [14]) *Assume that there exists a polynomial $q \in M(Y)$ such that $\{y \in \mathbb{R}^m \mid q(y) \geq 0\}$ is compact. Then, every positive polynomial on Y belongs to $M(Y)$.*

This theorem has been quite instrumental in developing efficient approaches to the global optimization of polynomials over semialgebraic sets. The compactness assumption of the theorem above can be restated in many equivalent ways. We refer the interested reader to Schweighofer Ref. [12] for a discussion. Since we are interested in the practical solution of the problem in (1) we will follow Lassere Ref. [9] and add a redundant constraint $b - \sum_i y_i^2 \geq 0$ to the set Y. Under this alteration of the constraint set, the assumptions of Theorem 2.1 can be satisfied. If b is chosen to be

large enough then this modification of the problem will not change the solution of the problem we are concerned with. A final remark about Putinar's theorem as applied to the constraint (5) is that the result concerns positive polynomials. The effect of this incongruity can be mitigated by enforcing (5) within $\epsilon > 0$. Since ϵ can be taken to be arbitrarily small, the modification will not affect the numerical performance of the proposed method.

The following lemma is an application of a result of Powers and Wormann Ref. [15] to the continuous minimax problem.

Lemma 2.1 *Assume that, for fixed x and θ to x_k and θ_k respectively, the polynomial*

$$f(x_k, y) - \theta_k = \sum_{\alpha \in \mathcal{C}} c(x_k, \theta_k) y^\alpha$$

is of degree $2d$ in the y variables and let $\mathcal{C}$ denote its support. Then, if the following semidefinite programming problem is feasible:

$$\sum_{\beta+\gamma=\alpha, |\beta|\le d, |\gamma|\le d} X_{\beta\gamma} = c(x_k, \theta_k)_\alpha \quad |\alpha| \le 2d,$$

$$X \succcurlyeq 0, \quad X \in S\mathbb{R}^{\mathcal{C}},$$

then (x_k, θ_k) satisfies the semi-infinite constraints

$$\theta_k - f(x_k, y) \ge 0, \quad \forall y \in \mathbb{R}^m.$$

Proof Let z denote the vector of monomials with degree of at most d. Since $X \succcurlyeq 0$ is positive semidefinite, it can be written as

$$X = \sum_i \lambda_i v_i v_i^T,$$

where λ_i denotes the nonnegative eigenvalue associated with the ith eigenvector v_i. Let $\widehat{v}_i = \sqrt{\lambda} v_i$; then

$$z^T \Big(\sum \widehat{v}_i \widehat{v}_i^T \Big) z = \sum_i \Big(\sum_\alpha \widehat{v}_{\alpha i} y^\alpha \Big)^2,$$

and by matching coefficients,

$$z^T \Big(\sum \widehat{v}_i \widehat{v}_i^T \Big) z = \sum_{\beta,\gamma} X_{\beta\gamma} y^\alpha y^\beta = \sum_\alpha c(x_k, \theta_k)_\alpha y^\alpha,$$

the result follows. □

The lemma above, while useful for understanding the link between SDP and sum-of-square representations, it is not sufficiently general for our intended application.

We will follow Lassere's method [9] and formulate the max function as follows:

$$\begin{aligned}\Phi(x) &= \inf\{\phi \mid f(x,y) < \phi\}\\ &= \inf\{\phi \mid f(x,y) \le \phi\}\\ &= \inf\{\phi \mid \phi - f(x,y) \in M(Y)\}.\end{aligned}$$

Where the last equality follows from Putinar's theorem. In order to make sure the assumptions of Theorem 2.1 are satisfied, we follow the approach in Ref. [9] and add a redundant ball constraint: $\|y\| \le \varrho$; where ϱ is selected to be large enough so that the value of the max-function remains the same. Thus, given a point (x_k, θ_k), we can check if

$$\theta_k - f(x_k, y) \in M(Y). \tag{7}$$

If the preceding equation is satisfied we declare (x_k, θ_k) as feasible. Otherwise the point must be infeasible and we generate a y_k that violates (5). Generating such a point is a difficult issue, we will return to this point later in this section. We first discuss how can one numerically check condition (7).

Checking for membership in the set $M(Y)$ is not computationally tractable since the latter set involves polynomials of arbitrary degree. The basic idea is then to truncate this structure to $M_\tau(Y)$, where [12]

$$\begin{aligned}M_\tau(Y) &\triangleq \sum \mathbb{R}[Y]^2_{d_0} + \sum_i \sum \mathbb{R}[Y]^2_{d_i} h_i\\ &\triangleq \Big\{u_0(y) + \sum_i u_i(y) h_i(y) \mid u_i \in \sum \mathbb{R}[Y]^2,\\ &\qquad \deg(u_0) \le \tau, \deg(u_i h_i) \le \tau, i = 1, \ldots, n_y\Big\}.\end{aligned}$$

$\sum \mathbb{R}[Y]^2_{d_i}$ represents the set of sum-of-squares of polynomials with degree of at most d_i, where

$$\begin{aligned}d_0 &= \max\{k \in \mathbb{N} \mid 2k \le \tau\},\\ d_i &= \max\{k \in \mathbb{N} \mid 2k + \deg(h_i) \le \tau\}, \quad i = 1, \ldots, n_y.\end{aligned}$$

τ is selected to belong to the following set:

$$\tau \in \Big\{s \in \mathbb{N} \mid s \ge \max\{\deg(h_1), \deg(h_2), \ldots \deg(h_m), \deg(f_x)\}\Big\},$$

where f_x denotes the polynomial $f(x, y)$ in the y variables when x is fixed to some value. Using this truncation, we need to check the following problem for feasibility:

$$\begin{aligned}&\theta_k - f(x_k, y) = u_0(y) + \sum_i u_i(y) h_i(y),\\ &u_i(y) \in \sum \mathbb{R}[Y]^2, \quad \deg(u_0),\ \deg(u_1 h_i), \ldots, \deg(u_m h_m) \le \tau.\end{aligned} \tag{8}$$

According to Putinar's theorem, as τ is increased, one will eventually be able to ascertain set membership for the full set $M(Y)$. In practice, this seems to happen early on in the truncation process [9, 16].

The SDP associated with (8) is given by

$$\begin{aligned} &\theta_k - f_{x_k} = \left\langle (\hat{Y}^{\beta+\gamma}), G_0 \right\rangle + \sum_i \left\langle (\hat{Y}^{\beta+\gamma} h_i), G_i \right\rangle, \quad \forall (\beta,\gamma) \in \hat{d}_i \times \hat{d}_i, \\ &G_i \in S\mathbb{R}^{\Lambda(d_i)\times\Lambda(d_i)}, \quad i = 0,\ldots,m, \end{aligned} \tag{9}$$

where $\hat{d}_i \triangleq \{a \in \mathbb{N}^n \mid |a| \le d_i\}$, and $\bar{Y}^a_{a\in\hat{d}}$ is the basis of $\mathbb{R}[Y]_d$. Finally, f_{x_k} denotes the coefficients of the polynomial $f(x, y)$ when x is fixed to x_k.

We are interested in both checking the feasibility of a point (x_k, θ_k) and in extracting a vector y_k in the case (x_k, θ_k) is infeasible. For these reasons it will be more efficient to solve an optimization problem rather than the feasibility problem in (8). The optimization problem is given by

$$\begin{aligned} &\max \gamma, \\ &\text{s.t.} \quad \gamma - f_{x_k} = \left\langle (\hat{Y}^{\beta+\gamma}), G_0 \right\rangle + \sum_i \left\langle (\hat{Y}^{\beta+\gamma} h_i), G_i \right\rangle, \quad \forall (\beta,\gamma) \in \hat{d}_i \times \hat{d}_i, \\ &\qquad G_i \in S\mathbb{R}^{\Lambda(d_i)\times\Lambda(d_i)}, \quad i = 0,\ldots,m. \end{aligned} \tag{10}$$

Let γ^* denote the objective function value of the problem above. Then it is easy to see that if $\gamma^* = \theta_k$ then (x_k, θ_k) is feasible. Otherwise a violating y vector will need to be computed. We now turn our attention to the thorny issue of extracting such a vector. Under certain conditions (given below) this vector can be extracted from the dual of (10). Following the usual procedure for taking duals in SDPs, we find

$$\begin{aligned} \min \quad & \sum_\alpha (f_{x_k})_\alpha z_\alpha, \\ \text{s.t.} \quad & M_\tau(z) \succcurlyeq 0, \\ & M_{\tau-d_i}(h_i z) \succcurlyeq 0, \quad i = 1,\ldots,m, \end{aligned}$$

where the matrices M_τ, and $M_{\tau-d_i}$ are called moment and localizing matrices respectively [9, 12, 17]. These matrices are constructed as follows: let

$$S_k = \left\{ \alpha \in \mathbb{Z}^n_+ \mid |\alpha| = \sum_i \alpha_i \le k \right\};$$

given a sequence $z = (z)_{\alpha\in S_k}$, then the moment matrix $M_\tau(z)$ is the matrix indexed by S_k with the (α, β)th entry given by $z_{\alpha+\beta}$. The localizing matrix $M_{k-d_i}(h_i z)$ is constructed as follows: define a shifted vector $h_i z \triangleq M_k(z) h_i$, where the αth entry of $h_i z$ is given by

$$(h_i z)_\alpha = \sum_\beta h_{i\beta} z_{\alpha+\beta}.$$

The moment matrix of $h_i z$ is defined as the localizing matrix.

Suppose that, at the τth relaxation of (10), an optimal solution is obtained. If $\gamma^* = \theta_k$, then (x_k, θ_k) is feasible. Otherwise, we extract a vector y_k that violates (5). A sufficient condition that ensures optimality of the τth relaxation is given by the following rank condition:

$$\operatorname{rank} M_\tau(z^*) = \operatorname{rank} M_{\tau-d}(z^*), \tag{12}$$

where

$$d = \max_j \left\lceil \frac{\deg(q_j)}{2} \right\rceil.$$

Whenever condition (12) holds, it was shown in Ref. [18] that one could extract an optimal solution vector out of the dual problem (11). In addition it was shown in Ref. [12] that when the problem has a unique global minimum then the relaxation (11) is guaranteed to (asymptotically) converge to this unique point. In theory the SDP relaxations in (10) and its dual (11) are guaranteed to eventually yield the optimal objective function value. In terms of theoretical results, much less is available when the solution vector is required. However, using higher relaxations, and perturbations we were able to solve problems with many, and even infinite number of maximizers. We will discuss practical aspects of the numerical implementation of the algorithm in Sect. 4.

We can now specify the algorithm for the global optimization of continuous minimax problems. The algorithm consists of two main steps. In the first step the set Y is discretized and solved to global optimality to obtain (x_k, θ_k). The second step consists of checking whether (x_k, θ_k) is feasible. If it is then we stop, and declare x_k as the optimal solution vector of the minimax problem. Otherwise, we compute a y_k that proves the infeasibility of (x_k, θ_k). We then add y_k to the discretized version of the set Y and we repeat the process. The algorithm is given below; we omit the details on how Step 1 is performed, since it is derived in a similar way as (10) and (11) (see also [9, 12]).

Step 0. Let $k = 0$ and let Y_k be some finite subset of Y. Let $\tau_{\max} > 0$ be a given scalar.

Step 1. Solve

$$\begin{aligned} \min \quad & \hat{\theta}, \\ \text{s.t.} \quad & M_\tau(\hat{z}) \succcurlyeq 0, \\ & M_{\tau-d_i}(\hat{z}(\theta - f_y)) \succcurlyeq 0, \quad \forall y \in Y_k, \\ & M_{\tau-d_i}(\hat{z} g_i) \succcurlyeq 0, \quad i = 1, \ldots, n_x. \end{aligned} \tag{13}$$

Extract a solution (x_k, θ_k) from the problem above. If extraction is not possible, then increase τ. If $\tau > \tau_{\max}$, then stop; the problem may violate the rank assumption (12). If a solution is extracted, go to the next step.

Step 2. Solve:

$$\begin{aligned} \min \quad & \sum f_\alpha^{x_k} z_\alpha, \\ \text{s.t.} \quad & M_\tau(z) \succcurlyeq 0, \\ & M_{\tau-d_i}(h_i z) \succcurlyeq 0, \quad i = 1, \ldots, m. \end{aligned}$$

Let z^* denote the optimal solution of the problem above. If $\theta_k = \sum f_\alpha^{x_k} z_\alpha^*$ then stop the vector x_k solves the problem. Otherwise, extract a solution vector y^k that violates (5). If extraction is not possible, then (as above) increase τ until either extraction is possible or τ exceeds $\tau_{\max}$.

Step 3. Set $Y_{k+1} = Y_k \cup \{y_k\}$, set $k := k + 1$ and go to Step 1.

3 Convergence Analysis

In this section, we establish the convergence of the algorithm under certain conditions. The proof makes uses of point-to-set mappings. We refer the interested reader to Refs. [19, 20] for more information on these mappings. The following definitions are taken from Ref. [19].

Assumption 3.1 A point-to-set map η from a set A into a set B is a map which associates a subset of B with each point of A.

Assumption 3.2 η is open at a point $\hat{\alpha} \in A$ if, for a sequence $\{\alpha_k\} \subset A$ with $\alpha_k \to \hat{\alpha}$, and $\hat{\beta} \in \eta(\hat{\alpha})$ imply the existence of a sequence $\{\beta_k\} \subset B$ such that $\beta_k \in \eta(\alpha_k)$ and $\beta_k \to \hat{\beta}$.

Assumption 3.3 η is closed at a point $\hat{\alpha} \in A$ if for a sequence $\{\alpha_k\} \subset A$ with $\alpha_k \to \hat{\alpha}$, $\beta_k \in \eta(\hat{\alpha}_k)$ and $\beta_k \to \hat{\beta}$ imply that $\hat{\beta} \in \eta(\hat{\alpha})$.

Assumption 3.4 η is continuous at a point $\hat{\alpha} \in A$ if it is both open and closed.

We will say that η is open, closed or continuous on A if it has the respective property for every $\alpha \in A$. We now proceed to place the proposed algorithm in the framework of point-to-set maps. For the rest of this section we assume that the rank assumption in (12) is satisfied, and that $\tau_{\max}$ has been allowed to be large enough so that a solution vector is eventually extracted. In practice we have found that a modest $\tau_{\max}$ is enough to render the algorithm usable in practice. More on the practical implementation of the algorithm will be given in the next section.

We will use $\hat{Y}$ to denote all finite subsets of Y. Step 2 of the algorithm can be viewed as

$$\eta : \mathbb{R}^n \times \mathbb{R} \times \hat{Y} \to \hat{Y}.$$

Let (x_k, θ_k) and Y_k represent the solution vector obtained at Step 1 and the finite subset of Y available at Step 1. k will denote the iteration number throughout this section. The η mapping is defined as follows:

$$\eta(x_k, \theta_k, \hat{Y}_k) = \{Y_k \cup \{\hat{y}\} \mid \theta_k - f(x_k, \hat{y}) < 0,\ \hat{y} \in Y\},$$

where $\hat{y}$ is the optimal vector extracted from the solution (11). If more than one solution is extracted then, without loss of generality, the one with the highest norm is chosen.

Step 1 of the proposed algorithm can be viewed as the following map:

$$\zeta(\eta(x_k, \theta_k, Y_k)) \to X \times \mathbb{R},$$

where

$$\zeta(\hat{Y}) = \{(\hat{x}, \hat{\theta}) \in X \times \mathbb{R} \mid (\hat{x}, \hat{\theta}) \text{ is the solution extracted from (13)}\}.$$

Lemma 3.1 *$\zeta \circ \eta$ is closed.*

Proof We need to show that, if

$$(x_k, \theta_k, Y_k) \to (x^*, \theta^*, Y^*)$$

and

$$\begin{aligned} Y_{k+1} &\in \eta(x_k, \theta_k, Y_k), \\ x_{k+1} &\in \zeta(Y_k), \end{aligned}$$

then

$$\begin{aligned} Y^* &\in \eta(x^*, \theta^*, Y^*), \\ x^* &\in \zeta(Y^*). \end{aligned}$$

Since $Y_k \to Y^*$ for k large enough, we must have that

$$Y_k = Y_{k+1} = Y^*.$$

But for any (x, θ), we must then have

$$\eta(x, \theta, Y_k) = \eta(x, \theta, Y_{k+1}) = \eta(x, \theta, Y^*).$$

Since $Y_k \in \beta(x, \theta, Y_k) = \eta(x, \theta, Y^*)$ and $Y^* = Y_k$, it follows that the inner map is closed.

For the outer map, we need to show that, if

$$(x_k, \theta_k) \to (x^*, \theta^*) \quad \text{with } (x_{k+1}, \theta_{k+1}) \in \zeta(\eta(x_k, \theta_k, Y_k)),$$

then

$$(x^*, \theta^*) \in \zeta(\eta(x^*, \theta^*, Y^*)).$$

Since x_k is solution to

$$\min_x \Phi_{k-1}(x),$$

where

$$\Phi_k(x) = \max_{y \in Y_k} f(x, y),$$

it follows by the continuity of Φ_k that x^* is a local solution to

$$\min_x \Phi_\infty(x),$$

where

$$\Phi_\infty(x) = \max_{y \in Y^*} f(x, y).$$

In other words, if x_k is optimal for $\Phi_\infty(x)$, then $\{x_k\} \to x^*$ imply that is a solution to

$$\min_x \Phi_\infty(x).$$

Since

$$\Phi_\infty(x_k) \leq \Phi_\infty(x),$$

by continuity

$$\Phi_\infty(x^*) \leq \Phi_\infty(x). \qquad \square$$

Theorem 3.1 *Let (x^*, θ^*), Y^* be accumulation points of the sequences $\{x_k, \theta_k\}$ and let $\{Y_k\}$ be generated by the algorithm. Then, x^* is a global optimum solution of* (1).

Proof Let $\{x_k, \theta_k\} \to (x^*, \theta^*)$ and $\{Y_k\} \to Y^*$; then, we must have

$$(x^*, \theta^*) \in \zeta(Y^*).$$

Since $Y^* \in \eta(x^*, \theta^*, Y^*)$, we also have

$$f(x^*, y) \leq f(x^*, y^*), \quad \exists y^* \in Y^*, \ \forall y \in Y.$$

Since x^* is a local solution to

$$\min_x \Phi_\infty(x),$$

we have that, for k sufficiently large,

$$\Phi_\infty(x^*) \leq \Phi_\infty(x) = \max_{y \in Y^*} f(x, y) = \max_{y \in Y} f(x, y) = \Phi(x). \qquad \square$$

4 Numerical Results

In this section we discuss the numerical implementation of the algorithm. A possible deficiency of the proposed algorithm is its dependence on the existence of a sum-of-squares representation. While the latter condition can be asymptotically satisfied, a more strict assumption is the possibility of extracting solution vectors at every iteration. The aim of our numerical experiments was to assess whether the rank condition (12) was too strict for our intended application.

The algorithm was implemented in the MATLAB environment, and GloptiPoly Ref. [21] was used to solve the subproblems in Steps 1 and 2 of the algorithm. The test problems were taken from the literature and they can be found in Chap. 5 of Ref. [1]. The dimensions of the problems are given in Table 1, where x-Dim stands

Table 1 Test problems

Name	# x-Dim	# y-Dim
1	2	2
2	2	2
3	2	2
4	2	3
5	3	3
6	4	3
7	5	5
8	1	1
9	2	2
10	2	2
11	4	2
12	4	3

Table 2 Solution results

Name	#Iterations	Max-Relax	Purturb-Param
1	6	0	0
2	6	0	0
3	10	3	0
4	8	0	0
5	4	0	0
6	10	0	0
7	25	0	0
8	8	0	0
9	28	9	$u * 0.001$
10	24	7	$u * 0.001$
11	16	0	$u * 0.001$
12	19	4	$u * 0.001$

for the dimension of the function in the x variables, and similarly for y-Dim. Problems 1–7 are convex in x, and concave in the y variables. As was mentioned in the introduction such problems can be solved efficiently with more specialized methods. Moreover there are no issues concerning the global optimization of such functions since local minima are also global minima. The purpose of these problems was to test the correctness of the implementation. More importantly, these tests allow us to assess the efficiency of the algorithm in computing the global optimum as compared with more efficient (but less general) methods.

Test problems 8–12 have no convexity properties. Problems 8–10 have an infinite number of maximizers for each x, while problems 11–12 have multiple (but finite) number of maximizers. As mentioned above, the purpose of these problems is to assess the numerical performance of the algorithm in view of these multiple optima.

The solution statistics for each problem are reported in Table 2. Max-Relax corresponds to the maximum relaxation required to solve the test problems in Steps 1

and 2 of the proposed method. All problems were solved with a $10e{-}8$ precision. As it can be seen the convex problems require a modest amount of iterations. While the concave problems require much more iterations. In order to solve problems 8–12, we had to resort to slight permutations of the decisions variables. This was done as follows:

$$\hat{x}_i = x_i + u_i * 0.001, \quad i = 1, \ldots, n,$$

where u_i is a uniform random number in the interval [0, 10]. The same strategy was followed for the y variables. Perturbing the variables allows the algorithm to break the symmetries in the problem and facilitates the extraction of the optimum solution vector.

References

1. Rustem, B., Howe, M.A.: Algorithms for Worst-case Design with Applications to Risk Management. Princeton University Press, Princeton (2001)
2. Zakovic, S., Rustem, B., Pantelides, C.: An interior point algorithm for computing saddle points of constrained continuous minimax. Ann. Oper. Res. **99**, 59–78 (2000)
3. Rustem, B., Nguyen, Q.: An algorithm for inequality constrained discrete min–max problems. SIAM J. Optim. **8**, 256–283 (1998)
4. Rustem, B., Zakovic, S., Parpas, P.: An interior point algorithm for continuous minimax. J. Optim. Theory Appl. **136**, 87–103 (2008)
5. Blankenship, J.W., Falk, J.E.: Infinitely constrained optimization problems. J. Optim. Theory Appl. **19**, 261–281 (1976)
6. Demyanov, V.F., Malozemov, V.N.: Introduction to Minimax. Wiley, New York (1974)
7. Stein, O., Still, G.: Solving semi–infinite optimization problems with interior point techniques. SIAM J. Control Optim. **42**, 769–788 (2003)
8. Zakovic, S., Rustem, B.: Semi–infinite programming and applications to minimax problems. Ann. Oper. Res. **124**, 81–110 (2003)
9. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2000/2001)
10. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Math. Program. Ser. B **96**(2), 293–320 (2003). Algebraic and geometric methods in discrete optimization
11. Prestel, A., Delzell, C.N.: Positive Polynomials. Springer Monographs in Mathematics. Springer, Berlin (2001). From Hilbert's 17th problem to real algebra
12. Schweighofer, M.: Optimization of polynomials on compact semialgebraic sets. SIAM J. Optim. **15**(3), 805–825 (2005) (electronic)
13. Reznick, B.: Some concrete aspects of Hilbert's 17th problem. In: Real Algebraic Geometry and Ordered Structures, Baton Rouge, LA, 1996. Contemp. Math., vol. 253, pp. 251–272. Am. Math. Soc., Providence (2000)
14. Putinar, M.: Positive polynomials on compact semi-algebraic sets. Indiana Univ. Math. J. **42**(3), 969–984 (1993)
15. Powers, V., Wörmann, T.: An algorithm for sums of squares of real polynomials. J. Pure Appl. Algebra **127**(1), 99–104 (1998)
16. Kim, S., Kojima, M., Waki, H.: Generalized Lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems. SIAM J. Optim. **15**(3), 697–719 (2005)
17. Laurent, M.: Moment matrices and optimization over polynomials—a survey. Lecture Notes (September 2005)
18. Henrion, D., Lasserre, J.B.: Detecting global optimality and extracting solutions in gloptipoly. In Garulli, A., Henrion, D. (eds.) Positive Polynomials in Control. Lecture Notes on Control and Information Sciences
19. Hogan, W.: Point-to-set maps in mathematical programming. SIAM Rev. **15**, 591–603 (1973)
20. Zangwill, W.I., Mond, B.: Nonlinear Programming: A Unified Approach. Prentice-Hall International Series in Management. Prentice-Hall Inc., Englewood Cliffs (1969)
21. Henrion, D., Lasserre, J.B.: Gloptipoly: Global optimization over polynomials with matlab and sedumi. ACM Trans. Math. Softw. **29**(2) (2003)