Computational Optimization and Applications (2022) 82:89–106
https://doi.org/10.1007/s10589-022-00361-3

Local saddle points for unconstrained polynomial optimization

Wenjie Zhao[1] · **Guangming Zhou**[1]

Received: 6 January 2021 / Accepted: 21 February 2022 / Published online: 14 March 2022

Abstract

This paper gives an algorithm for computing local saddle points for unconstrained polynomial optimization. It is based on optimality conditions and Lasserre's hierarchy of semidefinite relaxations. It can determine the existence of local saddle points. When there are several different local saddle point values, the algorithm can get them from the smallest one to the largest one.

Keywords Saddle point · Polynomial optimization · Lasserre's hierarchy · Semidefinite relaxation

Mathematics Subject Classification 90C22 · 90C47 · 49K35 · 65K05

1 Introduction

Let $f(x, y)$ be a continuous function in $(x, y) \in X \times Y$, where $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$ are two sets (n and m are positive integers). A pair $(x^*, y^*) \in X \times Y$ is said to be a saddle point of $f(x, y)$ over $X \times Y$ if

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant f(x, y^*), \ \forall (x, y) \in X \times Y.$$

Saddle point problems can be applied to wide fields, such as game theory and equilibrium theory [25, 27, 28, 41], robust optimization [3], optimal control problems [44], generative adversarial nets in deep learning [15], etc. We refer to [5] for the basic theory of saddle point problems.

✉ Guangming Zhou
zhougm@xtu.edu.cn

Wenjie Zhao
zhaowj@smail.xtu.edu.cn

1 School of Mathematics and Computational Science & Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan 411105, Hunan, China

There exists much work on saddle point problems based on subgradients [30], primal-dual method [8, 17], perturbations [20], variational inequalities [9, 31], Uzawa method [13], Krylov subspace method [40], splitting iteration [2], preconditioners for saddle point problems [4, 7, 12, 16] and other approaches. These classical methods focus on convex-concave type saddle point problems. Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio studied the saddle point problems for non-convex optimization [11, 39]. Nie, Yang and Zhou [38] proposed a numerical method for obtaining saddle points of polynomial optimization. Zhou, Wang and Zhao [45] proposed an approach for computing saddle points of rational functions. The polynomials or rational functions in [38, 45] are not limited to the convex-concave types. Recently, Adolphs, Daneshmand, Lucchi and Hofmann [1] defined local saddle points.

Definition 1.1 A pair $(x^*, y^*) \in X \times Y$ is said to be a local saddle point for continuous function $f(x, y)$ if there exists $\tau > 0$ such that

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant f(x, y^*)$$

for all $(x, y) \in (X \cap B(x^*, \tau)) \times (Y \cap B(y^*, \tau))$, where

$$B(x^*, \tau) := \{x \in \mathbb{R}^n \mid \|x - x^*\|^2 \leqslant \tau\}, \quad B(y^*, \tau) := \{y \in \mathbb{R}^m \mid \|y - y^*\|^2 \leqslant \tau\}.$$

In Definition 1.1, let $f(x, y) \in \mathbb{R}[x, y]$ be a polynomial in $(x, y) \in X \times Y$, where $X := \mathbb{R}^n$ and $Y := \mathbb{R}^m$, if there exist $\tau > 0$ and $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$ such that

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant f(x, y^*) \tag{1.1}$$

for all $(x, y) \in B(x^*, \tau) \times B(y^*, \tau)$ (resp., $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$), we say (x^*, y^*) is a local (resp., global) saddle point of the polynomial $f(x, y)$.

In this paper, we focus on computing local saddle points for unconstrained polynomial optimization. Suppose that a polynomial has finitely many local saddle points. We propose an algorithm based on second-order optimality conditions and Lasserre's hierarchy of relaxations for computing them and their corresponding local saddle point values. If a polynomial does not have local saddle points, the algorithm can detect the nonexistence. In addition, under the assumption that the polynomial has finitely many critical points, the algorithm can terminate after finitely many iterations.

The remaining of this paper is organized as follows. We describe the basic theory of polynomial optimization in Sect. 2. In Sect. 3, after the second-order optimality conditions and Lasserre's hierarchy of relaxations are introduced, we give an algorithm for solving local saddle points of unconstrained polynomial optimization. Convergent results and proofs are given in Sect. 4. Some examples are given to illustrate that the algorithm is effective in Sect. 5. Conclusions and discussions are shown in Sect. 6.

2 Preliminaries

2.1 Notation

Let $\mathbb{N}$, $\mathbb{N}^{+}$, $\mathbb{R}$ and $\mathbb{C}$ be the set of nonnegative integers, positive integers, real numbers and complex numbers, respectively. For $n, m \in \mathbb{N}^{+}$, $\mathbb{R}^n$ and $\mathbb{R}^m$ denote n and m dimension Euclidean space, respectively. $\mathbb{R}[x] := \mathbb{R}[x_1, x_2, \dots, x_n]$ is the ring of polynomials in $x := (x_1, x_2, \dots, x_n)$ with real coefficients, and $\mathbb{R}[x]_d$ represents the set of polynomials in $\mathbb{R}[x]$ with their degrees not more than d. $\mathbb{R}[x, y]$ and $\mathbb{R}[x, y]_d$ can be defined similarly. For $n \in \mathbb{N}^{+}$, let $\mathbb{N}^n_d := \{\alpha \in \mathbb{N}^n \mid \alpha_1 + \alpha_2 + \cdots + \alpha_n \leqslant d\}$. The symbol $\deg(f)$ stands for the degree of the polynomial f. The norm $\|\cdot\|$ is the standard Euclidean norm. $\lceil k \rceil$ indicates the smallest integer not less than k. We denote by $\mathrm{tr}\,(A)$ the trace of matrix A. The superscript T means the transpose of a matrix or vector. We write $A \succ 0$, $A \succeq 0$, $A \prec 0$ and $A \preceq 0$ to express that the matrix A is positive definite, positive semidefinite, negative definite and negative semidefinite, respectively. The symbol $\mathrm{diag}\,(D_1, D_2, \dots, D_n)$ denotes the block diagonal matrix whose diagonal square blocks are $D_1, D_2, \dots, D_n$. $\nabla_x f = (f_{x_1}, f_{x_2}, \dots, f_{x_n})$ denotes the gradient vector of the polynomial $f(x, y)$ in $x := (x_1, x_2, \dots, x_n)$, where f_{x_i} is the partial derivative of $f(x, y)$ with respect to x_i. $\nabla_y f$ and f_{y_i} are defined similarly. $\nabla^2_x f$ (resp., $\nabla^2_y f$) stands for the Hessian matrix of the polynomial $f(x, y)$ in x (resp., y).

For $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathbb{N}^n$ and $d \in \mathbb{N}$, we define the following symbols

$$x^{\alpha} := x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}, \quad [x]_d := (1, x_1, x_2, \dots, x_n, x_1x_2, x_1x_3, \dots, x_n^d)^{\mathrm{T}}.$$

A polynomial $p(x) \in \mathbb{R}[x]_d$ can be written as

$$p(x) = \sum_{\alpha \in \mathbb{N}^n_d} p_{\alpha} x^{\alpha} = vec(p)^{\mathrm{T}} [x]_d,$$

where $vec(p)$ means the column coefficient vector of the polynomial $p(x)$ with respect to the basis $[x]_d$.

2.2 Sum of squares, ideals, quadratic modules and moments

A polynomial $s(x)$ is a sum of squares (SOS) if it can be written as $s(x) = s_1^2(x) + s_2^2(x) + \cdots + s_l^2(x)$ for some polynomials $s_1(x), s_2(x), \dots, s_l(x) \in \mathbb{R}[x]$. The symbol $\Sigma[x]$ denotes the set of SOS polynomials, and its k-th truncation is $\Sigma[x]_k := \Sigma[x] \cap \mathbb{R}[x]_k$. Obviously, if a polynomial is SOS, then it is nonnegative everywhere, but the inverse is not necessarily true.

A subset $I \subset \mathbb{R}[x]$ is an ideal if it satisfies $f + g \in I$ for all $f \in I, g \in I$ and $fh \in I$ for all $f \in I, h \in \mathbb{R}[x]$. For a tuple $p = (p_1, p_2, \dots, p_{l_1})$ of polynomials in $\mathbb{R}[x]$, $\mathrm{I}\,(p)$ denotes the smallest ideal which contains all p_i, i.e., $\mathrm{I}\,(p)$ is defined as

$$\mathrm{I}\,(p) := \{p_1h_1 + p_2h_2 + \cdots + p_{l_1}h_{l_1} \mid h_i \in \mathbb{R}[x],\ i = 1, 2, \dots, l_1\}.$$

We often need its truncation in computation, which is defined as

$$\mathrm{I}_{2k}(p) := \{p_1h_1 + p_2h_2 + \cdots + p_{l_1}h_{l_1} \mid h_i \in \mathbb{R}[x], \\ \deg(p_ih_i) \leqslant 2k,\ i = 1, 2, \dots, l_1\}.$$

The real and complex varieties of an ideal $I \in \mathbb{R}[x]$ are defined respectively as

$$\mathrm{V}_{\mathbb{R}}(I) := \{u \in \mathbb{R}^n \mid f(u) = 0,\ \forall f \in I\}, \\ \mathrm{V}_{\mathbb{C}}(I) := \{v \in \mathbb{C}^n \mid f(v) = 0,\ \forall f \in I\}.$$

For some nonnegative polynomials $q_1, q_2, \dots, q_{l_2} \in \mathbb{R}[x]$, the quadratic module of the tuple $q = (q_1, q_2, \dots, q_{l_2})$ is defined as

$$\mathrm{Q}(q) := \{s_0 + s_1q_1 + s_2q_2 + \cdots + s_{l_2}q_{l_2} \mid s_i \in \Sigma[x],\ i = 0, 1, 2, \dots, l_2\}.$$

The truncation of quadratic module $\mathrm{Q}(q)$ is defined as

$$\mathrm{Q}_k(q) := \{s_0 + s_1q_1 + s_2q_2 + \cdots + s_{l_2}q_{l_2} \mid s_0 \in \Sigma[x]_{2k}, \\ s_i \in \Sigma[x]_{2k-\deg(q_i)},\ i = 1, 2, \dots, l_2\}.$$

Let $\mathbb{R}^{\mathbb{N}_d^n}$ be the space of real sequences indexed by $\alpha \in \mathbb{N}_d^n$. A truncated multi-sequence (tms) $z \in \mathbb{R}^{\mathbb{N}_d^n}$, labelled as $(z)_d := (z_\alpha)_{\alpha \in \mathbb{N}_d^n}$, gives a Riesz linear functional such that

$$\mathscr{L}_z : f(x) = \sum_{\alpha \in \mathbb{N}_d^n} f_\alpha x^\alpha \quad \longmapsto \quad \mathscr{L}_z(f) = \sum_{\alpha \in \mathbb{N}_d^n} f_\alpha z_\alpha.$$

Let $u(x) \in \mathbb{R}[x]$ with $\deg(u) \leqslant 2k$. The k-th localizing matrix of $u(x)$ generated by a tms $z \in \mathbb{R}^{\mathbb{N}_{2k}^n}$, is the symmetric matrix $L_u^{(k)}(z)$ satisfying

$$\mathscr{L}_z(uab) = vec(a)^{\mathrm{T}} \left(L_u^{(k)}(z)\right) vec(b)$$

for all $a(x), b(x) \in \mathbb{R}[x]_{k-\lceil \deg(u)/2 \rceil}$. For example, when $n = 2$, $k = 2$ and $u(x) = 2 - 3x_1x_2$, for $z \in \mathbb{R}^{\mathbb{N}_4^2}$, we have

$$L_u^{(2)}(z) = \begin{bmatrix} 2z_{00} - 3z_{11} & 2z_{10} - 3z_{21} & 2z_{01} - 3z_{12} \\ 2z_{10} - 3z_{21} & 2z_{20} - 3z_{31} & 2z_{11} - 3z_{22} \\ 2z_{01} - 3z_{12} & 2z_{11} - 3z_{22} & 2z_{02} - 3z_{13} \end{bmatrix}.$$

When $u(x) = 1$, $L_u^{(k)}(z)$ is called the moment matrix, and it is denoted as

$$M_k(z) := L_1^{(k)}(z).$$

For example, when $n = 2$ and $k = 1$, for $z \in \mathbb{R}^{\mathbb{N}_2^2}$, we have

$$M_1(z) = \begin{bmatrix} z_{00} & z_{10} & z_{01} \\ z_{10} & z_{20} & z_{11} \\ z_{01} & z_{11} & z_{02} \end{bmatrix}.$$

Let H be an $l \times l$ symmetric matrix, whose each element H_{ij} is a polynomial in $\mathbb{R}[x]$. The k-th localizing matrix of H generated by the tms $z \in \mathbb{R}^{\mathbb{N}^n_{2k}}$ is the block symmetric matrix $L_H^{(k)}(z)$, which is defined as

$$L_H^{(k)}(z) := \left(L_{H_{ij}}^{(k)}(z)\right)_{1 \leqslant i \leqslant l, 1 \leqslant j \leqslant l},$$

and each block $L_{H_{ij}}^{(k)}(z)$ is a standard localizing matrix of the polynomial H_{ij}.

Let $\Sigma[x]^{l \times l}$ be the cone of all sums of $s_1 s_1^{\mathrm{T}} + s_2 s_2^{\mathrm{T}} + \cdots + s_r s_r^{\mathrm{T}}$ with $s_1, s_2, \ldots, s_r \in \mathbb{R}[x]^l$. When $l = 1$, $\Sigma[x]^{l \times l}$ is $\Sigma[x]$. The quadratic module of H is

$$\mathrm{Q}(H) := \Sigma[x] + \left\{ \mathrm{tr}(HS) \mid S \in \Sigma[x]^{l \times l} \right\}.$$

The k-th truncation of $\mathrm{Q}(H)$ is defined as

$$\mathrm{Q}_k(H) := \Sigma[x]_{2k} + \left\{ \mathrm{tr}(HS) \,\middle|\, \begin{array}{l} S \in \Sigma[x]^{l \times l}, \quad \deg(H_{ij} S_{ij}) \leqslant 2k, \\ \forall\, 1 \leqslant i \leqslant l,\ 1 \leqslant j \leqslant l \end{array} \right\}.$$

If there exists $r(x) \in \mathrm{I}(p) + \mathrm{Q}(q) + \mathrm{Q}(H)$ such that $\{x \mid r(x) \geqslant 0\}$ defines a compact set in $\mathbb{R}^n$, $\mathrm{I}(p) + \mathrm{Q}(q) + \mathrm{Q}(H)$ is said to be archimedean. If $\mathrm{I}(p) + \mathrm{Q}(q) + \mathrm{Q}(H)$ is archimedean, then the set $\mathbb{K} := \{x \mid p_1(x) = 0, \ldots, p_{l_1}(x) = 0,\ q_1(x) \geqslant 0, \ldots, q_{l_2}(x) \geqslant 0,\ H \succeq 0\}$ is compact. The converse is in general not true. However, if $\mathbb{K}$ is compact set, and let $\mathbb{K} \subseteq \{x \mid N - \|x\|^2 \geqslant 0\}$ where $N \geqslant 0$, then $\mathrm{I}(p) + \mathrm{Q}(q, N - \|x\|^2) + \mathrm{Q}(H)$ is archimedean. If $\mathrm{I}(p) + \mathrm{Q}(q) + \mathrm{Q}(H)$ is archimedean and the polynomial $\psi(x) \in \mathbb{R}[x]$ is strictly positive over $\mathbb{K}$, then $\psi(x) \in \mathrm{I}(p) + \mathrm{Q}(q) + \mathrm{Q}(H)$. Polynomial optimization is closely related to truncated moment problems. Optimizers can be extracted from Lasserre's hierarchy of relaxations. We refer to the work [14, 33, 37].

3 Computation of local saddle points

3.1 Optimality conditions

There exists classical work on second-order optimality conditions for unconstrained optimization. Let $\phi(x)$ be a polynomial in x, if there are $\tau > 0$ and x^* such that $\phi(x^*)$ is the smallest value of $\phi(x)$ on $B(x^*, \tau)$, then x^* is a local minimizer of $\phi(x)$, and $\phi(x^*)$ is a local minimum. If x^* is a local minimizer of the polynomial $\phi(x)$, then

$$\nabla_x \phi(x^*) = 0, \quad \nabla_x^2 \phi(x^*) \succeq 0.$$

Conversely, if there exists x^* such that

$$\nabla_x \phi(x^*) = 0, \quad \nabla_x^2 \phi(x^*) \succ 0,$$

then x^* is a local minimizer of the polynomial $\phi(x)$.

We detect local saddle points by the second-order optimality conditions. If the polynomial $f(x, y)$ has local saddle points, then it has finitely many local saddle point values. This is shown in Proposition 4.1.

Suppose that (x', y') is a minimizer of the problem

$$\begin{cases} f_1 := \min f(x,y) \\ \text{subject to } \nabla_x f(x,y) = 0, \quad \nabla_x^2 f(x,y) \succeq 0, \\ \qquad\qquad\quad \nabla_y f(x,y) = 0, \quad \nabla_y^2 f(x,y) \preceq 0. \end{cases} \tag{3.1}$$

According to the second-order sufficient optimality condition, if $\nabla_x^2 f(x', y') \succ 0$ and $\nabla_y^2 f(x', y') \prec 0$, then (x', y') is a local saddle point of the polynomial $f(x, y)$ and f_1 is corresponding local saddle point value. Otherwise, according to the inequality (1.1), if x' is a minimizer of

$$\begin{cases} \min \qquad\quad f(x,y') \\ \text{subject to } \nabla_x f(x,y') = 0, \quad \nabla_x^2 f(x,y') \succeq 0, \\ \qquad\qquad\quad \|x - x'\|^2 \leqslant \tau \end{cases} \tag{3.2}$$

for some small $\tau > 0$, and y' is a maximizer of

$$\begin{cases} \max \qquad\quad f(x',y) \\ \text{subject to } \nabla_y f(x',y) = 0, \quad \nabla_y^2 f(x',y) \preceq 0, \\ \qquad\qquad\quad \|y - y'\|^2 \leqslant \tau \end{cases} \tag{3.3}$$

for some small $\tau > 0$, then (x', y') is a local saddle point of the polynomial $f(x, y)$ and f_1 is corresponding local saddle point value. If $\tau > 0$ is very small, but x' is not a minimizer of the problem (3.2) or y' is not a maximizer of the problem (3.3), it is mostly open how to identify if it is a local saddle point or not, because detecting local optimality is NP-hard [29].

We assume that f_{r-1} $(r \geqslant 2)$ is obtained. We consider the following problem for detecting new local saddle points by adding the new constraint $f(x,y) \geqslant f_{r-1} + \delta$ with some $\delta > 0$

$$\begin{cases} f_r := \min f(x,y) \\ \text{subject to } \nabla_x f(x,y) = 0, \quad \nabla_x^2 f(x,y) \succeq 0, \\ \qquad\qquad\quad \nabla_y f(x,y) = 0, \quad \nabla_y^2 f(x,y) \preceq 0, \\ \qquad\qquad\quad f(x,y) - f_{r-1} - \delta \geqslant 0. \end{cases} \tag{3.4}$$

Analogously, suppose (x'', y'') is a minimizer of (3.4). According to the second-order sufficient optimality condition, if $\nabla_x^2 f(x'', y'') \succ 0$ and $\nabla_y^2 f(x'', y'') \prec 0$, then (x'', y'') is a local saddle point of $f(x, y)$ and f_r is corresponding local saddle point value. Otherwise, according to the inequality (1.1), if x'' is a minimizer of

$$\begin{cases} \min & f(x, y'') \\ \text{subject to} & \nabla_x f(x, y'') = 0, \quad \nabla_x^2 f(x, y'') \succeq 0, \\ & f(x, y'') - f_{r-1} - \delta \geqslant 0, \\ & \|x - x''\|^2 \leqslant \tau \end{cases} \tag{3.5}$$

for some small $\tau > 0$, and y'' is a maximizer of

$$\begin{cases} \max & f(x'', y) \\ \text{subject to} & \nabla_y f(x'', y) = 0, \quad \nabla_y^2 f(x'', y) \preceq 0, \\ & f(x'', y) - f_{r-1} - \delta \geqslant 0, \\ & \|y - y''\|^2 \leqslant \tau \end{cases} \tag{3.6}$$

for some small $\tau > 0$, then (x'', y'') is a local saddle point of $f(x, y)$, and f_r is corresponding local saddle point value. If $\tau > 0$ is very small, but x'' is not a minimizer of the problem (3.5) or y'' is not a maximizer of the problem (3.6), it is mostly open how to identify if it is a local saddle point or not.

We assume that f_{r-1} ($r \geqslant 2$) is obtained. In order to avoid losing local saddle points arising from inappropriate δ, we introduce a new maximization problem

$$\begin{cases} f_r^\delta := \max f(x, y) \\ \text{subject to} \quad \nabla_x f(x, y) = 0, \quad \nabla_x^2 f(x, y) \succeq 0, \\ \qquad\qquad\quad \nabla_y f(x, y) = 0, \quad \nabla_y^2 f(x, y) \preceq 0, \\ \qquad\qquad\quad f_{r-1} + \delta - f(x, y) \geqslant 0. \end{cases} \tag{3.7}$$

It is obvious that $f_r^\delta \geqslant f_{r-1}$. Note that if $f_r^\delta > f_{r-1}$, then a smaller positive value for δ is required. Therefore, the criterion for choosing a suitable positive value for δ is $f_r^\delta = f_{r-1}$, which can guarantee that there is no other minimum of the problem (3.1) between f_{r-1} and f_r. We refer to the paper [36] for details.

When we solve the problem (3.2), (3.3), (3.5) and (3.6), a suitable value for τ is very important for checking local saddle points. There is not a good approach for choosing a suitable value for τ, but in general it can not be too tiny. In practice, the value like 0.01 or 0.05 is small enough.

3.2 Lasserre's hierarchy of semidefinite relaxations

The optimization problems (3.1)–(3.7) are solved by Lasserre's hierarchy of relaxations. In this section, we illustrate how to solve the problems (3.1) and (3.4) in detail. The other problems can be solved by same approach, so we do not give details for the other problems. The theory of Lasserre's hierarchy of relaxations can be found in [22, 23].

For convenience, we introduce the following notation,

$$
\begin{aligned}
& g(x,y) := f(x,y) - f_{r-1} - \delta, \quad H_f := \operatorname{diag}(\nabla_x^2 f, -\nabla_y^2 f), \\
& d_f := \lceil \deg(f)/2 \rceil, \ d_g := \lceil \deg(g)/2 \rceil, \\
& d_G := \max\{ \lceil \deg(f_{x_i})/2 \rceil, \lceil \deg(f_{y_j})/2 \rceil, \ i = 1,2,\dots,n, \ j = 1,2,\dots,m \}, \\
& d_H := \max\{ \lceil \deg((H_f)_{ij})/2 \rceil, \ 1 \leqslant i \leqslant m+n, \ 1 \leqslant j \leqslant m+n \}.
\end{aligned}
$$

Applying the Lasserre's hierarchy of relaxations to (3.1), we get the following problem

$$
\begin{cases}
\rho_1^k := \min \ \mathscr{L}_z(f) \\
\text{subject to} \ M_k(z) \succeq 0, \ z_0 = 1, \\
\qquad L_{f_{x_i}}^{(k)}(z) = 0, \ i = 1,2,\dots,n, \\
\qquad L_{f_{y_j}}^{(k)}(z) = 0, \ j = 1,2,\dots,m, \\
\qquad L_{H_f}^{(k)}(z) \succeq 0.
\end{cases}
\tag{3.8}
$$

in which k is called relaxation order, and $k \geqslant \max\{d_f, d_G, d_H\}$. The relaxation problem (3.8) can be reformulated as a semidefinite programming problem (SDP), which can be solved by many SDP solvers using interior point methods, such as `SeDuMi` [42], `SDPT3` [43], etc. According to Lasserre's hierarchy of relaxations, the inequality $\rho_1^k \leqslant f_1$ holds for every k because the feasible region of (3.8) is larger than that of (3.1). However, as the relaxation order k increases, more constraints are added to the relaxation problem (3.8), so the sequence $\{\rho_1^k\}$ is monotonically increasing, namely

$$
\rho_1^k \leqslant \rho_1^{k+1} \leqslant \rho_1^{k+2} \leqslant \cdots \leqslant f_1.
$$

Suppose z^* is a minimizer of (3.8), if z^* has a flat truncation [33], which means there exists a positive integer t such that

$$
\operatorname{rank}(M_t(z^*)) = \operatorname{rank}(M_{t-s}(z^*)), \tag{3.9}
$$

where $\max\{d_f, d_G, d_H\} \leqslant t \leqslant k$, $s := \max\{1, d_G, d_H\}$, then we can get at least $r := \operatorname{rank}(M_k(z^*))$ minimizers of (3.1). According to Theorem 1.1 in [10] or Proposition 2.1 in [36], if z^* is feasible for (3.8) and (3.9), then z^* admits a unique r-atomic measure support in the feasible set of (3.1), which means there exist $\mu_1 > 0, \mu_2 > 0, \dots, \mu_r > 0$ and r distinct points $p_1, p_2, \dots, p_r$ in feasible set of (3.1) such that

$$
z^* = \mu_1 [p_1]_{2k} + \mu_2 [p_2]_{2k} + \cdots + \mu_r [p_r]_{2k},
$$

where $\mu_1 + \mu_2 + \cdots + \mu_r = 1$. If the rank condition (3.9) holds, then $f_1 = \rho_1^k$ and all points $p_1, p_2, \dots, p_r$ are the minimizers of (3.1). They can be obtained by solving some singular value decompositions and eigenvalue problems. One can refer to [18] for more details.

The dual problem of (3.8) is

$$\begin{cases} \gamma_1^k := \max\ \gamma \\ \text{subject to}\quad f-\gamma \in \mathrm{I}_{2k}(\nabla f) + \mathrm{Q}_k(H_f). \end{cases} \tag{3.10}$$

The problem (3.10) is also equivalent to a semidefinite programming (SDP) problem. In terms of weak duality, the inequality $\gamma_1^k \leqslant \rho_1^k$ holds for every k. When the Slater condition holds for the problem (3.8), i.e., there exists an interior point in the feasible set of (3.8), then (3.10) achieves its optimal value, and $\gamma_1^k = \rho_1^k$ for every k. Besides, the feasible region of (3.10) will expand as k increases, so the sequence $\{\gamma_1^k\}$ is monotonically increasing, that is

$$\gamma_1^k \leqslant \gamma_1^{k+1} \leqslant \gamma_1^{k+2} \leqslant \cdots \leqslant f_1.$$

If the archimedean condition holds [33], i.e., there exists $\xi(x,y) \in \mathrm{I}_{2k}(\nabla f) + \mathrm{Q}_k(H_f)$ for some k such that $\xi(x,y) \geqslant 0$ defines a compact set in (x, y), then $\gamma_1^k \to f_1$ as $k \to +\infty$.

Applying Lasserre's hierarchy of relaxations to (3.4), we get the following optimization problem

$$\begin{cases} \rho_r^k := \min\ \mathscr{L}_z(f) \\ \text{subject to}\quad M_k(z) \succeq 0,\ \ z_0 = 1, \\ \qquad\qquad\quad L_{f_{x_i}}^{(k)}(z) = 0,\ i = 1, 2, \ldots, n, \\ \qquad\qquad\quad L_{f_{y_j}}^{(k)}(z) = 0,\ j = 1, 2, \ldots, m, \\ \qquad\qquad\quad L_{H_f}^{(k)}(z) \succeq 0,\ \ L_g^{(k)}(z) \succeq 0, \end{cases} \tag{3.11}$$

where $k \geqslant \max\{d_f, d_G, d_H, d_g\}$. Likewise, if a minimizer $\tilde{z}^*$ of (3.11) has a flat truncation [33], i.e., there exists a positive integer $\tilde{t}$ such that

$$\operatorname{rank}(M_{\tilde{t}}(\tilde{z}^*)) = \operatorname{rank}(M_{\tilde{t}-\tilde{s}}(\tilde{z}^*)), \tag{3.12}$$

where $\max\{d_f, d_G, d_H, d_g\} \leqslant \tilde{t} \leqslant k$, and $\tilde{s} := \max\{1, d_G, d_H, d_g\}$, then we can get at least $r := \operatorname{rank}(M_k(z^*))$ minimizers of (3.4).

The dual problem of (3.11) is

$$\begin{cases} \gamma_r^k := \max\ \gamma \\ \text{subject to}\quad f-\gamma \in \mathrm{I}_{2k}(\nabla f) + \mathrm{Q}_k(g) + \mathrm{Q}_k(H_f). \end{cases} \tag{3.13}$$

For the above problems (3.11) and (3.13), we can obtain some conclusions similar to (3.8) and (3.10).

3.3 An algorithm

We now give an algorithm for computing local saddle points for unconstrained polynomial optimization. It can detect the existence of local saddle points. If there exist several distinct local saddle point values, we can get them from the smallest one to the largest one.

Algorithm 3.1 Let $\delta_0 > 0, \tau > 0, \mathcal{S} := \emptyset, r := 1$.

Step 1 If the problem (3.1) is infeasible, then there is no local saddle point and stop. Otherwise, we get the minimum f_1 and the set of minimizers $\mathcal{C}_1$.

Step 2 For a point $(x'_{1,i}, y'_{1,i}) \in \mathcal{C}_1$, if $\nabla_x^2 f(x'_{1,i}, y'_{1,i}) \succ 0$ and $\nabla_y^2 f(x'_{1,i}, y'_{1,i}) \prec 0$, then let $\mathcal{S} := \mathcal{S} \cup \{(x'_{1,i}, y'_{1,i})\}$, and continue to verify the rest of points in $\mathcal{C}_1$. Otherwise, go to Step 3.

Step 3 Solve the problem (3.2) replaced y' with $y'_{1,i}$ and get the minimum η_1. If $f(x'_{1,i}, y'_{1,i}) > \eta_1$, then $(x'_{1,i}, y'_{1,i})$ is not a local saddle point. Update $\mathcal{C}_1 := \mathcal{C}_1 \backslash \{(x'_{1,i}, y'_{1,i})\}$ and go back to Step 2. Otherwise, go to Step 4.

Step 4 Solve the problem (3.3) replaced x' with $x'_{1,i}$ and get the maximum ν_1. If $f(x'_{1,i}, y'_{1,i}) < \nu_1$, then $(x'_{1,i}, y'_{1,i})$ is not a local saddle point. Update $\mathcal{C}_1 := \mathcal{C}_1 \backslash \{(x'_{1,i}, y'_{1,i})\}$ and go back to Step 2. Otherwise, let $\mathcal{S} := \mathcal{S} \cup \{(x'_{1,i}, y'_{1,i})\}$, and go back to Step 2 to verify the rest of points in $\mathcal{C}_1$. If all points in $\mathcal{C}_1$ are verified, then let $r := r + 1$, $\delta := \delta_0$, and go to Step 5.

Step 5 Solve the problem (3.7), and get the maximum f_r^δ, then go to Step 6.

Step 6 If $f_r^\delta > f_{r-1}$, then let $\delta := \delta/2$ and go back to Step 5. If $f_r^\delta = f_{r-1}$, then go to Step 7.

Step 7 If the problem (3.4) is infeasible, then there is no local saddle point value that is not less than f_{r-1} and stop. Otherwise, we get the minimum f_r and the set of minimizers $\mathcal{C}_r$.

Step 8 For a point $(x'_{r,i}, y'_{r,i}) \in \mathcal{C}_r$, if $\nabla_x^2 f(x'_{r,i}, y'_{r,i}) \succ 0$ and $\nabla_y^2 f(x'_{r,i}, y'_{r,i}) \prec 0$, then let $\mathcal{S} := \mathcal{S} \cup \{(x'_{r,i}, y'_{r,i})\}$, and continue to verify the rest of points in $\mathcal{C}_r$. Otherwise, go to Step 9.

Step 9 Solve the problem (3.5) replaced y'' with $y'_{r,i}$ and get the minimum η_r. If $f(x'_{r,i}, y'_{r,i}) > \eta_r$ then $(x'_{r,i}, y'_{r,i})$ is not a local saddle point. Update $\mathcal{C}_r := \mathcal{C}_r \backslash \{(x'_{r,i}, y'_{r,i})\}$ and go back to Step 8. Otherwise, go to Step 10.

Step 10 Solve the problem (3.6) replaced x'' with $x'_{r,i}$ and get the maximum ν_r. If $f(x'_{r,i}, y'_{r,i}) < \nu_r$, then $(x'_{r,i}, y'_{r,i})$ is not a local saddle point. Update $\mathcal{C}_r := \mathcal{C}_r \backslash \{(x'_{r,i}, y'_{r,i})\}$ and go back to Step 8. Otherwise, let $\mathcal{S} := \mathcal{S} \cup \{(x'_{r,i}, y'_{r,i})\}$, then go back to Step 8 to verify the rest of points in $\mathcal{C}_r$. If all points in $\mathcal{C}_r$ are verified, then let $r := r + 1$, $\delta := \delta_0$, and go back to Step 5.

In Step 1, we compute the smallest minimum f_1 and the set of minimizers $\mathcal{C}_1$ via the second-order necessary optimality condition if the problem (3.1) is feasible. In Step 2–4, we use the second-order sufficient optimality condition and the definition of local saddle point to verify whether each minimizer in $\mathcal{C}_1$ is a local saddle point or not. In Step 5–6, solving the problem (3.4) can obtain a suitable value for δ. Further, by Step 7, we obtain the minimum f_r and the set of minimizers $\mathcal{C}_r$ of the problem (3.4) if the problem (3.4) is feasible. In Step 8–10, we verify whether each point in $\mathcal{C}_r$ is a local saddle point or not by the second-order sufficient optimality condition and the definition of local saddle point.

4 Convergence properties

The first result aims to prove that a polynomial has finitely many local saddle point values.

Proposition 4.1 *If a polynomial f(x, y) has local saddle points, then it has finitely many local saddle point values.*

Proof If (x^*, y^*) is a local saddle point of the polynomial $f(x, y)$, then

$$\nabla f(x^*, y^*) = 0.$$

So (x^*, y^*) is a critical point of $f(x, y)$, and $f(x^*, y^*)$ is a critical value. In terms of the proof of Theorem 8 in [32] or Lemma 3.2 in [34], the polynomial $f(x, y)$ has finitely many critical values. Meanwhile, the local saddle point (x^*, y^*) also satisfies

$$\nabla_x^2 f(x^*, y^*) \succeq 0, \quad \nabla_y^2 f(x^*, y^*) \preceq 0,$$

so the set of local saddle point values of $f(x, y)$ is finite. □

The finite termination of Algorithm 3.1 is given as follows.

Theorem 4.2 *For the polynomial $f(x, y)$, let $\mathcal{C}_r$ be the minimizers of (3.1) for $r = 1$, and the minimizers of (3.4) for $r \geqslant 2$. Let $\mathcal{S}_r$ be the set of points in $\mathcal{C}_r$ such that $f(x^*_{r,i}, y'_{r,i}) = f(x'_{r,i}, y'_{r,i})$ and $f(x'_{r,i}, y^*_{r,i}) = f(x'_{r,i}, y'_{r,i})$ for $r \geqslant 1$. If $\nabla f(x, y) = 0, \nabla_x^2 f(x, y) \succeq 0, \nabla_y^2 f(x, y) \preceq 0$ has finitely many real solutions, then Algorithm 3.1 will stop after finitely many iterations. If $\mathcal{S}_r$ is not empty, then each point $(x^*, y^*) \in \mathcal{S}_r$ is a local saddle point of $f(x, y)$ and $f(x^*, y^*)$ is a local saddle point value.*

Proof If the problem (3.1) is infeasible, i.e., the set $\mathcal{C}_1$ is empty, then the original problem does not have a local saddle point. If there exists a positive integer $r \geqslant 1$ make $\mathcal{C}_r$ nonempty, according to the assumption, then these sets $\mathcal{C}_r$ are always finite. Therefore, the algorithm will terminate after finitely many iterations. If $\mathcal{S}_r$ is not empty, it means each $(x^*, y^*) \in \mathcal{S}_r$ is feasible for optimization problems both (3.2) and (3.3) for $r = 1$ or both (3.5) and (3.6) for $r \geqslant 2$. In terms of the inequality (1.1), each point (x^*, y^*) in $\mathcal{S}_r$ is a local saddle point of the polynomial $f(x, y)$, and $f(x^*, y^*)$ is a local saddle point value. □

Remark For a generic polynomial $f(x, y)$, the polynomial system $\nabla f(x, y) = 0$ has finitely many complex solutions, thus Algorithm 3.1 will have finite convergence [38, Theorem 3.3].

Some properties related to the relaxation (3.8), its dual (3.10) and the problem (3.1) are listed as follows.

Theorem 4.3 *Let f_1, ρ_1^k and γ_1^k be the optimal value of* (3.1), (3.8) *and* (3.10), *respectively. Suppose* $\mathrm{I}(\nabla f) + \mathrm{Q}(H_f)$ *is archimedean, then*

(i) *If* (3.8) *is infeasible, then* (3.1) *is infeasible*;
(ii) *If* $\mathrm{V}_{\mathbb{R}}(\nabla f)$ *is finite, and* (3.1) *is infeasible, then* (3.10) *is unbounded for all k big enough and* (3.8) *is infeasible*;
(iii) *If* $\mathrm{V}_{\mathbb{R}}(\nabla f)$ *is finite, and* (3.1) *is feasible, then* $\rho_1^k = \gamma_1^k = f_1$ *for some k*;
(iv) *If* $\mathrm{V}_{\mathbb{R}}(\nabla f)$ *is finite, and* (3.1) *is feasible, then all minimizers of* (3.8) *hold the flat truncation condition* (3.9) *with respect to the feasible set of* (3.1) *for all k big enough.*

Proof (i) The conclusion is obvious, because (3.8) is a relaxation problem of (3.1).

(ii) Because $\mathrm{V}_{\mathbb{R}}(\nabla f)$ is finite, $\mathrm{V}_{\mathbb{R}}(\nabla f)$ is compact, then the ideal $\mathrm{I}(\nabla f)$ is archimedean. Since $-\|(\nabla f)\|^2 \geqslant 0$ defines a compact set in $\mathbb{R}^{m+n}$. If (3.1) is infeasible, then $-H_f$ is non-negative semidefinite matrix for all $x \in \mathrm{V}_{\mathbb{R}}(\nabla f)$. According to Corollary 3.16 in [21], we can get

$$-1 \in \mathrm{I}(\nabla f) + \mathrm{Q}(H_f).$$

Hence, for all k big enough, (3.10) is unbounded, by weak duality, which means (3.8) is infeasible.

(iii) When $\mathrm{V}_{\mathbb{R}}(\nabla f)$ is finite, $\mathrm{V}_{\mathbb{R}}(\nabla f)$ can be written as $\{w_1, w_2, \dots, w_L\}$, where L is a positive integer and $w_i \neq w_j$ for all $i \neq j$. Let $t_1, t_2, \dots, t_L$ be the interpolating polynomials such that $t_i(w_j) = 0$ for $i \neq j$ and $t_i(w_j) = 1$ for $i = j$.

For each w_i, if $f(w_i) - f_1 \geqslant 0$, let

$$Q_i := (f(w_i) - f_1)t_i^2.$$

If $f(w_i) - f_1 < 0$, then H_f is non-positive semidefinite matrix. Hence, there exists a eigenvalue $\lambda < 0$ of H_f, and its eigenvector is denoted by v. Let $h(x, y) := v^{\mathrm{T}} H_f v$ and

$$Q_i := \left(\frac{f(w_i) - f_1}{h(w_i)}\right) h t_i^2.$$

Hence, each $Q_i \in \mathrm{Q}(H_f)$. Let $Q := Q_1 + Q_2 + \cdots + Q_L$, so $Q \in \mathrm{Q}_{N_1}(H_f)$ for some $N_1 > 0$. The polynomial

$$p := f - f_1 - Q \in \{b \in \mathbb{R}[x, y] \mid b(u, v) = 0,\ \forall (u, v) \in \mathrm{V}_{\mathbb{R}}(\nabla f)\},$$

By Corollary 4.1.8 in [6], there exist an integer $l > 0$ and $q \in \Sigma[x, y] \subseteq \mathrm{Q}(H_f)$ such that

$$p^{2l} + q \in \mathrm{I}(\nabla f).$$

Applying Lemma 2.1 in [35] to p, q, with ∇f, H_f and any $c \geqslant 1/2l$, then there exists $N^* > N_1$ such that for all $\epsilon > 0$, $p + \epsilon = \phi_\epsilon + \theta_\epsilon$, where $\phi_\epsilon \in \mathrm{I}_{2N^*}(\nabla f)$, $\theta_\epsilon \in \mathrm{Q}_{N^*}(H_f)$. So we have

$$f - (f_1 - \epsilon) = \phi_\epsilon + (\theta_\epsilon + Q), \quad \theta_\epsilon + Q \in \mathrm{Q}_{N^*}(H_f),$$

which means that for arbitrary $\epsilon > 0$, $\gamma_1^k = f_1 - \epsilon$ is feasible in (3.10) for $k = N^*$. $\gamma_1^k \leqslant f_1$ for arbitrary k and $\{\dots, \gamma_1^k, \gamma_1^{k+1}, \gamma_1^{k+2}, \dots\}$ is a monotonically increasing sequence, so $\gamma_1^k = f_1$ as $k \to +\infty$. According to $\gamma_1^k \leqslant \rho_1^k \leqslant f_1$, so $\rho_1^k = \gamma_1^k = f_1$ for some $k \in \mathbb{N}^+$.

(iv) Because $\mathrm{V}_{\mathbb{R}}(\nabla f)$ is finite, according to Proposition 4.6 in [24], there exists k big enough, each z is feasible for (3.8), the truncation $(z)_{2k}$ is flat with respect to $\nabla f = 0$. Because $d_G = d_H + 1$, and $L_{H_f}^{(k)}(z) \succeq 0$, $(z)_{2k}$ is also flat with respect to $H_f \succeq 0$, so the conclusion is proved. □

We will show some results of the relaxation (3.11), its dual (3.13) and the problem (3.4).

Theorem 4.4 *Let f_r, ρ_r^k and γ_r^k be the optimal value of* (3.4), (3.11) *and* (3.13), *respectively. Suppose $\mathrm{I}(\nabla f) + \mathrm{Q}(g) + \mathrm{Q}(H_f)$ is archimedean, then*

(i) *If* (3.11) *is infeasible, then* (3.4) *is infeasible*;
(ii) *If $\mathrm{V}_{\mathbb{R}}(\nabla f) \cap \{(x, y) \mid g(x, y) \geqslant 0\}$ is finite, and* (3.4) *is infeasible, then* (3.13) *is unbounded for all k big enough and* (3.11) *is infeasible*;
(iii) *If $\mathrm{V}_{\mathbb{R}}(\nabla f) \cap \{(x, y) \mid g(x, y) \geqslant 0\}$ is finite, and* (3.4) *is feasible, then $\rho_r^k = \gamma_r^k = f_r$ for some k*;
(iv) *If $\mathrm{V}_{\mathbb{R}}(\nabla f) \cap \{(x, y) \mid g(x, y) \geqslant 0\}$ is finite, and* (3.4) *is feasible, then all minimizers of* (3.11) *hold the flat truncation condition* (3.12) *with respect to the feasible set of* (3.4) *for all k big enough.*

Proof (i) The conclusion is obvious, because (3.11) is a relaxation problem of (3.4).

(ii) Because $\mathrm{V}_{\mathbb{R}}(\nabla f) \cap \{(x, y) \mid g(x, y) \geqslant 0\}$ is finite, $\mathrm{V}_{\mathbb{R}}(\nabla f) \cap \{(x, y) \mid g(x, y) \geqslant 0\}$ is compact, then $\mathrm{I}(\nabla f) + \mathrm{Q}(g)$ is archimedean. Because (3.4) is infeasible, $-H_f$ is non-negative semidefinite matrix for all $x \in \mathrm{V}_{\mathbb{R}}(\nabla f) \cap \{(x, y) \mid g(x, y) \geqslant 0\}$. According to Corollary 3.16 in [21], we can get

$$-1 \in \mathrm{I}(\nabla f) + \mathrm{Q}(g) + \mathrm{Q}(H_f).$$

Hence, for all k big enough, (3.13) is unbounded, by weak duality, which means (3.11) is infeasible.

(iii) The proof is similar to that of the item (iii) in Theorem 4.3.

(iv) The proof is similar to that of the item (iv) in Theorem 4.3, which can be proved by Remark 4.9 in [24]. □

Table 1 The numerical results of Example 5.1

r	f_r^*	Local saddle points
1	-2.2295	$(-1.2961, -0.6051), (1.2961, 0.6051)$
2	-0.5437	$(-1.1092, 0.7683), (1.1092, -0.7683)$

Table 2 The numerical results of Example 5.2

r	f_r^*	Local saddle points
1	-2.0436	$(-1.2435, 0.8657)$
2	0	$(-0.6958, -0.6958)$
3	2.0436	$(0.8657, -1.2435)$

5 Numerical experiments

In this section, we apply Algorithm 3.1 to solve local saddle points of unconstrained polynomial optimization, and global saddle point is computed by Algorithm 3.1 in [38]. All examples of this section are computed in MATLAB R2016b on a Lenovo laptop with dual core CPU @ 2.5GHz and RAM 4.0 GB. All problems of Lasserre's hierarchy of relaxations are solved by MATLAB software package `YALMIP` [26] and `GloptiPoly 3` [19], in which the semidefinite optimization solver `SeDuMi` [42] is called.

Example 5.1 Consider the function over $(x, y) \in \mathbb{R} \times \mathbb{R}$

$$f(x, y) = -4x^2 + \frac{21}{10}x^4 - \frac{1}{3}x^6 - xy + 4y^2 - 4y^4.$$

Applying Algorithm 3.1, after 2 iterations, local saddle point values f_1^* and f_2^* were found, as shown in Table 1.

Since the problem (3.11) is infeasible for $r = 3$ and $f_3^\delta = f_2^*$, which means f_2^* is the biggest local saddle point value. Hence, we got 2 local saddle point values and 4 local saddle points. Moreover, we got that the example has no global saddle point by Algorithm 3.1 in [38].

Example 5.2 Consider the function over $(x, y) \in \mathbb{R} \times \mathbb{R}$

$$f(x, y) = x^4y^2 - x^2y^4 + x^3y - xy^3 + x - y.$$

Applying Algorithm 3.1, after 3 iterations, local saddle point values f_1^*, f_2^* and f_3^* were found, as shown in Table 2.

Since the problem (3.11) is infeasible for $r = 4$ and $f_4^\delta = f_3^*$, which means f_3^* is the biggest local saddle point value. Hence, we got 3 local saddle point values and 3 local saddle points. Moreover, we got that the example has no global saddle point by Algorithm 3.1 in [38].

Example 5.3 Consider the function over $(x, y) \in \mathbb{R}^2 \times \mathbb{R}^2$

$$f(x,y) = x_1^3y_1 + x_2^3y_2 - x_1y_1^3 - x_2y_2^3 + x_1^2y_1 + x_2^2y_2 - x_1y_1^2 - x_2y_2^2 + x_1y_1 + x_2y_2 + x_1 + x_2 - y_1 - y_2.$$

Applying Algorithm 3.1, after 1 iteration, we got local saddle point value $f_1^* \approx 1.9346$ and local saddle point

$$(-0.7189, -0.7189, -1.2503, -1.2503).$$

Since the problem (3.11) is infeasible for $r = 2$ and $f_2^\delta = f_1^*$, which means f_1^* is the biggest local saddle point value. Hence, we got 1 local saddle point value and 1 local saddle point. Moreover, we got that the example has no global saddle point by Algorithm 3.1 in [38].

Example 5.4 Consider the function over $(x, y) \in \mathbb{R}^2 \times \mathbb{R}^2$

$$f(x,y) = x_1^6 + x_2^6 - y_1^6 - y_2^6 - 5(x_1^3x_2^2 + x_2^2y_1^3 + y_1x_2^4) + 6x_1^2x_2^2 + 6y_1^3y_2 + 6x_1x_2y_1y_2 - 7(x_1x_2y_1 + x_2y_1y_2) + (x_1 + x_2 + y_1 + y_2 - 1)^2.$$

Applying Algorithm 3.1, after 2 iterations, local saddle point values f_1^* and f_2^* were found, as shown in Table 3.

Since the problem (3.11) is infeasible for $r = 3$ and $f_3^\delta = f_2^*$, which means f_2^* is the biggest local saddle point value. Hence, we got 2 local saddle point values and 2 local saddle points. Moreover, we got the global saddle point

$$(-0.7087, 0.5995, -1.6767, -1.3325)$$

and its saddle point value 18.2221 by Algorithm 3.1 in [38].

Example 5.5 Consider the function over $(x, y) \in \mathbb{R}^3 \times \mathbb{R}^3$

$$f(x,y) = (x_1 + x_2 + x_3 + y_1 + y_2 + y_3)^2 + x_1x_2y_1y_2 + x_2x_3y_2y_3 + x_1x_3y_1y_3 + x_1^2y_3^2 + x_2^2y_1^2 + x_3^2y_2^2.$$

Applying Algorithm 3.1, since the problem (3.8) is infeasible for $r = 1$, we got that the example has no local saddle point.

Example 5.6 Consider the function over $(x, y) \in \mathbb{R}^3 \times \mathbb{R}^3$

Table 3 The numerical results of Example 5.4

r	f_r^*	Local saddle points
1	18.2221	$(-0.7087, 0.5995, -1.6767, -1.3325)$
2	20.1531	$(1.0880, -0.1122, -1.6630, -1.4138)$

$$f(x,y) = \sum_{i=1}^{3} x_i^4 + \sum_{i=1}^{3} (x_i - 1)^2 - \sum_{i=1}^{3} y_i^4 - \sum_{i=1}^{3} (y_i - 1)^2 + \\ x_3^2 y_1 y_2 + 2x_2^2 y_1 y_3 + 3x_1^2 y_2 y_3 - y_3^2 x_1 x_2 - 5y_2^2 x_1 x_3 - 4y_1^2 x_2 x_3.$$

Applying Algorithm 3.1, after 1 iteration, we got local saddle point value $f_1^* \approx -0.3681$ and local saddle point

$$(0.5785, 0.6043, 0.6936, 0.4363, 0.4233, 0.6357).$$

Since the problem (3.11) is infeasible for $r = 2$ and $f_2^\delta = f_1^*$, which means f_1^* is the biggest local saddle point value. Hence, we got 1 local saddle point value and 1 local saddle point. Moreover, we got that the local saddle point is just the global saddle point by Algorithm 3.1 in [38].

6 Conclusions and discussions

The paper concentrates on detecting local saddle points of unconstrained polynomial optimization. An algorithm based on second-order optimality conditions is proposed for getting local saddle points, in which all polynomial optimization problems of the algorithm are solved by Lasserre's hierarchy of relaxations. The algorithm can detect the nonexistence if a polynomial does not have local saddle points. When a polynomial has several local saddle points, the algorithm can get them from the smallest local saddle point value to the largest one. Besides, we give the finite termination of the algorithm.

Finally, for future work, one can consider the problem: for a polynomial on constraints, how can we get its saddle points?

Acknowledgements The authors would like to thank the editors and anonymous referees for the valuable advice. Wenjie Zhao was supported by Postgraduate Scientific Research Innovation Project of Hunan Province (CX20210605). Guangming Zhou was supported by Natural Science Foundation of China (12071399) and Key Projects of Hunan Provincial Education Department (18A048).

Data Availability Statement Data sharing is not applicable to this article as no datasets are generated or analyzed during the study of this article.

References

1. Adolphs, L., Daneshmand, H., Lucchi, A., Hofmann, T.: Local saddle point optimization: A curvature exploitation approach. Presented at the (2019)
2. Bai, Z., Golub, G.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**(1), 1–23 (2007)
3. Tal, A., Ghaoui, L., Nemirovski, A.: Robust optimization, vol. 28. Princeton University Press (2009)
4. Benzi, M., Gander, M., Golub, G.: Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems. BIT Numer. Math. **43**(5), 881–900 (2003)
5. Bertsekas, D., Nedic, A., Ozdaglar, A.: Convex Analysis and Optimization. Athena Scientific (2003)
6. Bochnak, J., Coste, M., Roy, M.: Real algebraic geometry, vol. 36. Springer (1998)

7. Botchev, M., Golub, G.: A class of nonsymmetric preconditioners for saddle point problems. SIAM J. Matrix Anal. Appl. **27**(4), 1125–1149 (2006)
8. Chen, Y., Lan, G., Ouyang, Y.: Optimal primal-dual methods for a class of saddle point problems. SIAM J. Optim. **24**(4), 1779–1814 (2014)
9. Cox, B., Juditsky, A., Nemirovski, A.: Decomposition techniques for bilinear saddle point problems and variational inequalities with affine monotone operators. J. Optim. Theory Appl. **172**(2), 402–435 (2017)
10. Curto, R., Fialkow, L.: Truncated k-moment problems in several variables. J. Oper. Theory, pp 189–226 (2005)
11. Dauphin, Y., Pascanu, R., Gulcehre, C., et al.: Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In: Advances in Neural Information Processing Systems, pp. 2933–2941 (2014)
12. Dollar, H., Wathen, A.: Approximate factorization constraint preconditioners for saddle-point matrices. SIAM J. Sci. Comput. **27**(5), 1555–1572 (2006)
13. Elman, H., Golub, G.: Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J. Numer. Anal. **31**(6), 1645–1661 (1994)
14. Fialkow, L., Nie, J.: The truncated moment problem via homogenization and flat extensions. J. Funct. Anal. **263**(6), 1682–1700 (2012)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
16. Greif, C., Schötzau, D.: Preconditioners for saddle point linear systems with highly singular $(1, 1)$ blocks. ETNA, Special Volume on Saddle Point Problems **22**, 114–121 (2006)
17. He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imag. Sci. **5**(1), 119–149 (2012)
18. Henrion, D., Lasserre, J.: Detecting global optimality and extracting solutions in gloptipoly. In: Positive Polynomials in Control, pp. 293–310. Springer (2005)
19. Henrion, D., Lasserre, J., Löfberg, J.: Gloptipoly 3: moments, optimization and semidefinite programming. Optim. Methods Softw. **24**(4–5), 761–779 (2009)
20. Kallio, M., Ruszczynski, A.: Perturbation methods for saddle point computation. (1994)
21. Klep, I., Schweighofer, M.: Pure states, positive matrix polynomials and sums of Hermitian squares. Indiana Univ. Math. J. **59**(3), 857–874 (2010)
22. Lasserre, J.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2001)
23. Lasserre, J.: Moments, positive polynomials and their applications, vol. 1. World Scientific (2010)
24. Lasserre, J., Laurent, M., Rostalski, P.: Semidefinite characterization and computation of zero-dimensional real radical ideals. Found. Comput. Math. **8**(5), 607–647 (2008)
25. Brown, K., Shoham, Y.: Essentials of game theory: a concise multidisciplinary introduction. Synth. Lect. Artificial Intell. Mach. Learn. **2**(1), 1–88 (2008)
26. Löfberg, J.: Yalmip: A toolbox for modeling and optimization in matlab. Presented at the (2004)
27. Mamer, J., Schilling, K.: Finite approximations to a zero-sum game with incomplete information. Int. J. Game Theory **19**(1), 101–106 (1990)
28. Megahed, A.: A differential game related to terrorism: min-max zero-sum two persons differential game. Neural Comput. Appl. **30**(3), 865–870 (2018)
29. Murty, K., Kabadi, S.: Some NP-complete problems in quadratic and nonlinear programming. Math. Program. **39**(2), 117–129 (1987)
30. Nedić, A., Ozdaglar, A.: Subgradient methods for saddle-point problems. J. Optim. Theory Appl. **142**(1), 205–228 (2009)
31. Nemirovski, A.: Prox-method with rate of convergence $o(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM J. Optim. **15**(1), 229–251 (2004)
32. Nie, J., Demmel, J., Sturmfels, B.: Minimizing polynomials via sum of squares over the gradient ideal. Math. Program. **106**(3), 587–606 (2006)
33. Nie, J.: Certifying convergence of Lasserre's hierarchy via flat truncation. Math. Program. **142**(1–2), 485–510 (2013)
34. Nie, J.: An exact Jacobian SDP relaxation for polynomial optimization. Math. Program. **137**(1–2), 225–255 (2013)
35. Nie, J.: Polynomial optimization with real varieties. SIAM J. Optim. **23**(3), 1634–1646 (2013)

36. Nie, J.: The hierarchy of local minimums in polynomial optimization. Math. Program. **151**(2), 555–583 (2015)
37. Nie, J.: Linear optimization with cones of moments and nonnegative polynomials. Math. Program. **153**(2), 247–274 (2015)
38. Nie, J., Yang, Z., Zhou, G.: The saddle point problem of polynomials. arXiv preprint, p. 1809.01218 (2018)
39. Pascanu, R., Dauphin, Y., Ganguli, S., Bengio, Y.: On the saddle point problem for non-convex optimization. Presented at the (2014)
40. Rozložník, M., Simoncini, V.: Krylov subspace methods for saddle point problems with indefinite preconditioning. SIAM J. Matrix Anal. Appl. **24**(2), 368–391 (2002)
41. Singh, S., Kearns, M., Mansour, Y.: Nash convergence of gradient dynamics in general-sum games. In *UAI*, pp. 541–548 (2000)
42. Sturm, J.: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**(1–4), 625–653 (1999)
43. Toh, K., Todd, M., Tütüncü, R.: On the implementation and usage of SDPT3–a matlab software package for semidefinite-quadratic-linear programming, Version 4.0, pp. 715–754, Springer (2012)
44. Vasilyev, F., Khoroshilova, E., Antipin, A.: An extragradient method for finding the saddle point in an optimal control problem. Mosc. Univ. Comput. Math. Cybern. **34**(3), 113–118 (2010)
45. Zhou, G., Wang, Q., Zhao, W.: Saddle points of rational functions. Comput. Optim. Appl. **75**, 817–832 (2020)