A parametric representation of totally mixed Nash equilibria

Gabriela Jeronimo\textsuperscript{a,c,*}, Daniel Perrucci\textsuperscript{a}, Juan Sabia\textsuperscript{a,b,c}

\textsuperscript{a} Departamento de Matemática, FCEN, Universidad de Buenos Aires, Ciudad Universitaria, (1428) Buenos Aires, Argentina
\textsuperscript{b} Departamento de Ciencias Exactas, CBC, Universidad de Buenos Aires, Ciudad Universitaria, (1428) Buenos Aires, Argentina
\textsuperscript{c} CONICET, Argentina

\textbf{A R T I C L E I N F O}

\textbf{Article history:}
Received 31 October 2007
Received in revised form 23 April 2009
Accepted 16 June 2009

\textbf{Keywords:}
Noncooperative game theory
Nash equilibria
Polynomial equation solving
Multihomogeneous resultants
Complexity

\textbf{A B S T R A C T}

We present an algorithm to compute a parametric description of the totally mixed Nash equilibria of a generic game in normal form with a fixed structure. Using this representation, we also show an algorithm to compute polynomial inequality conditions under which a game has the maximum possible number of this kind of equilibria. Then, we present symbolic procedures to describe the set of isolated totally mixed Nash equilibria of an arbitrary game and to compute, under certain general assumptions, the exact number of these equilibria. The complexity of all these algorithms is polynomial in the number of players, the number of each player's strategies and the generic number of totally mixed Nash equilibria of a game with the considered structure.

\section{1. Introduction}

Noncooperative game theory is used to model and analyze strategic interaction situations. Among its most outstanding applications, we can mention the fundamental role this theory has played in economics (see, for example, the classical reference book [1]). Moreover, game theory has also been applied to politics, sociology and psychology, and to biology and evolution as well.

One of the main concepts in this theory is that of Nash equilibrium, which consists in a situation in which no player can increase his payoff by unilaterally changing his strategy. Since within this theory the players cannot communicate in order to decide a simultaneous change of strategies, in a Nash equilibrium the game stabilizes. In [2], it is proved that any noncooperative game in normal form has at least one Nash equilibrium. However, the proof is not constructive and does not give any information about the existence of more than one Nash equilibrium. The question posed is how to compute algorithmically Nash equilibria and to determine the number of them in a given game.

Nash equilibria of noncooperative games in normal form can be regarded as real solutions to systems of polynomial equations and inequalities (see, for instance, [3, Chapter 6]). In the case of two players, each equilibrium is the solution of a linear system of equations, and therefore, equilibria may be found exactly by using simplex type algorithms (see, for instance, [4]); however, there is no polynomial time algorithm solving the problem (see [5]). In the general case of a game with more than two players, the polynomials appearing are multilinear. To solve the problem of finding one equilibrium, some numerical methods have been applied successfully (for example, some methods derived from Scarf's algorithm, [6]). Nevertheless, sometimes it is not sufficient to compute only one equilibrium because, depending on the problem to be solved, not all the equilibria of a game are equally interesting and the methods developed to compute only one equilibrium do not allow us to decide beforehand whether it fulfills some additional properties or to compare different equilibria.

A comparative study of different known methods for the computation of all the Nash equilibria of a game may be found in [7]. In [8], a new algorithm solving this problem for generic games by means of homotopy methods is presented, but no complexity bounds are shown (for a recent treatment of numerical methods for solving polynomial systems see the...
book [9]). Regarding implementation, the Gambit software (see [10]) provides some tools for finding Nash equilibria and studying games. In addition, the application of symbolic algorithms solving systems of equations and inequalities over the real numbers (see, for instance [11]) is being studied in this context, motivated by the characterization of the set of all the Nash equilibria of a game as a semi-algebraic set (an example of this fact is the application of quantifier elimination algorithms over the real numbers to compute approximated equilibria in [12]; see also the survey [13]). However, up until now, no significative result had been obtained concerning the adaptation of these algorithms in order to profit from the particular properties of the algebraic systems arising in game theory.

In this paper, we study totally mixed Nash equilibria, that is to say, Nash equilibria in which every player allocates a positive probability to each of his available strategies. Note that a procedure to compute these equilibria can be used as a subroutine to compute all Nash equilibria of the game by recursing over all possible subsets of used strategies.

The aim of this paper is to design symbolic algorithms to describe the set of totally mixed Nash equilibria of a game, either in the generic parametric case or in particular cases, taking into account the multihomogeneity of the polynomials involved in its definition. Our goal is to do so within a complexity polynomial in certain natural invariants associated to the problem, lower than the one that could be obtained by directly applying the known general polynomial equation solving algorithms (see, for instance, [14–22]; also [23] and the references therein).

A key ingredient to achieve the desired complexities in all the algorithms in this paper is the use of straight-line programs (see Section 2.1) to encode the polynomials we work with. This alternative data structure, which comes from numerical analysis, has already been applied in the polynomial equation solving framework yielding a significant reduction in the previously known complexities (see, for instance, [16] and [20] among many other works).

Our first result presents a symbolic method to find a parametric description (a so-called geometric resolution, see Section 2.4 for a definition) of the set of totally mixed Nash equilibria of a generic game with a pre-fixed structure. This method is based on the symbolic procedure for the computation of multihomogeneous resultants with complexity polynomial in the degree and the number of variables of the resultant described in [24] (see also this paper for previous works on resultant computation). We summarize it as follows:

**Theorem I.** There is an algorithm which computes a geometric resolution of the set of totally mixed Nash equilibria of a generic game with \( r \) players having \( n_1 + 1, \ldots, n_r + 1 \) pure strategies respectively, within complexity \( O(\delta^{2}n^{8}(\log(n) + \log(\delta))) \), where \( \delta \) is the number of totally mixed Nash equilibria of a generic game with the given structure and \( n = \sum_{1 \leq i \leq r} n_i \).

A more precise statement of our result will be given in Theorem 1. Note that the complexity of the algorithm is polynomial in the number of players, the number of each player's pure strategies and the number of totally mixed Nash equilibria of a generic game with the given structure. These are the invariants we mentioned above, in terms of which we will express the complexity of all the algorithms in this paper.

There are already known probabilistic algorithms using straight-line programs that could be adapted in order to compute a geometric resolution of the set of totally mixed Nash equilibria of a generic game: for instance, the algorithm solving parametric polynomial equation systems by means of deformation techniques given in [25], or the one developed in [26] in the bihomogeneous setting, which takes into account the structure of the polynomial equations within a similar approach. However, the more ad hoc procedure we present in Theorem 1, based on multihomogeneous resultant computations, is deterministic and works within the same or even better complexities.

Our following result is concerned with the characterization of games with the maximum possible finite number of totally mixed Nash equilibria for the considered structure. The existence of such games was proved in [27]; however, no characterization has been provided. Using the description obtained by the algorithm in Theorem 1, we give an algorithm to compute a finite set of polynomial inequalities in the payoff values under which a game with the given pre-fixed structure will have the maximum number of such equilibria (for a more precise statement, see Theorem 2):

**Theorem II.** Under the same notation as in Theorem 1, there is a family of \( n \delta + 1 \) polynomials with rational coefficients \( S_{ij}^{(h)} \), \( 1 \leq i \leq r, \ 1 \leq j \leq n_i \) and \( 1 \leq h \leq \delta \), in the payoff values of a game with \( r \) players with \( n_1 + 1, \ldots, n_r + 1 \) pure strategies such that for every payoff vector \( c \) satisfying the conditions

\[
S_{ij}(c) \neq 0, \quad S_{ij}^{(h)}(c) > 0 \quad (1 \leq i \leq r, \ 1 \leq j \leq n_i, \ 1 \leq h \leq \delta),
\]

the associated game has \( \delta \) totally mixed Nash equilibria. The polynomials \( S_{ij} \) and \( S_{ij}^{(h)} \) have degrees bounded by \( 4\delta^{2}n^{2} \) and can be computed within complexity \( O(\delta^{2}(n\delta^{2} + L)) \), from a straight-line program of length \( L \) encoding a geometric resolution of the set of totally mixed Nash equilibria of the generic game.

A further parametrical classification of games according to their set of Nash equilibria could be achieved using the algorithms in [28], but they rely on the more expensive Gröbner bases approach. In the same spirit, the work in [29] might be adapted to handle this problem.

After analyzing the generic situation, we deal with particular games. In this case, we give algorithms to compute a geometric resolution of a finite set of points including all the isolated (in the complex space) totally mixed Nash equilibria of the game. First, in Theorem 4, we obtain a procedure to achieve this task under some genericity assumptions implying, in particular, that the number of totally mixed Nash equilibria of the game is finite, and then we show how to compute this
number (see Proposition 5). Finally, in Theorem 7, we consider the same problem in the general case, for which we design a probabilistic algorithm. The output of this algorithm enables us to bound the number of isolated equilibria of the game.

This paper is organized as follows:

In Section 2, we introduce some basic notions on game theory and polynomial system solving. In Section 3, we present algorithms for computing a geometric resolution of the totally mixed Nash equilibria of a generic game and for obtaining conditions under which a game has the maximum number of these equilibria. Section 4 deals with the isolated totally mixed Nash equilibria of particular games. In Sections 5 and 6 we make some concluding remarks on complexity and future implementation issues. Finally, two appendices are devoted to proving some complementary results about the algorithmic computation of multihomogeneous resultants and upper bounds for their degrees which are used throughout the paper, implying that the complexities of all the algorithms in this work are polynomial in the number of players, the number of each player’s pure strategies and the number of totally mixed Nash equilibria of a generic game with the considered structure.

2. Preliminaries

2.1. Basic definitions and notation

Throughout this paper $\mathbb{Q}$ denotes the field of rational numbers, $\mathbb{N}$ denotes the set of positive integers and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}.$

If $K$ is a field, we denote an algebraic closure of $K$ by $\overline{K}.$ As usual, the ring of polynomials in the variables $x_1, \ldots, x_n$ with coefficients in $K$ is denoted by $K[x_1, \ldots, x_n].$ For a polynomial $f \in K[x_1, \ldots, x_n]$ we write $\deg f$ to refer to the total degree of $f$ and $\deg_x f$ to refer to the degree of $f$ in the variable $x_i.$

For $n \in \mathbb{N}$ and an algebraically closed field $k,$ we denote by $A^n(k)$ and $P^n(k)$ (or simply by $A^n$ or $P^n$ if the base field is clear from the context) the $n$-dimensional affine space and projective space over $k$ respectively, equipped with their Zariski topologies. We adopt the usual notion of dimension of an algebraic variety $V$ (see for instance $[30, 31]).$

The algorithms we consider in this paper are described by arithmetic networks over the base field $\mathbb{Q}$ (see [32]). The notion of complexity of an algorithm we consider is the number of operations and comparisons over $\mathbb{Q}.$

The objects we deal with are polynomials with coefficients in $\mathbb{Q}.$ Throughout our algorithms we represent each polynomial either as the array of all its coefficients in a pre-fixed order of its monomials (dense form) or by a straight-line program. Roughly speaking, a straight-line program (or slp, for short) over $\mathbb{Q}$ encoding a polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ is a program (an arithmetic circuit) which enables us to evaluate the polynomial $f$ at any given point in $\mathbb{Q}^n.$ The number of instructions in the program is called the length of the slp (for a precise definition we refer to $[33, Definition\ 4.2];$ see also $[34]).$

2.2. Game theory

In this section we present some basic game theory concepts. For a more detailed account on the subject we refer the reader to any standard game theory text such as $[35].$

We consider noncooperative games in normal form; that is to say, games in which there is only one time step at which all the players move simultaneously without communicating among themselves. We will assume that there are $r$ players in the game having $n_1, \ldots, n_r + 1$ different available pure strategies respectively ($n_1, \ldots, n_r \in \mathbb{N}).$

For $i = 1, \ldots, r,$ $c^{(i)} := (c^{(i)}_{j_1 \ldots j_r})_{0 \leq j_k \leq n_k}$ is the given payoff matrix of player $i,$ where $c^{(i)}_{j_1 \ldots j_r}$ is the payoff to player $i$ if, for every $1 \leq k \leq r,$ player $k$ chooses the strategy $x_j$ and $X_i := (x_{i0}, x_{i1}, \ldots, x_{in_i})$ is a vector representing a mixed strategy of the $i$th player, which is a probability distribution on his set of pure strategies (that is to say, $x_i$ is the probability that the $i$th player allocates to his $j$th pure strategy). With these notations, for every $1 \leq i \leq r,$ the payoff to player $i$ if the mixed strategies $X_1, \ldots, X_r$ are played is

$$\pi_i(X_1, \ldots, X_r) := \sum_{0 \leq j_1 \leq n_1} \cdots \sum_{0 \leq j_r \leq n_r} c^{(i)}_{j_1 \ldots j_r} x_{1j_1} \cdots x_{rj_r}. $$

A Nash equilibrium is a vector of mixed strategies such that no player can increase his payoff by changing unilaterally to another strategy while the other players keep their strategies fixed; that is, a vector of mixed strategies $X_1, \ldots, X_r$ satisfying $\pi_i(X_1, \ldots, X_{i-1}, X_i, X_{i+1}, \ldots, X_r) \geq \pi_i(X_1, \ldots, X_{i-1}, X'_i, X_{i+1}, \ldots, X_r)$ for every $1 \leq i \leq r$ and every mixed strategy $X'_i.$ A totally mixed Nash equilibrium is a Nash equilibrium in which each pure strategy is assigned a positive probability, that is, one that satisfies $x_{ij} > 0$ for every $1 \leq i \leq r, 0 \leq j \leq n_i.$

The totally mixed Nash equilibria of an $r$-person game in normal form can be regarded as real solutions to a polynomial equation system (see, for example, $[3, Section\ 6.3]).$ They are the real vectors $(X_1, \ldots, X_r)$ with $X_i := (x_{i0}, \ldots, x_{in_i})$ for every $1 \leq i \leq r$ satisfying:

(a) $x_{ij} > 0$ for $i = 1, \ldots, r$ and $j = 0, \ldots, n_i,$

(b) $\sum_{0 \leq j \leq n_i} x_{ij} = 1$ for $i = 1, \ldots, r,$

(c) $\sum_{j_l \leq n_l} (c^{(i)}_{j_1 \ldots j_l 10j_{l+1} \ldots j_r} - c^{(i)}_{j_1 \ldots j_l 00j_{l+1} \ldots j_r}) x_{1j_1} \cdots x_{l-1j_{l-1}} x_{l+1j_{l+1}} \cdots x_{rj_r} = 0$ for $i = 1, \ldots, r, k = 1, \ldots, n_i,$ where the sum runs over all $j_l \leq n_l$ with $0 \leq j_l \leq n_l$ for every $t \neq i.$

Observe that (c) is a system of $n := n_1 + \cdots + n_r$ multihomogeneous polynomial equations in the $r$ groups of variables $X_1, \ldots, X_r$ with $n_1 + 1, \ldots, n_r + 1$ variables respectively (with degrees $1$ or $0$ with respect to each group) and, therefore, it defines a (possibly empty) projective variety in $\mathbb{P}^{n_1}(\mathbb{C}) \times \cdots \times \mathbb{P}^{n_r}(\mathbb{C}).$ The projective complex solutions to the polynomial
equation system (c) will be called quasi-equilibria of the game [see [7]], and those solutions not lying in any of the infinite hyperplanes \( x_{0i} = 0 \) \((1 \leq i \leq r)\) will be called affine quasi-equilibria of the game.

Every quasi-equilibrium \( \xi \) determines at most one totally mixed Nash equilibrium of the game: for every \( 1 \leq i \leq r \), let \( s_i \) be the sum of the coordinates of \( \xi_i \). If \( s_i \neq 0 \) for every \( 1 \leq i \leq r \), the unique associated representation of \( \xi \) whose coordinates satisfy condition (b) is \( (\xi_1/s_1, \ldots, \xi_\ell/s_\ell) \), and it will be a totally mixed Nash equilibrium if and only if all its coordinates are positive real numbers.

2.3. On the number of solutions to a multihomogeneous system

Let \( r \in \mathbb{N} \). Fix positive integers \( n_1, \ldots, n_r \), and consider \( r \) groups of variables \( \mathcal{X}_j := (x_0, \ldots, x_{n_j}), j = 1, \ldots, r \). We say that \( F \in \mathbb{K}[\mathcal{X}_1, \ldots, \mathcal{X}_r] \) is multihomogeneous of multidegree \( \upsilon := (\upsilon_1, \ldots, \upsilon_r) \), where \( (\upsilon_1, \ldots, \upsilon_r) \in \mathbb{N}_0^r \), if \( F \) is homogeneous of degree \( \upsilon_j \) in the group of variables \( \mathcal{X}_j \) for every \( 1 \leq j \leq r \).

Set \( n := \sum_{j=1}^{r} n_j \). The classical Multihomogeneous Bézout Theorem, which follows from the intersection theory for divisors [see for instance [30, Chapter 4]], states that the set of common zeroes (over an algebraically closed field) in the projective variety \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r} \) of \( n \) generic multihomogeneous polynomials \( F_1, \ldots, F_n \) with multidegrees \( \upsilon_j \) for \( j = 1, \ldots, n \) is a zero-dimensional variety with

\[
\text{Bez}_{n_1, \ldots, n_r}(\upsilon_1; \ldots; \upsilon_n) := \sum_{\mathfrak{J} \in \mathcal{J}} \left( \prod_{i=1}^{n} \upsilon_{\mathfrak{J}_i} \right)
\]

points, where \( \mathcal{J} := \{ (j_1, \ldots, j_n) / \# \{ k : j_k = i \} = n_i \ \forall \ 1 \leq i \leq r \} \). For alternative proofs of this result using deformation techniques, we refer the reader to [36,37] and [26]. Note that this can also be seen as a particular case of Bernstein's theorem on the number of common roots of sparse systems [38].

The quantity \( \text{Bez}_{n_1, \ldots, n_r}(\upsilon_1; \ldots; \upsilon_n) \) is called the Bézout number of the generic multihomogeneous system.

If \( k_1, \ldots, k_r \) are positive integers with \( \sum_{j=1}^{r} k_j = n \), \( \text{Bez}_{n_1, \ldots, n_r}(\upsilon_1; \ldots; \upsilon_n) \) will denote the Bézout number of a multihomogeneous system with \( k_j \) polynomials of multidegree \( \upsilon_j \) for every \( 1 \leq j \leq t \).

The equations arising in the computation of totally mixed Nash equilibria of a game are multilinear (see the previous section): for a game with \( r \) players with \( n_1 + 1, \ldots, n_r + 1 \) pure strategies respectively, we have a system of \( n = \sum_{j=1}^{r} n_j \) polynomial equations consisting of exactly \( n_j \) polynomials of multidegrees equal to \( d_i := (1, \ldots, 1, 0, 1, \ldots, 1) \in \{0\}^n \) (where the \( 0 \) lies in the \( i \)th coordinate of \( d_i \)) for every \( 1 \leq i \leq r \). The multihomogeneous Bézout number associated to this system will be denoted by \( \delta \). In fact, for a "generic" game, this is the number of totally mixed Nash equilibria (see [27]).

Taking into account that \( d_0 = 0 \) for every \( 1 \leq i \leq r \), it is straightforward to see that \( \delta \) equals the cardinalinity of the set

\[
\mathfrak{J}_0 := \{ (j_1, \ldots, j_n) / j_k \neq i \ \forall \ 1 \leq k \leq r \text{ and } \# \{ j_k/k = i \} = n_i \ \forall \ 1 \leq i \leq r \}.
\]

We are going to deal with the case in which \( \delta > 0 \). This inequality can be determined by considering the set of exponents appearing with nonzero coefficients in each of the polynomials in the system (see [39, Chapter IV, Proposition 2.3]) and in our particular case, it is equivalent to the fact that \( n_j \leq \sum_{1 \leq j \leq r, \ k \neq j} n_k = n - n_i \) for every \( 1 \leq j \leq r \). From now on, we will assume that these inequalities hold.

2.4. Geometric resolutions

A way of representing zero-dimensional affine varieties which is widely used in computer algebra nowadays is a geometric resolution. This notion was first introduced in the works of Kronecker and König in the last years of the XIXth century ([40] and [41]) and appears in the literature under different names (rational univariate representation, shape lemma, etc.). For a detailed historical account on its application in the algorithmic framework, we refer the reader to [21]. An efficient polynomial algorithm using geometric resolutions and straight-line program encoding of polynomials can be found in [22] (see also [42] for a simplified approach to this solver). Roughly speaking, a geometric resolution consists in a rational parametrization of the variety in which the parameter values range over the set of roots of a univariate polynomial. Now, we give the precise definition we are going to use.

Let \( V = (\xi^{(1)}; \ldots; \xi^{(\ell)}) \subseteq \mathbb{A}^n \) be a zero-dimensional variety defined by polynomials in \( \mathbb{K}[x_1, \ldots, x_n] \). Given a separating linear form \( \ell = u_1x_1 + \cdots + u_nx_n \in \mathbb{K}[x_1, \ldots, x_n] \) for \( V \) (that is, a linear form \( \ell \) such that \( \ell(\xi^{(i)}) \neq \ell(\xi^{(j)}) \) if \( i \neq k \)), the following polynomials completely characterize the variety \( V \):

- the minimal polynomial \( p := \prod_{1 \leq j \leq \delta} (T - \ell(\xi^{(j)})) \in \mathbb{K}[T] \) of \( \ell \) over the variety \( V \) (where \( T \) is a new variable),
- polynomials \( w_1, \ldots, w_n \in \mathbb{K}[T] \) with \( \text{deg} w_j < \delta \) for every \( 1 \leq j \leq n \) satisfying

\[
V = \left\{ \frac{w_1}{p}(\eta), \ldots, \frac{w_n}{p}(\eta) \in \mathbb{K}^n / \eta \in \mathbb{K}, \ p(\eta) = 0 \right\}.
\]

The family of univariate polynomials \( p, w_1, \ldots, w_n \in \mathbb{K}[T] \) is called the geometric resolution of \( V \) (associated with the linear form \( \ell \)).

In our particular setting of totally mixed Nash equilibria computation, we will not only deal with zero-dimensional varieties in an affine space, but we will also consider zero-dimensional subvarieties of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r} \).
Write \( \xi := (\xi_1, \ldots, \xi_r) \) with \( \xi_i := (\xi_{i0} : \ldots : \xi_{ir}) \) \((1 \leq i \leq r)\) to denote a point in \( \mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_r \). Assume that \( V \subseteq \mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_r \) is a zero-dimensional variety defined by multihomogeneous polynomials in \( K[X_1, \ldots, X_r] \) such that \( \xi_{i0} \neq 0 (1 \leq i \leq r) \) holds for every point \( \xi \in V \). Then, we may associate with \( V \) the following zero-dimensional variety in \( \mathbb{A}^{n_r} \), where \( n := n_1 + \cdots + n_r \):

\[
\{(\xi'_1, \ldots, \xi'_r) \in \mathbb{A}^{n_r} / \xi'_i = (\xi_{i1}/\xi_{i0}, \ldots, \xi_{ir}/\xi_{i0}) \forall 1 \leq i \leq r, \xi \in V\}.
\]

A geometric resolution \( p, w_{11}, \ldots, w_{1n_1}, \ldots, w_{r1}, \ldots, w_{rn_r} \in K[T] \) of this zero-dimensional variety will also be called a geometric resolution of \( V \subseteq \mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_r \). In this case, the geometric resolution of \( V \) provides the following description of the variety:

\[
V = \{(p'(\eta) : w_{11}(\eta) : \ldots : w_{1n_1}(\eta)), \ldots, (p'(\eta) : w_{r1}(\eta) : \ldots : w_{rn_r}(\eta)))/ \eta \in K, p(\eta) = 0\}.
\]

3. On the totally mixed Nash equilibria of a generic game

This section is devoted to the study of totally mixed Nash equilibria of generic games. In order to do this, we will treat the payoff values as parameters and compute a geometric resolution of the set of quasi-equilibria of the associated generic game.

3.1. The set of quasi-equilibria of a generic game

Here we present an algorithm that computes a geometric resolution of the set of quasi-equilibria of a generic game with \( r \) players with \( n_1 + 1, \ldots, n_r + 1 \) pure strategies respectively, where \( n_i \in \mathbb{N} \) for every \( 1 \leq i \leq r \).

For \( 1 \leq i \leq r, 1 \leq k \leq n_i \), let \( A^{(k)} : (a^{(k)}_{j_1, \ldots, j_k}, \ldots, a^{(k)}_{j_1, \ldots, j_k}), \xi \in \mathbb{A}^{n_r} \) be a set of new indeterminates and

\[
F_k^{(i)} := \sum_{j=1}^{n_i} a^{(k)}_{j_1, \ldots, j_k} x_{j_1} \ldots x_{j_k} \ldots x_{j_{n_k}} x_{j_{n_k+1}} \ldots x_{j_{n_k}},
\]

where the sum runs over all \( j_1 \cdots j_{n_k} \), with \( 0 \leq j_k \leq n_k \) for every \( t \neq i \); that is, \( F_k^{(i)} \) is a generic multihomogeneous polynomial of multidegree \( d_i := (1, \ldots, 1, 0, \ldots, 1) \in (\mathbb{N})^r \) (where \( 0 \) is in the \( i \)-th coordinate). We also introduce a set of new indeterminates \( A^{(0)} : (A^{(0)}_{j_1, \ldots, j_k}), 1 \leq i \leq r, 1 \leq j \leq n_i \) which stand for the coefficients of a generic affine linear form \( A^{(0)} := \sum_{1 \leq i \leq r} A^{(0)}_{j_1, \ldots, j_k} x_j \) in the \( n = n_1 + \cdots + n_r \) variables \( x_j (1 \leq i \leq r, 1 \leq j \leq n_i) \), and we consider the multilinear polynomial

\[
F_0 := A^{(0)}_{j_1, \ldots, j_k} x_{j_1} \ldots x_{j_k} \ldots x_{j_{n_k}} x_{j_{n_k+1}} \ldots x_{j_{n_k}},
\]

which is obtained by homogenizing the generic affine linear form with respect to each group of variables \( X_{j_1}, \ldots, X_{j_k} \).

**Algorithm GenericGame**

**Input:** The number of players \( r \) and the number of pure strategies \( n_1 + 1, \ldots, n_r + 1 \) to each player.

**Output:** A (parametric) geometric resolution \( \{P, W_0; 1 \leq i \leq r, 1 \leq j \leq n_i\} \in \mathbb{C}[C^{(i)}_{j_1, \ldots, j_k}, 1 \leq i \leq r, 1 \leq j \leq n_i, 1 \leq k \leq n_i] \) of the set of quasi-equilibria of a generic game with the input structure.

**Procedure:**

1. Compute an slp encoding the resultant \( R = \text{Res}(F_0, F_1^{(1)}, \ldots, F_{r_1}^{(1)}, \ldots, F_{r_r}^{(r)}). \)

2. Compute the partial derivatives \( R_0 = \partial R / \partial A^{(0)}_{j_1, \ldots, j_k} \) and \( R_j = \partial R / \partial A^{(0)}_{j_1, \ldots, j_k} \) for \( 1 \leq i \leq r, 1 \leq j \leq n_i \).

3. Specialize \( A^{(0)} = T \), \( A^{(0)}_{j_1, \ldots, j_k} = -1 \) for \( 1 \leq i \leq r, A^{(0)}_{j_1, \ldots, j_k} = 0 \) for \( 1 \leq i \leq r, 2 \leq j \leq n_i \), and \( A^{(i)} = C^{(i)}_{j_1, \ldots, j_k} \) for \( 1 \leq i \leq r, 1 \leq k \leq n_i \) and each \( j \) in \( \delta \) in the polynomials \( R, R_0, R_j \) to obtain \( P, \partial R / \partial T \) and \( W_0 \) for \( 1 \leq i \leq r, 1 \leq j \leq n_i \).

**Theorem 1.** Algorithm GenericGame computes a geometric resolution of the set of quasi-equilibria of a generic game with \( r \) players having \( n_1 + 1, \ldots, n_r + 1 \) pure strategies respectively, with complexity \( O(D^2 (D + n_1 \ldots n_r) \delta \log(D)r^2 n^4 (n^3 + rN)) \), where

\[
D := \sum_{0 \leq i \leq r} n_i \text{Bez}_{n_1, \ldots, n_r}(d_0, n_0, d_1, n_1; \ldots; d_i, n_i - 1; \ldots; d_r, n_r) \quad (n_0 := 1, d_0 := (1, \ldots, 1)),
\]

\[
\delta := \text{Bez}_{n_1, \ldots, n_r}(d_1, n_1; \ldots; d_r, n_r),
\]

\[
n := \sum_{1 \leq i \leq r} n_i,
\]

\[
N := \prod_{1 \leq i \leq r} (n_i + 1) + \sum_{1 \leq i \leq r} n_i (n_i + 1)(n_i - 1)(n_i + 1) \ldots (n_r + 1).
\]
The algorithm obtains polynomials \( P(T), W_0(T) \in \mathbb{Q}[C_{j_1 \ldots j_r}] [T] \) giving the geometric resolution with \( \deg T \cdot P = \delta, \deg T \cdot W_0 < \delta \) and degrees bounded by \( D \) in the parameters \( C_{j_1 \ldots j_r} \), which are encoded by a straight-line program of length \( O(D^2(D + n_1 \ldots n_r \delta \log(D)r^2n^4(n^3 + rN))) \).

**Proof.** The first step of the algorithm consists in the computation of the polynomial \( R \) which is the specialization of the resultant of a system of multihomogeneous polynomials of respective multidegrees \( d_0 = (1, \ldots, 1), d_j^{(i)} = (0, 1, \ldots, 1) \) for \( 1 \leq j \leq n_1, \ldots, d_j^{(r)} = (1, \ldots, 1, 0) \) for \( 1 \leq j \leq n_r \), in which all the coefficients of the polynomial of multidegree \( d_0 \) corresponding to monomials not appearing in \( F_0 \) are substituted for 0 (see [43, Theorem 1]). This polynomial is obtained by applying an adapted version of the algorithm in [24, Theorem 5] (see Appendix A). The complexity of this algorithm is of order \( O(D^2(D + n_1 \ldots n_r \delta \log(D)r^2n^4(n^3 + rN))) \) and it computes an slp for \( R \) whose length is of the same order.

The algorithm to compute a geometric resolution of the zero-dimensional variety defined by the system \( F_k^{(i)} = 0 (1 \leq i \leq r, 1 \leq k \leq n_i) \) from \( R \) is standard: the parametrizations of the points in the variety are obtained from the partial derivatives \( R_0 = \partial R / \partial A_0^{(k)} \) and \( R_j = \partial R / \partial A_j^{(k)} \) for \( 1 \leq i \leq r, 1 \leq j \leq n_i \). An slp which encodes these derivatives can be computed from the slp representing \( R \) within the same complexity order and length of the same order as the slp which encodes \( R \) (see [33]). Note that \( \deg A_0^{(k)} R_j < \delta \) for every \( 1 \leq i \leq r, 1 \leq j \leq n_i \).

Let \( L := \sum_{1 \leq j \leq r} L_j x_j \) be a generic linear form in the variables \( x_j \), where \( L_j \) are new variables, and let \( T \) be another new variable. Let \( P_k^{(i)} \in \mathbb{Q}[A^{(k)}, L_j][T] \) be the polynomial obtained by specializing

\[
A_0^{(k)} \mapsto T, \quad A_j^{(k)} \mapsto -L_j x_0 (1 \leq i \leq r, 1 \leq j \leq n_i)
\]  

in \( R \). Since \( R \in (F_0, F_k^{(i)} : 1 \leq k \leq r, 1 \leq i \leq n_i) \), substituting \( L \) for \( T \) in \( P_k^{(i)} \), we obtain a polynomial \( P \in (F_k^{(i)} : 1 \leq k \leq r, 1 \leq i \leq n_i) \). As \( \deg_T (P_k) = \delta, P_k \) must be a multiple by a nonzero factor in \( \mathbb{Q}[A^{(k)}] \) of the minimal polynomial of \( L \). On the other hand,

\[
\frac{\partial P}{\partial L_j} = -\frac{\partial R}{\partial A_j^{(k)}} (L, -L_j, A^{(k)}_0 x_0) + \frac{\partial R}{\partial A_0^{(k)}} (L, -L_j, A^{(k)}_0 x_0)
\]

belongs to the ideal \( (F_k^{(i)} : 1 \leq k \leq r, 1 \leq i \leq n_i) \). We conclude that making the substitution (4) in \( R_0 \) and \( R_j (1 \leq i \leq r, 1 \leq j \leq n_i) \), polynomials which complete the geometric resolution of the variety defined by \( F_k^{(i)} = 0 \) with respect to the generic linear form \( L := \sum_{1 \leq j \leq r} L_j x_j \) can be obtained.

Now, we choose a separating linear form and we substitute its coefficients for the parameters \( L_j \). As the multihomogeneous system \( F_k^{(i)} = 0 \) is generic, it has no zeroes in the hyperplanes \( x_0 = 0 (1 \leq i \leq r) \) and we can consider its zeroes as affine points by setting \( x_0 = 1 (1 \leq i \leq r) \). The linear form \( \ell := \sum_{1 \leq i \leq r} x_i \) separates these affine points. To see this, choose coefficient vectors for the polynomials \( F_k^{(i)} \) so as to obtain a specific system \( f_k^{(i)} \) with the maximum number of affine solutions, and take a linear form \( \ell \in \mathbb{Q}[x_i; 1 \leq i \leq r, 1 \leq j \leq n_i] \) separating these solutions. Now, making a linear change of variables in each group \( X_i (1 \leq i \leq r) \), the linear form \( \ell \) maps to \( \ell \) and the specific system considered leads to a system of the same structure in the new variables having the maximum number of affine roots and that are separated by \( \ell \). As \( \ell \) is a separating linear form for a specific system, it is also separating for the generic one.

Hence, specializing

\[
A_0^{(k)} \mapsto T, \quad A_1^{(k)} \mapsto -1 (1 \leq i \leq r), \quad A_j^{(k)} \mapsto 0 (1 \leq i \leq r, 2 \leq j \leq n_i)
\]

in the polynomials \( R, R_0 \) and \( R_j (1 \leq i \leq r, 1 \leq j \leq n_i) \), new polynomials giving a geometric resolution of the set of common zeroes of the polynomials \( f_k^{(i)} \) in \( \mathbb{P}^{n_1} \times \ldots \times \mathbb{P}^{n_r} \) are obtained. Finally, substituting

\[
A_j^{(k)} = C_{j_1 \ldots j_r}^{(i)} - C_{j_1 \ldots j_r}^{(i)}
\]

for every \( 1 \leq i \leq r, 1 \leq k \leq n_i \) and each \( j \) with \( j \neq i \), the algorithm obtains polynomials \( P(T), \partial P / \partial T(T) \) and \( W_j(T) \in \mathbb{Q}[C_{j_1 \ldots j_r}] [T] \) such that the set of quasi-equilibria of the generic game in \( \mathbb{P}^{n_1} \times \ldots \times \mathbb{P}^{n_r} \) is represented as follows:

\[
\left\{ \left( \frac{\partial P}{\partial T}(t) : W_1(t) : \ldots : W_{n_1}(t) : \ldots : \frac{\partial P}{\partial T}(t) : W_{r_1}(t) : \ldots : W_{n_r}(t) \right) \right\} \quad t \in \mathbb{K}, \quad P(t) = 0
\]

where \( \mathbb{K} := \mathbb{Q}[C_{j_1 \ldots j_r}] \) (note that the linear form \( \ell \) still separates the quasi-equilibria of the generic game). The polynomials \( P, \partial P / \partial T \) and \( W_j \) are encoded by an slp of length \( O(D^2(D + n_1 \ldots n_r \delta \log(D)r^2n^4(n^3 + rN))) \) over \( \mathbb{Q} \), which is also the order of complexity of the whole computation. The upper bounds \( \deg_T P \leq \delta \) and \( \deg_T W_j < \delta \) follow from the stated upper bounds for the degrees of \( R \) and \( R_j \) in the variable \( A_0^{(k)} \).
3.2. Games with the maximum number of totally mixed Nash equilibria

The existence of games with the maximum possible number of totally mixed Nash equilibria, namely the multihomogeneous Bézout number \( \delta \) of the associated polynomial equation system, was proved in [27]. In this subsection, we will give an algorithm to obtain a finite family of polynomial conditions (inequalities over the reals) ensuring that a given game satisfies those conditions has \( \delta \) totally mixed Nash equilibria. To this end, we will use signed subresultant sequences as in [44]. We do not use the most sophisticated algorithmic version of this approach (see, for instance, [45]) because the polynomials we are working with depend on some parameters and are encoded by sp's. So, we will use the classical determinant-based construction of subresultants suitably adapted to our situation.

We will first recall some definitions and notation we will use (see, for example, [11, Sections 2.2.2 and 4.2.2]). For arbitrary polynomials \( P, Q \in \mathbb{R}[T] \) with \( P \neq 0 \), the Tarski query (also known as Sturm query) of \( Q \) for \( P \) is the number \( \text{TaQ}(Q, P) = \# \{ t \in \mathbb{R} : P(t) = 0,\ \text{Q}(t) > 0 \} - \# \{ t \in \mathbb{R} : P(t) = 0,\ \text{Q}(t) < 0 \} \) and the Cauchy index, \( I(Q/P) \) of the rational function \( Q/P \) is, informally, the number of “jumps” from \( -\infty \) to \( +\infty \) minus the number of “jumps” from \( +\infty \) to \( -\infty \) of the rational function \( Q/P \). For polynomials \( P \) and \( Q \) over any field, if \( p > q \) are the degrees of \( P \) and \( Q \) respectively, then for \( 0 \leq h \leq q \), the \( h \)th Sylvester–Habicht matrix of \( P \) and \( Q \), \( \text{SyHa}(P, Q) \), is the \( (p + q - 2h) \times (p + q - h) \)-matrix of the polynomials \( T^{q-h}P, \ldots, P, Q, \ldots, T^{p-h}Q \) in the basis \( \{ T^{p+q-h-1}, \ldots, T, 1 \} \) and the \( h \)th signed subresultant coefficient, \( s_{\text{Res}}(P, Q) \), is the determinant of the square matrix \( \text{SyHa}(P, Q) \) obtained from \( \text{SyHa}(P, Q) \) by deleting the last \( h \) columns. Besides, for \( P \) and \( Q \) in \( \mathbb{R}[T] \), \( \text{ResTP}(P, Q) \) is defined as the sign of the main coefficient in \( P \) to the \( (p - q) \)th power.

The main result of this section is the following:

**Theorem 2.** With our previous notation, there is a family of \( n\delta + 1 \) polynomials \( S_0, S_0^{(i)} \), \( 1 \leq i \leq r \), \( 1 \leq j \leq n_1 \), \( 1 \leq h \leq \delta \), in \( \mathbb{Q}[c_{j_1...j_r}]_{1 \leq i \leq r,\ 0 \leq j_i \leq n_i} \setminus \{ 0 \} \) with total degrees bounded by \( 45\delta \), such that for every vector \( c : = (c_{j_1...j_r})_{1 \leq i \leq r,\ 0 \leq j_i \leq n_i} \) with real coordinates satisfying the conditions

\[
S_0(c) \neq 0, \quad S_0^{(i)}(c) > 0 \quad (1 \leq i \leq r, \ 1 \leq j \leq n_1, \ 1 \leq h \leq \delta),
\]

the game with \( r \) players with \( n_1 + 1, \ldots, n_1 + 1 \) pure strategies and payoff values given by \( c \) has \( \delta \) totally mixed Nash equilibria.

The polynomials \( S_0 \) and \( S_0^{(i)} \) can be computed within complexity \( O(\delta^2(n\delta^2 + L)) \) from a straight-line program of length \( L \) encoding polynomials \( P, W \) in \( \text{Theorem 1} \). The algorithm obtains straight-line programs of length \( O(\delta^2(\delta^2 + L)) \) which encode these polynomials.

**Proof.** Consider a specific choice of payoff values \( c : = (c_{j_1...j_r})_{1 \leq i \leq r,\ 0 \leq j_i \leq n_i} \) over \( \mathbb{R} \) and assume that the polynomials \( P(c)(T) \) and \( W_j(c)(T) \) obtained from \( P(T) \) and \( W_j(T) \) by specializing the parameters at \( c \) provide a geometric resolution of the set of quasi-equilibria of the game with the given payoffs. Then, the totally mixed Nash equilibria of the game are those points \( (\xi_1, \ldots, \xi_r) \in \mathbb{R}^{n_1+1} \times \cdots \times \mathbb{R}^{n_r+1} \) of the form

\[
\xi_i = \left( P'(c)(t) W_{i1}(c)(t) S_i(c)(t), \ldots, W_{in}(c)(t) S_i(c)(t) \right) \quad (1 \leq i \leq r),
\]

where \( P' := \partial P/\partial T \) and \( S_i := P' + \sum_{1 \leq j \leq n_i} W_j \), having all their coordinates real and positive; that is, with \( t \) belonging to \( \{ t \in \mathbb{R} : P(c)(t) = 0, P'(c)(t) > 0, W_j(c)(t) > 0 \ \forall 1 \leq i \leq r, 1 \leq j \leq n_i \} \) or \( \{ t \in \mathbb{R} : P(c)(t) = 0, P'(c)(t) < 0, W_j(c)(t) < 0 \ \forall 1 \leq i \leq r, 1 \leq j \leq n_i \} \). Equivalently, \( t \) must belong to the intersection

\[
\bigcap_{1 \leq i \leq r,\ 1 \leq j \leq n_i} \{ t \in \mathbb{R} : P(c)(t) = 0, (P'(c)W_j(c)(t))(t) > 0 \}.
\]

For every \( 1 \leq i \leq r, 1 \leq k \leq n_i \), let \( c_k^{(i)} \in \mathbb{Q}[c_{j_1...j_r}][X_1, \ldots, X_r] \) be the polynomial obtained from \( c_k^{(i)} \) (see (3)) by means of the substitution stated in (5). Let us consider the resultant

\[
S_0 := \text{Res}_{\delta-1}(P(T), P'(T)) \in \mathbb{Q}[c_{j_1...j_r}]
\]

of \( P(T) \) and \( P'(T) \) regarded as polynomials in the variable \( T \) with coefficients in \( \mathbb{Q}[c_{j_1...j_r}] \), where the subindices indicate the degrees in \( T \) of \( P \) and \( P' \) respectively. Observe that, for every real vector \( c = (c_{j_1...j_r})_{1 \leq i \leq r,\ 0 \leq j_i \leq n_i} \) with \( S_0(c) \neq 0 \), \( P(c)(T) \) is a nonzero square-free polynomial of degree \( \delta \). Furthermore, the solution set of the system \( G_0(c) = 0, 1 \leq i \leq r, 1 \leq k \leq n_i \), is a zero-dimensional sub-variety of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r} \) with \( \delta \) distinct points and \( P(c)(T), W_j(c)(T) (1 \leq i \leq r, 1 \leq j \leq n_i) \) give a geometric resolution of this variety. Therefore, if the condition \( S_0(c) \neq 0 \) holds, the game with payoff vector \( c \) will have \( \delta \) different totally mixed Nash equilibria if and only if, for every \( 1 \leq i \leq r, 1 \leq j \leq n_i \),

\[
\# \{ t \in \mathbb{R} : P(c)(t) = 0, (P'(c)W_j(c)(t))(t) > 0 \} = \delta.
\]

Let us fix \( i \) and \( j, 1 \leq i \leq r, 1 \leq j \leq n_i \). Since the condition \( S_0(c) \neq 0 \) implies that the degree of \( P(c) \) is \( \delta \), we have that (7) is equivalent to \( \text{TaQ}(P'(c)W_j(c), P(c)) = \delta \). By [11, Proposition 2.57], this equality is the same as \( I((P'(c))^2W_j(c))/
where $P$, $W_0$ are regarded as polynomials in the variable $T$ with coefficients in $\mathbb{Q}[C_{j_1 \cdots j_r}]$ and $P$ is the leading coefficient of $P$. Then, for every payoff vector $c$ with $S_0(c) \neq 0$, we have that (7) is equivalent to $S_y^{(h)}(c) > 0$ for $1 \leq h \leq \delta$ (since $S_y^{(h)}(P(c), W_0(c))$ equals the coefficient of degree $\delta - 1$ of $W_0(c)$, the condition on the degree of $W_0(c)$ is ensured by $S_y^{(h)}(c)$ being positive).

Note that each of the polynomials $S_0$ and $sRes_0(P, W_0)$ for $0 \leq h \leq \delta - 1$ is the determinant of a matrix of size at most $2\delta - 1$ whose entries are polynomials in $\mathbb{Q}[C_{j_1 \cdots j_r}]$ of total degrees bounded by $D$ and so, their degrees are bounded by $2\delta D$.

Therefore, the degree of $S_y^{(h)}(1 \leq i \leq r, 1 \leq j \leq n_i, 1 \leq h \leq \delta)$ is bounded by $4\delta D$.

Finally, we detail the successive steps of the algorithm and we estimate its complexity and the length of an slp encoding the output polynomials. First, from an slp of length $L$ encoding $P$ and $W_0$ ($1 \leq i \leq r, 1 \leq j \leq n_i$), we obtain an slp of length $O(\delta^2L)$ within the same complexity for the coefficients of $P, P'$ and $W_0'$ ($1 \leq i \leq r, 1 \leq j \leq n_i$) in the variable $T$ (see [33, Lemma 21.25]). Then, we apply the division-free algorithm described in [46] in order to compute the determinants giving the characteristic polynomial of a matrix and it proceeds recursively, computing at each step the characteristic polynomial of a matrix obtained by deleting a row and a column from the matrix considered in the previous step. In our case, for every $1 \leq i \leq r$. $1 \leq j \leq n_i$, we have that for $h = 1, \ldots, \delta - 1$, the matrix $S_y^{(h)}(P, W_0)$ may be obtained from $S_y^{(h-1)}(P, W_0)$ by deleting the first and last row and the last two columns. Thus, all the signed subresultants $sRes_0(P, W_0)$ ($1 \leq h \leq \delta - 1$) are obtained as intermediate results in the computation of $sRes_0(P, W_0)$ by means of an adequate choice of rows and columns during the execution of the algorithm in [46]. Therefore, for each $1 \leq i \leq r$, $1 \leq j \leq n_i$, the algorithm obtains slps of length $O(\delta^2(\delta^2 + L))$ encoding the polynomials $S_y^{(h)}$ within complexity $O(\delta^6 \log(\delta^2 + L))$. The computation of an slp of length $O(\delta^2(\delta^2 + L))$ encoding $S_0$ is achieved within the same complexity order. The overall complexity of the algorithm is then $O(\delta^5(\delta^6 + L))$.

Note that, with the same notation as in the previous theorem, for a generic game with the given structure (namely, any game whose payoff vector $c$ satisfies $S_0(c) \neq 0$) the conditions $S_y^{(h)}(c) > 0$ are equivalent to the fact that the considered game has the maximum possible number of totally mixed Nash equilibria.

4. The set of totally mixed Nash equilibria of an arbitrary game

When dealing with a particular game with specific payoff values, evaluating the geometric resolution of the set of quasi-equilibria of a generic game with the same structure may fail to describe the quasi-equilibria of the given game. This section is aimed at adapting the procedures previously developed in order to handle this problem.

In fact, the main idea of this section is to describe by means of geometric resolutions finite sets which contain the isolated (in the complex sense) totally mixed Nash equilibria of any game. In the context of computational algebra, there are already algorithms computing the isolated solutions for any system of polynomial equations. The possible advantage of our approach is that, as we take into account the multihomogeneous structure of the systems involved, as in [26], the order of complexity of our algorithms may be lower than the known ones when the multihomogeneous bound for the number of equilibria is small.

4.1. Games with a zero-dimensional set of quasi-equilibria

As before, consider an $r$-person noncooperative game in normal form in which the players have $n_1 + 1, \ldots, n_r + 1$ distinct available pure strategies respectively. Let $c^{(i)} := (c^{(i)}_{j_1 \cdots j_r})_{0 \leq j_i \leq n_i}$ denote the payoff matrix to player $i$ for every $1 \leq i \leq r$. The polynomial equations defining the Nash equilibria of the game (see Section 2.2) can be obtained by specializing the coefficients of $F_{k}^{(i)}$ introduced in (3) in $a^{(k)} := (a_{j_1 \cdots j_r}^{(k)})_{0 \leq j_i \leq n_i}$ defined as follows $a_{j_1 \cdots j_r}^{(k)} := c^{(i)}_{j_1 \cdots j_r} - c^{(i)}_{j_1 \cdots j_r \theta_{j_i+1 \cdots j_r}}$. Thus, if $a := (a_{j_1 \cdots j_r})_{1 \leq i \leq r, 1 \leq j_i \leq n_i}$, the set of quasi-equilibria of the game is

$$V_a := \{ (\xi_1, \ldots, \xi_r) \in \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r} : F_{0}^{(k)}(a, \xi) = 0 \, \forall \, 1 \leq i \leq r, 1 \leq k \leq n_i \}.$$
and the resultant $R := \text{Res}(F_0, F_1^{(1)}, \ldots, F_{n_r}^{(r)})$. Let $R_a(A_{j_1, \ldots, j_r}^{(0)})$ be the polynomial obtained by substituting the coordinates of $a = (a^{(k)})_{1 \leq k \leq n_i}$ for $A^{(k)}$ in $R$. Then $V_a$ is either zero-dimensional or empty if and only if $R_a$ is not identically zero; if $V_a$ is empty, the result is straightforward. If $V_a$ is zero-dimensional, there exists a multilinear polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ which does not vanish at any of the (finitely many) points of $V_a$ and therefore, $R_a$ does not vanish at the coefficients of $f$. On the other hand, if $V_a$ has positive dimension, any multilinear polynomial $f$ has zeros in $V_a$ and, therefore, $R_a$ is identically zero.

The first step of the algorithm computes an slp of length $L := O(D^2(D + n_1 \ldots n_r \log(D)r^2n^4(n_3 + rN)))$ which encodes the multihomogeneous resultant $R$ by using the algorithm described in [24], adapted according to Appendix A below, within complexity of the same order as $L$. The specialization to obtain $R_a$ does not modify this complexity order.

Let $f_0 \in \mathbb{Q}[t][x_1, \ldots, x_n]$ be the polynomial obtained by specializing the variables $A_{j_1, \ldots, j_r}^{(0)}$ in $F_0$ into successive powers of a new variable $t$:

$$A_{j_1, \ldots, j_r}^{(0)} a^1 = t^{j_1+(n_1+1)j_2+(n_2+1)(n_2+1)j_3+\cdots+(n_1+1)j_r} \left( \prod_{j=0}^{r-1} (n_j+1) \right).$$

For every $\xi := (\xi_1, \ldots, \xi_r) \in \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$, we have

$$f_0(\xi)(t) = \sum_{j_1, \ldots, j_r} \xi_1^{(j_1)} \cdots \xi_r^{(j_r)} t^{j_1+(n_1+1)j_2+(n_2+1)(n_2+1)j_3+\cdots+(n_1+1)j_r} \in \mathbb{C}[t],$$

which is a nonzero polynomial due to the fact that there exists at least one choice of $j_1, \ldots, j_r$ for which the product $\xi_1^{(j_1)} \cdots \xi_r^{(j_r)}$ is not 0. Now, if $V_a$ is a finite set, let $\Delta(t) := \prod_{\xi \in V_a} f_0(\xi)(t)$. Note that there exists $t_0 \in \mathbb{Q}$ with $\Delta(t_0) \neq 0$. Then $f_0(t_0) \in \mathbb{Q}[x_1, \ldots, x_n]$ does not vanish at any point of $V_a$ and therefore the polynomial $R_{a, t} \in \mathbb{Q}[t]$ obtained by specializing $R_a$ following (8) does not vanish at $t_0$. Then $R_{a, t}$ is not the zero polynomial.

To decide whether $R_a \in \mathbb{Q}[A_{j_1, \ldots, j_r}^{(0)}]$ is zero or not, it suffices to decide whether $R_{a, t} \in \mathbb{Q}[t]$ is zero or not. Taking into account that $\deg(R_{a, t}) \leq (\prod_{1 \leq i \leq r} (n_i + 1) - 1)\delta$, this task can be achieved by evaluating $R_{a, t}$ at $\left( \prod_{1 \leq i \leq r} (n_i + 1) \right)$ different values of $t$, which is done by substituting the powers of these values for $A_{j_1, \ldots, j_r}^{(0)}$ as in (8) in the slp for $R_a$. The overall complexity of this procedure is $\left( \prod_{1 \leq i \leq r} (n_i + 1) \right) + L \left( \prod_{1 \leq i \leq r} (n_i + 1) \right).$ \hfill \square

Once we know that $V_a$ is zero-dimensional, even though the game does not have the maximum number of quasi-equilibria, we can obtain a geometric resolution of its set of affine quasi-equilibria

$V_a^{\text{aff}} := \{ \xi \in V_a : \xi_0 \neq 0 \ \forall \ 1 \leq i \leq r \}$

by means of a deterministic algorithm within a complexity polynomial in the same parameters as in the generic case.

We will use the following notation: given $F \in \mathbb{Q}[A_{j_1, \ldots, j_r}^{(0)}], \ A_{j_1, \ldots, j_r}^{(0)} F(\xi) \in \mathbb{Q}[t][A_{j_1, \ldots, j_r}^{(0)}]$ will denote the polynomial obtained from $F$ by specializing, as before, all the variables, except for $A_{j_1, \ldots, j_r}^{(0)}$, into successive powers of a variable $t$.

**Algorithm AffineQuasiEquilibria**

**Input:** A family of payoff matrices $c^{(i)} := (c^{(i)}_{j_1, \ldots, j_r})_{0 \leq k \leq n_k}$ for $1 \leq i \leq r$ defining a game with a finite number of quasi-equilibria.

**Output:** A geometric resolution of the set of affine quasi-equilibria of the game.

**Procedure:**

1. Compute the resultant $R$ of a family of multihomogeneous polynomials with multidegrees $d_0 = (1, \ldots, 1), d_j = (1, \ldots, 0, \ldots, 1)$ (where the 0 is in the $i$th coordinate) for $1 \leq i \leq r$ and $1 \leq j \leq n_i$.
2. Compute $A_{j_1, \ldots, j_r}^{(i)} := c^{(i)}_{j_1, \ldots, j_r} - c^{(i)}_{j_1, \ldots, j_r-1}$ and set $a$ for the vector with these entries.
3. Specialize $A^{(k)} := a^{(k)}$ in $R$ to obtain the polynomial $R_a(A_{j_1, \ldots, j_r}^{(0)}).$
4. Compute the leading coefficient of $R_a$ in the variable $A_{0, \ldots, 0}^{(0)}$ and divide $R_a$ by this coefficient, obtaining a polynomial $R_a^{\text{aff}}.$
5. Specialize $A_{j_1, \ldots, j_r}^{(0)} = A_{j_1, \ldots, j_r}^{(0)} A_{j_1, \ldots, j_r}^{(0)} = A_{j_1, \ldots, j_r}^{(0)}$ (where the index $j$ is in the $i$th place) for $1 \leq j \leq n_i$ and $A_{j_1, \ldots, j_r}^{(0)} = 0$ otherwise in $R_a^{\text{aff}},$ obtaining a polynomial $P_a.$
6. Compute the first nonzero subresultant of $P_a^{(0)}, \ \frac{\partial P_a^{(0)}}{\partial a^{(0)}}$, and an element $\tau \in \mathbb{Q}$ at which this subresultant does not vanish.
7. Compute $P_a = \frac{P_a^{(0)}}{\text{gcd}(\tau, \frac{\partial P_a^{(0)}}{\partial a^{(0)}})}$ and its partial derivatives.
8. Evaluate the variables $A_{j_1, \ldots, j_r}^{(0)}$ for $1 \leq i \leq r, 1 \leq j \leq n_i,$ at $(-\tau)^{0 \leq k \leq n_i-1}$ in the polynomial $P_a$ and its partial derivatives to obtain the polynomials giving the geometric resolution.
Theorem 4. Algorithm AffineQuasiEquilibria computes a geometric resolution of the set of affine quasi-equilibria of the game with \( r \) players having \( n_1 + 1, \ldots, n_r + 1 \) pure strategies respectively and payoff matrices \( c^{(1)}, \ldots, c^{(r)} \) within complexity \( O(\delta^D D^2 (D + n_1 \ldots n_r \delta \log(D)r^2n^2(n^3 + rnN))) \) provided that the associated set of quasi-equilibria is zero-dimensional.

Proof. Set \( \mathcal{F}_a := \prod_{\xi \in V_\mathcal{F}} F_0(\xi) \in \mathbb{Q}[A^{(0)}] \). For a given coefficient vector \( a^{(0)} \), \( \mathcal{F}_a(a^{(0)}) = 0 \) if and only if there is \( \xi \in V_a \) such that \( F_0(a^{(0)}, \xi) = 0 \); that is, if and only if \( F_0(a^{(0)}), F_1(a^{(1)}), \ldots, F_m(a^{(m)}) \) have a common root in \( \mathbb{P}^m \). But this is equivalent to the fact that \( \text{Res}(F_0, F_1, \ldots, F_m) \) vanishes at \( a^{(0)}, a^{(1)}, \ldots, a^{(m)} \) or, equivalently, that \( R_a(a^{(0)}) = 0 \). Then

\[
R_a = C_a \prod_{\xi \in V_\mathcal{F}} F_0(\xi)^{m_\xi}, \quad \text{with } C_a \in \mathbb{Q}, \ m_\xi \in \mathbb{N}.
\] (9)

To describe the set \( V^{\text{aff}}_a \) of affine quasi-equilibria of the game, we compute \( R^{\text{aff}}_a := \prod_{\xi \in V_\mathcal{F}} F_0(\xi)^{m_\xi} \). Note that \( R^{\text{aff}}_a \) is monic in the variable \( A^{(0)}_0 \) and \( \prod_{\xi \in V_\mathcal{F}} F_0(\xi)^{m_\xi} \) does not depend on this variable. Then, \( C_a \prod_{\xi \in V_\mathcal{F}} F_0(\xi)^{m_\xi} \) is the leading coefficient of \( R_a \) in the variable \( A^{(0)}_0 \).

Consider the generic form \( f_0 := A^{(0)}_{i_1} x_{i_1} \ldots x_{i_n} + \sum_{1 \leq i \leq n} A^{(0)}_i x_{i_1} \ldots x_{i_{i-1}} x_{i+1} x_{i+2} \ldots x_{i_0} \). After specializing \( R^{\text{aff}}_a \) as follows:

\[
A^{(0)}_{i_0} \mapsto A^{(0)}_0 \quad A^{(0)}_{i_0 \ldots i_n} \mapsto A^{(0)}_0 \quad \text{where the index } j \text{ is in the } i_{th} \text{ place} (1 \leq j \leq n) \]
\[
A^{(0)}_{i_0} \mapsto 0 \quad \text{otherwise}
\] (10)

we obtain \( \tilde{P}_a := \prod_{\xi \in V_\mathcal{F}} f_0(\xi)^{m_{\xi}} \). The geometric resolution of \( V^{\text{aff}}_a \) can be obtained from the square-free part \( P_a := \prod_{\xi \in V_\mathcal{F}} f_0(\xi) \) of \( P_a \) and the partial derivatives of \( P_a \) as shown in the proof of Theorem 1 and substituting afterwards the powers of a conveniently chosen scalar for the variables \( A^{(0)}_0 \).

Complexity and details of the different steps of the algorithm:

Computation of \( R^{\text{aff}}_a \): Let \( t_0 \in \mathbb{Q} \) be obtained as in the proof of Lemma 3 so that \( R_{a,t}(t_0) \neq 0 \), and let \( P^{(0)}_a(\xi) \in \mathbb{Q}[A^{(0)}_0, t] \) be the (nonzero) polynomial obtained from \( R_a \) after specializing it as in (8) for every \( (1, \ldots, t) \neq (0, \ldots, 0) \) setting \( t = t_0 \). Because of (9), \( \deg(R^{(0)}_a) = d_a \). Then, in order to compute \( d_a := \deg_{x_A^{(0)}_0} (R_a) \), it suffices to compute the coefficients of \( R^{(0)}_a \) up to degree \( \delta \) (an a priori upper bound for \( d_a \)). The complexity of this computation is of order \( O(\delta^2 \mathcal{L}) \). Now, after computing an sp of length \( O(\delta^2 \mathcal{L}) \) for the coefficient of \( (A^{(0)}_0)^{m_0} \), \( R^{\text{aff}}_a \) can be obtained by dividing \( R_a \) by this coefficient. As the divisor does not vanish when its variables are specialized in the successive powers of \( t_0 \), this division can be done by a classical division avoiding algorithm within complexity \( O(\delta^2 \mathcal{L}) \), which produces an sp of the same order [47].

Computation of \( P_a \): This polynomial is obtained by applying the well-known subresultant-based procedure for the computation of the gcd of two polynomials (see, for instance, [48]).

Let \( G = \gcd(\tilde{P}_a, \tilde{P}_a) \). For \( \xi_1 \neq \xi_2 \), \( f_0(\xi_1) \neq f_0(\xi_2) \) are relatively prime irreducible polynomials in \( \mathbb{Q}[A^{(0)}_0, t] \), therefore \( G(t) := \prod_{\xi \in V_\mathcal{F}} f_0(\xi)^{m_{\xi} - 1} = \gcd(f_0(\xi)^{m_{\xi}}, \frac{\frac{\partial f_0^{(0)}}{\partial A^{(0)}_0}}{\frac{\partial f_0^{(0)}}{\partial A^{(0)}_0}}) \). Then, \( d_a := \deg(\gcd(\tilde{P}_a, \frac{\partial \tilde{P}_a^{(0)}}{\partial A^{(0)}_0})) \) can be obtained as the degree of the gcd of the polynomials \( \tilde{P}_a^{(i)}(t) \) and \( \frac{\partial \tilde{P}_a^{(0)}}{\partial A^{(0)}_0} \). To compute this degree, the algorithm looks for its first nonzero subresultant. In each step, to decide whether the considered subresultant (which is a polynomial of degree at most \( 2\delta^2 n \) in \( \mathbb{Q}[t] \)) is zero or not, the algorithm evaluates the variable \( t \) in a sufficient number of elements of \( \mathbb{Q} \).

First, we obtain an sp of length \( O(\delta^4 \mathcal{L} + n) \) for \( \tilde{P}_a^{(i)}(t) \) and then an sp of length \( O(\delta^4 (\delta^4 \mathcal{L} + n)) \) for its coefficients in the variable \( A^{(0)}_0 \). For a specific evaluation of \( t \) the complexity of the computation of all the subresultants is \( O(\delta^4 \mathcal{L} + \delta^3 n^2) \), and therefore, the whole complexity of this step is bounded by \( O(\delta^4 \mathcal{L} n + \delta^4 n^2) \). The polynomial \( \tilde{G} = s_a \cdot \gcd(\tilde{P}_a, \frac{\partial \tilde{P}_a^{(0)}}{\partial A^{(0)}_0}) \) is obtained as the \( d_a \)th polynomial subresultant of \( \tilde{P}_a \) and \( \frac{\partial \tilde{P}_a^{(0)}}{\partial A^{(0)}_0} \) within complexity \( O(\delta^4 \mathcal{L}) \) (here \( s_a = s_{\text{Res}}(\tilde{P}_a, \frac{\partial \tilde{P}_a^{(0)}}{\partial A^{(0)}_0}) \)). Finally, \( P_a \) is obtained by dividing \( s_a \tilde{P}_a \) by \( \tilde{G} \). Note that we already know a point \( t \) where \( s_a \) does not vanish (the value obtained when computing \( d_a \)). Then, by evaluating the nonzero polynomial \( \tilde{G}(t)\tilde{A}^{(0)}_0, t) \in \mathbb{Q}[A^{(0)}_0, t] \) of degree \( d_a \) in at most \( d_a + 1 \) elements in \( \mathbb{Q} \), we obtain \( A^{(0)}_0 \in \mathbb{Q} \) such that \( \tilde{G}(t) A^{(0)}_0, t) \neq 0 \). This enables us to compute the quotient \( P_a \) by applying the algorithm in [47]. The complexity of this step is of order \( O(\delta^4 \mathcal{L}) \).

Computation of a geometric resolution: Observe that the linear form \( t \) whose coefficients are \( (-t)^k \) \( 0 \leq k \leq n-1 \) is a separating linear form for \( V^{\text{aff}}_a \). As in the proof of Theorem 1, the algorithm proceeds to compute the partial derivatives of \( P_a \) and evaluate \( P_a \) at its derivatives at the coefficients of \( t \). This may be done within \( O(\delta^4 \mathcal{L} n) \) operations.

The overall complexity of the algorithm is \( O(\delta^4 \mathcal{L} n + \delta^4 n^2) = O(\delta^6 D^2 (D + n_1 \ldots n_r \delta \log(D)r^2n^2(n^3 + rnN))) \). □
Following the previous notation, there is an algorithm which computes the number of totally mixed Nash equilibria of a game with zero-dimensional set of quasi-equilibria from the geometric resolution given by Theorem 4.

**Proposition 5.** Following the previous notation, there is an algorithm which computes the number of totally mixed Nash equilibria of a game with \( r \) players having \( n_1 + 1, \ldots, n_i + 1 \) pure strategies respectively within complexity \( O(\delta^D r^2 n_0^2 (n_1^3 + n_1 r)) \) provided that the associated set of quasi-equilibria of the game is zero-dimensional.

**Proof.** Using Theorem 4, let \( p \) and \( w_{ij}(1 \leq i \leq r, \ 1 \leq j \leq n_i) \) be the polynomials giving the geometric resolution of \( V^a \). Then, the totally mixed Nash equilibria of the game are the points \( (\xi_1, \ldots, \xi_r) \) where \( \xi_i = (p'(t)/s_i(t), w_{11}(t)/s_i(t), \ldots, w_{ir}(t)/s_i(t)) \), where \( s_i = p' + \sum_{j=1}^{n_i} w_{ij} \) for \( i = 1, \ldots, r \), and \( t \) is a root of \( p \), having all their coordinates real and positive. Therefore, the number of totally mixed Nash equilibria of the game equals the cardinality of the union of the sets \( \{ t \in \mathbb{R} : p(t) = 0, p'(t) > 0, w_{ij}(t) > 0 \ 1 \leq i \leq r, \ 1 \leq j \leq n_i \} \) and \( \{ t \in \mathbb{R} : p(t) = 0, p'(t) < 0, w_{ij}(t) < 0 \ 1 \leq i \leq r, \ 1 \leq j \leq n_i \} \).

Once we have \( p, p' \) and \( w_{ij}(1 \leq i \leq r, \ 1 \leq j \leq n_i) \) encoded in dense format (which can be obtained by fast interpolation techniques as explained in [51]), by using the algorithm in [50, Section 3.3], it is possible to compute this cardinality within complexity \( O(n^3) \).

**Remark 6.** The algorithm of Proposition 5 can be adapted to compute the list of Thom encodings of the real roots of the polynomial \( p \) (see [52] for a definition) in the geometric resolution of the set of affine quasi-equilibria of the game leading to totally mixed Nash equilibria within the same complexity.

4.2. Computing the isolated affine quasi-equilibria of an arbitrary game

When we are dealing with an arbitrary game, it may happen that the set \( V_a \) of its quasi-equilibria has positive dimension and, therefore, the polynomial \( R_0 \) introduced in the previous subsection (see the proof of Lemma 3) is identically zero. However, the following probabilistic algorithm computes the isolated affine quasi-equilibria of the game.

**Algorithm IsolatedAffineQuasiEquilibria**

*Input:* A family of payoff matrices \( c^{(i)} := (c_{ij}^{(i)})_{0 \leq j \leq n_i} \) for \( 1 \leq i \leq r \) of a game.

*Output:* A geometric resolution of a set including the isolated affine quasi-equilibria of the game.

*Procedure:*

1. Compute the resultant \( R \) of a family of multihomogeneous polynomials with multidegrees \( d_0 = (1, \ldots, 1), d_j^{(i)} = (1, \ldots, 0, \ldots, 1) \) (where the 0 is in the ith coordinate) for \( 1 \leq i \leq r \) and \( 1 \leq j \leq n_i \).
2. Choose a vector \( b := (b^{(ik)})_{1 \leq i \leq r, 1 \leq k \leq n_i} \) at random.
3. Compute \( e_{ik}^{(i)} := e_{ik}^{(i)} := c_{ik}^{(i)} + c_{ij}^{(i)} - c_{ij}^{(i)} \) and set \( a \) for the vector with these entries.
4. Specialize \( A_0 = a^{(0)} + u \cdot (b^{(i)} - d^{(0)}), \) where \( u \) is a new variable, \( A_0^{(0)} = A_0, A_0^{(0)} = A_0^{(0)}, \) where the index \( j \) is in the ith place, and \( A_0^{(0)} := 0 \) otherwise in \( R \), obtaining \( \tilde{P}_{a+b(u-a)} \).
5. Compute by interpolation the coefficients \( (p_r(u))_{0 \leq r \leq \ell} \in \mathbb{Q}[A_0^{(0)}] \) of \( \tilde{P}_{a+b(u-a)}(A_0^{(0)}) \).
6. For \( 0 \leq \ell \leq \delta \), compute the coefficients \( (p_{ik})_{0 \leq k \leq \ell} \) of \( p_r(u) \) and \( \epsilon_i = \min(1, p_{ik} \not= 0) \).
7. Compute \( \epsilon = \min(\epsilon_i : 0 \leq \epsilon \leq \delta) \) and \( P = \sum_{\ell=0}^{\delta}(1)^{\ell}p_{\ell}T^\ell \).
8. From the polynomial \( P \) obtain the required geometric resolution.

**Theorem 7.** Algorithm IsolatedAffineQuasiEquilibria is a probabilistic procedure which computes a geometric resolution of a finite set of points including the isolated affine quasi-equilibria of the game with \( r \) players having \( n_1 + 1, \ldots, n_i + 1 \) pure strategies respectively and payoff matrices \( c^{(1)}, \ldots, c^{(r)} \) within complexity \( O(D^2 \delta^3 r^2 n_0^2 (n_1^3 + n_1 r)) \).

**Proof.** The deformation procedure we use was applied in [26]. For the sake of completeness, we are going to explain it briefly. We keep our previous notation.

Consider a sufficiently generic coefficient vector \( b := (b^{(ik)})_{1 \leq i \leq r, 1 \leq k \leq n_i} \) such that \( R_b \not= 0 \) (\( b \) can be either chosen at random or effectively constructed as the coefficient vector of a system with \( \delta \) many common roots). Then, if \( u \) is a new variable, \( R_{a+b(u-a)} \) is a nonzero polynomial. If \( f_0^{(i)} \) is the polynomial obtained from \( f_0^{(i)} \) by evaluating \( x_0 = 1 (1 \leq j \leq r) \), let \( \tilde{f}_k^{(i)} := f_k^{(i)} (a^{(ik)} + u(b^{(ik)} - a^{(ik)})) (1 \leq i \leq r, 1 \leq k \leq n_i) \). Let \( \tilde{P}_{a+b(u-a)} \) be the polynomial obtained from \( R_{a+b(u-a)} \) by specializing it as in (10) and let \( L := \sum_{i} A_0^{(0)} x_i \). Let \( \epsilon \in N_0 \) such that \( \tilde{P}_{a+b(u-a)} = u^\epsilon \tilde{P} \) for \( \tilde{P} \in \mathbb{Q}[u, A_0^{(0)}] \) with \( P(u, A_0^{(0)}) \mid u = 0 \not= 0 \) and set \( \tilde{P} := \tilde{P} |_{u=0, A_0^{(0)} = -T} \).

Since \( R \) is a linear combination of \( R_0, f_0^{(i)} (1 \leq i \leq r, 1 \leq k \leq n_i), \) we have that \( \tilde{P}_{a+b(u-a)} |_{A_0^{(0)} = -L} = u^\epsilon \tilde{P} |_{A_0^{(0)} = -L} \in \tilde{J}_k^{(i)} : 1 \leq i \leq r, 1 \leq k \leq n_i) \in \mathbb{Q}[u, x_0 : 1 \leq i \leq r, 1 \leq j \leq n_i] \). Now, each irreducible component of the variety \( V_a \) defined by
the ideal \(\widetilde{I}_b^{(i)}: 1 \leq i \leq r, 1 \leq k \leq n_i\) in \(\mathbb{A}^{n+1}\) has dimension at least 1. Then, for each isolated point \(\xi\) of \(V^\text{aff}_a\), \((0, \xi) \in \mathbb{A}^{n+1}\) lies in an irreducible component \(C\) of \(V_a\) such that \(u \not\in I(C)\). Therefore \(\widetilde{P}_{a} \mid \mathbb{A}^{n+1}_{\text{aff}} \in I(C)\) and \(P \mid \mathbb{A}^{n+1}_{\text{aff}}\) vanishes at \(\xi\). Then \(P\) is a multiple of the minimal polynomial of \(L\) over the set of isolated points of \(V^\text{aff}_a\).

Note that, if \(\widetilde{P}_{a+u(b-a)} = \sum_{\ell=0}^{\delta} p_\ell(u)(A_\ell^{(0)})^\ell\), where \(p_\ell(u) \in \mathbb{Q}[A^{(0)}_y][u]\) are such that \(p_\ell(u) \neq 0\) for some \(0 \leq \ell \leq D\), then \(\epsilon := \max\{k : u^k \mid p_\ell \forall 0 \leq \ell \leq D\}\).

The procedure for the computation of \(P\) from an slp encoding \(R\) runs as follows: First, \(a + u(b-a)\) is computed within \(3N\) operations and then, the slp for \(R\) is specialized in \(a + u(b-a)\) and according to \((10)\). If \(L\) is the length of an slp encoding \(R\), an slp of length \(3N + L\) for \(\widetilde{P}_{a+u(b-a)}\) is obtained. Then an slp encoding the coefficients \(p_\ell(u) (0 \leq \ell \leq \delta)\) is obtained by interpolation, which takes \(O(\delta^2(N + L))\) operations. Afterwards, for every \(0 \leq \ell \leq \delta\), an slp which encodes the coefficients of \(p_\ell(u) = \sum_{k=1}^{\ell} p_k u^k\) is obtained, within complexity \(O(D^2\delta^2(N + L))\), in order to compute \(\epsilon_\ell = \min\{k : p_\ell \neq 0\}\). To decide whether each of the multivariate polynomials \(p_\ell \in \mathbb{Q}[A^{(y)}_y]\) encoded by an slp is zero or not, we apply the probabilistic Zippel–Schwartz zero test (see [53]). In this way, \(\epsilon = \min(\epsilon_\ell : 0 \leq \ell \leq \delta)\) is computed. The overall complexity of this step is \(O(D^2\delta^3(N + L))\). The last step of this procedure obtains an slp for \(P = \sum_{\ell=0}^{\delta} (-1)^\ell p_\ell T^\ell\) from the slp's encoding \(p_\ell (0 \leq \ell \leq \delta)\) within the same order of complexity.

Finally, from the polynomial \(P\), the algorithm obtains a geometric resolution for a finite set of points including the isolated points of \(V^\text{aff}_a\) in the same way we showed in the proof of Theorem 1, except that the linear form is now taken at random, obtaining a description for the isolated affine quasi-equilibria of the game.

Proceeding as in the previous subsection, now it is possible to obtain an upper bound on the number of isolated (in the complex space) totally mixed Nash equilibria of the game and the corresponding Thom encodings within the same order of complexity as in Theorem 7.

### 5. A final comment on complexities

The upper bound stated in Proposition 9 in Appendix B below shows that all the algorithms presented in this paper are polynomial in the number of strategies \(n_1, \ldots, n_r\) of the \(r\) players, and the generic number \(\delta\) of totally mixed Nash equilibria of a game with the considered structure.

We point out that the polynomial dependence of our complexity estimates in the parameter \(\delta\) is due to the use of the multihomogeneous structure of the polynomial systems involved. This is not the case for general algorithms, which only take into account degree bounds for the polynomials leading to complexity estimates in terms of the classical Bézout number, which in our situation equals \((r - 1)^{n_1 + \cdots + n_r}\). Although it is not easy to give a precise estimate of the multihomogeneous Bézout number in terms of \(r\) and \(n_1, \ldots, n_r\), the following table borrowed from [27] illustrates its order of magnitude when \(n_1 = \cdots = n_r\). In each box of the table, the first entry contains the multihomogeneous Bézout number and the second one the classical Bézout number.

| \(r\) | \(n_1\) |
|------|--------|
| 2    | 1      | 1      | 1      | 2       |
| 3    | 2      | 8      | 10     | 64      |
| 4    | 9      | 81     | 297    | 6561    |
| 5    | 44     | 1024   | 13756  | 1.0 \times 10^6 |
| 6    | 265    | 15625  | 925705 | 2.4 \times 10^8 |

### 6. Conclusions

In this paper, we showed a deterministic algorithm to compute a geometric resolution of the totally mixed Nash equilibria of a generic game in normal form with a fixed structure within a complexity which is cubic in the number of these equilibria. This complexity is due to the use of straight-line programs to encode multivariate polynomials and an efficient procedure to compute multihomogeneous resultants.

We also presented a deterministic algorithm to compute polynomial inequality conditions on the payoff values under which a game has the maximum possible number of this kind of equilibria.

Then, we designed symbolic deterministic procedures to describe the set of isolated totally mixed Nash equilibria of a game and to compute their exact number provided the set of quasi-equilibria of the game is finite.

The complexity of all our algorithms is polynomial in the number of players, the number of each player’s strategies and the number of totally mixed Nash equilibria of a generic game with the considered structure. For generic games, the theoretical cost of our algorithms is comparable to the one that could be obtained by applying the best probabilistic algorithms dealing with parametric polynomial systems (see [25]).
We expect that the techniques in [54], where the authors explain how to implement algorithms handling straight-line programs efficiently, could be adapted to future implementation of our algorithms. The application of these techniques might also decrease the exponents in the complexity of the algorithms proposed to deal with specific payoff values without introducing extra probabilistic aspects. This implementation issues and the design of a deterministic algorithm for arbitrary games is the matter of future research.

Acknowledgements

The authors were partially supported by the Argentinian research grants UBACyT X847 (2006–2009) and CONICET PIP 5852/05.

The first author was also partially supported by the Argentinian research grant ANPCyT PICT 2005 17-33018.

The authors wish to thank Joos Heintz for suggesting them the problem.

Appendix A. Computing multihomogeneous resultants

The procedure in [24] computes multihomogeneous resultants under the assumption that the coordinates of each multidegree are all positive. As this is not the case in our setting, this subsection is devoted to showing how to adapt that procedure.

In order to do this, we are going to use the theory in [55] and [43]. We can apply these results to our situation because the multihomogeneous resultant of a family of multihomogeneous polynomials $G_0, \ldots, G_n$ in $r$ groups of variables $X_j = (x_{j0}, \ldots, x_{jN})$, with $\sum_{i=1}^n n_i = n$, coincides with the sparse resultant of the dehomogenized polynomials $g_0, \ldots, g_n$ obtained by setting $x_{j0} = 1$ for every $1 \leq j \leq r$.

Let $A_0, \ldots, A_n \subset \mathbb{Z}^n$ be finite sets and let $g_0, \ldots, g_n$ be polynomials with supports $A_0, \ldots, A_n$, respectively. For any subset $J \subseteq \{0, \ldots, n\}$, let $L_J$ be the lattice generated by $\sum_{j \in J} A_j$. Following [55], if $I \subseteq \{0, \ldots, n\}$, the collection of supports $\{A_I\}_{I \subseteq \mathbb{J}}$ is said to be essential if rank($L_I$) = $\#I - 1$ and rank($L_J$) $\geq \#J$ for each proper subset $J$ of $I$. If there is a unique subcollection $\{A_I\}_{I \subseteq \mathbb{J}}$ which is essential, the resultant $\text{Res}(g_0, \ldots, g_n)$ is not constant and coincides with the resultant $\text{Res}(g_I; I \in \mathbb{J})$ (see [55, Corollary 1.1]).

Proposition 8. Let $n_1, \ldots, n_r$ be positive integers such that $n_i \leq \sum_{1 \leq k \leq r, k \neq i} n_k$ for every $1 \leq i \leq r$, and let $n := \sum_{1 \leq i \leq r} n_i$. Then, the resultant of $n + 1$ generic multihomogeneous polynomials $F_0, F_1, \ldots, F_{n_1}, \ldots, F_{n_r}$ in $r$ groups of $n_1 + 1, \ldots, n_r + 1$ variables, respectively, where $F_0$ has multidegree $d_0 = (1, \ldots, 1)$ and, for every $1 \leq i \leq r$, $F_i$ are polynomials of multidegree $d_i = (1, \ldots, 0, 1)$ (with $0$ in the $i$th coordinate), is a nonconstant polynomial and can be computed algorithmically within complexity $O(D^5(D + n_1 \ldots n_r \delta \log(D)r^2n^4(n^3 + rN)))$, where $D$, $\delta$ and $N$ are as in Theorem 1.

Proof. The resultant $R$ will be computed recursively by applying Poisson’s formula ([43, Lemma 13]). Once this formula is established, all the required computations run in the same way as in [24] and, therefore, the complexity of the algorithm is of the same order. At each step, we will have to compute a multihomogeneous resultant in one of the following settings, where $m_1, \ldots, m_r \in \mathbb{N}$ and $m := m_1 + \cdots + m_r$:

(i) $m - 1$ multihomogeneous polynomials in $r$ groups of $m_1 + 1, \ldots, m_r + 1$ variables: one polynomial with multidegree $(1, \ldots, 1)$ and, for every $1 \leq i \leq r$, $m_i$ polynomials with multidegrees $d_i := (1, \ldots, 0, 1)$ with $0$ in the coordinate $i$, under the assumption that $m_i \leq \sum_{j \neq i} m_j$ for every $1 \leq i \leq r$.

(ii) $m + 1$ multihomogeneous polynomials in $r$ groups of $m_1 + 1, \ldots, m_r + 1$ variables each, $m$ multihomogeneous polynomials in $r$ groups of variables with $m_1, m_2 + 1, \ldots, m_r + 1$ variables each, with $m_i$ polynomials of multidegree $d_i$ for every $1 \leq i \leq r$, under the assumption that $m_i \leq \sum_{j \neq i} m_j$ for every $1 \leq i \leq r$.

Now we are going to start with the recursion. The polynomials $F_0, F_1, \ldots, F_{n_1}, \ldots, F_{n_r}$ we start with satisfy the conditions in (I) with $m_i := n_i$ and $m := n$.

Let us solve case (I) in general: Let $G_0^{(i)}$ and $G_k^{(i)}$ ($1 \leq i \leq r$, $1 \leq k \leq m_i$) be a family of polynomials satisfying the conditions in (I) and let $\mathbb{J} := \{(0, 0)\} \cup \{(i, k) : 1 \leq i \leq r, 1 \leq k \leq m_i\}$. First, note that rank($L_I$) = $m = \#I - 1$. Let $J$ be a proper subset of $I$. If there exist $(i, k), (i', k')$ in $J$ with $i \neq i'$, then rank($L_J$) = $m \geq \#J$ and the same holds if $(0, 0) \in J$. On the other hand, if $J \subseteq \{(i, k) : 1 \leq k \leq m_i\}$ for a fixed $i \neq 0$, then rank($L_J$) = $\sum_{j \in J} m_j \geq m_i \geq \#J$. Therefore, the set of all supports is the unique essential subset. So, the resultant is not constant and the following identity holds:

$$\text{Res}(G_0^{(i)}, G_k^{(i)})_{1 \leq i \leq r, 1 \leq k \leq m_i}) = \prod_{\xi \in \mathbb{V}} g_0^{(i)}(\xi) \prod_{1 \leq j \leq r} \text{Res}(G_k^{(i)})_{1 \leq j \leq r, 1 \leq k \leq m_j},$$

where $g_0^{(i)}$ is the dehomogenized polynomial obtained from $G_0^{(i)}$ by evaluating $x_{ik} = 1$ ($1 \leq \ell \leq r$), $V$ is the set of common zeros in $A^m$ of the polynomials $g_k^{(i)}$ obtained in the same way from the $G_k^{(i)}$ ($1 \leq i \leq r, 1 \leq k \leq m_i$), and, for each $1 \leq j \leq r$, $G_k^{(i)}$ is the polynomial obtained from $G_k^{(i)}$ by setting $x_{ijk} = 0$. (Note that this result, applied to the polynomials $F_0, F_1, \ldots, F_{n_1}, \ldots, F_{n_r}$ implies that the resultant $R$ we want to compute is a nonconstant polynomial.)
(I.a) If $m_i \geq 2$ for every $1 \leq i \leq r$, (up to renaming variables and polynomials) each of the resultant $\text{Res}(G_{1}^{(i)})$ involves a family of polynomials satisfying the conditions in (III).

(I.b) Without loss of generality, assume now that $m_1 = 1$. Here, when computing $\text{Res}((G_{k}^{(i)})_{1 \leq i \leq r, 1 \leq k \leq m_i})$ we can discard the first group of variables. Then, the resultant involves $m$ polynomials in $r - 1$ groups of $m_2 + 1, \ldots, m_r + 1$ variables each with $m_i$ polynomials with multidegree $(1, \ldots, 1)$, where the 0 is in the $(i - 1)$th coordinate for $2 \leq i \leq r$, and one with multidegree $(1, \ldots, 1)$. If, for every $2 \leq i \leq r$, $m_i = \sum_{1 \leq j \leq r, j \neq i} m_j$, since $m_1 = 1$ we deduce that $m_i \leq \sum_{2 \leq j \leq r, j \neq i} m_j$ and so, the polynomial system obtained is of the form (I) but with one group of variables less than the original one. On the other hand, if $m_i = 1 + \sum_{2 \leq j \leq r, j \neq i} m_j$ for some $2 \leq i \leq r$, then $m_i < m_1$ for every $j \neq i$. Therefore, the unique essential subset is $(i, k) : 1 \leq k \leq m_i$ and the resultant to be computed is the resultant of the corresponding family of $m_i$ polynomials of multidegree $(1, \ldots, 1)$ in $r - 2$ groups of $m_2 + 1, \ldots, m_{i-1} + 1, m_{i+1} + 1, \ldots, m_r + 1$ variables each, which is the situation in (II).

If the conditions in (II) are met, all the coordinates of the multidegrees are not zero and so, we can apply the algorithm in [24] for the computation of the resultant.

To analyze (III), let us consider first the case when $r = 2$. Here, the assumption on the numbers $m_i$ implies that $m_1 = m_2 := M$.

(III.a) We consider the resultant of $M$ polynomials with multidegrees $(0, 1)$ and $M$ polynomials with multidegrees $(1, 0)$ in two groups of $M$ and $M + 1$ variables respectively. Now the unique essential set is the corresponding to the first $M$ polynomials and, therefore, as they are linear forms, the resultant equals the determinant of their coefficient matrix.

Assume now that $r > 2$. Note that the equality $m_1 = \sum_{j \neq i} m_j$ may be valid for at most one value $i$. If, on the contrary, $m_1 = \sum_{j \neq i} m_j$ hold for $i_1 \neq i_2$, it follows that $\sum_{j \neq i_1, i_2} m_j = 0$, which implies $r = 2$. Let $G_{k}^{(i)} (1 \leq i \leq r, 1 \leq k \leq m_i)$ be a family of polynomials satisfying the conditions in (III).

(Ill.b) If $m_1 = 1$, we are under the same assumptions as in (I.b).

(Ill.c) If $m_1 \geq 2$ and $m_1 = \sum_{j \neq i} m_j$ for some $2 \leq i \leq r$, the set $(i, k) : 1 \leq k \leq m_i$ is the unique essential subset. Then, the resultant involves $m_i$ polynomials of multidegrees $(1, \ldots, 1)$ in $r - 1$ groups of $m_1, m_2 + 1, \ldots, m_r + 1$ variables respectively and we are in situation (II).

(Ill.d) If $m_1 \geq 2$ and $m_i < \sum_{j \neq i} m_j$ for every $2 \leq i \leq r$ then $m_i < m_i - 1 + \sum_{j \neq i} m_j$ for every $2 \leq i \leq r$. Therefore, the unique essential subset is the whole family of supports and applying Poisson’s formula we obtain:

$\text{Res}((G_{k}^{(i)})_{1 \leq i \leq r, 1 \leq k \leq m_i}) = \prod_{\xi \in W} g_{1}^{(1)} (\xi) \prod_{2 \leq i \leq r} \text{Res}((G_{k}^{(i)})_{2 \leq i \leq r, 1 \leq k \leq m_i}),$

where $g_{1}^{(1)}$ is the dehomogenized of $G_{1}^{(i)}$ by setting $x_{m_j} = 1$ for every $1 \leq j \leq r$, $W$ is the set of common zeros in $A^{m-1}$ of the polynomials $g_{k}^{(i)} (2 \leq k \leq m_i)$ obtained in the same way from the $G_{k}^{(i)}$, and $G_{1}^{(i)}$ is the polynomial obtained from $G_{1}^{(i)}$ by setting $x_{m_i} = 0$.

For $l = 2, \ldots, r$, setting $m' := m - 1$, $m'_1 := m_1 - 1$ and $m'_i := m_i$ for $i \neq 1$, the resultant to be computed involves $m'$ polynomials in $r$ groups of $m'_1 + 1, \ldots, m_{i-1} + 1, m'_i, m'_{i+1} + 1, \ldots, m'_r + 1$ variables each with $m'_i$ polynomials of multidegree $d_i$ for every $1 \leq i \leq r$. We have $m'_i \leq \sum_{j \neq i} m'_j$ and, for $i \neq 1$, the condition $m'_i < \sum_{j \neq i} m'_j$ implies that $m' \leq \sum_{j \neq i} m'_j$; therefore, renaming variables and polynomials, we are again under the assumptions of (III).

Appendix B. A bound for the degree of the resultant

As before, we assume $n = n_1 + \cdots + n_r$ with $n_i \in \mathbb{N}$ for every $1 \leq i \leq r$. We consider the resultant $R$ of a family of $n + 1$ multilinear polynomials in $r$ groups of $n_1 + 1, \ldots, n_r + 1$ variables each, consisting of a polynomial $F_0$ of multidegree $d_0 := (1, \ldots, 1)$ and, for every $1 \leq i \leq r$, a set of $n_i$ polynomials $F_k^{(i)} (1 \leq k \leq n_i)$ of multidegree $d_i := (1, \ldots, 1, 0, 1, \ldots, 1)$, where the 0 lies in the $i$th coordinate. Denoting

$\delta_i := \text{Bez}_{n_1, \ldots, n_r} (d_0, 1; d_1, n_1; \ldots; d_i, n_i - 1; \ldots; d_r, n_r), \quad i = 1, \ldots, r,$

the resultant $R$ is a multihomogeneous polynomial in the coefficients of $F_0$, $F_k^{(i)}$ of degree $\delta$ in the coefficients of $F_0$ and $\delta_i$ in the coefficients of $F_k^{(i)}$ for every $1 \leq i \leq r, 1 \leq k \leq n_i$ (see, for instance, [56]). Therefore, the total degree of $R$ equals

$D = \delta + \sum_{1 \leq i \leq r} n_i \delta_i.$

Proposition 9. Following the previous notations, $D \leq \left(1 + \sum_{1 \leq i \leq r} n_i (n_i + 1)\right) \delta \leq n^2 \delta.$
Proof. Let us prove that \( \delta \leq (n_1 + 1) \delta \) for every \( 1 \leq i \leq r \). Without loss of generality, suppose \( i = 1 \). As the Bézout number is additive in each of the multidegrees involved (see identity (1)) and \( d_0 = d_1 + e_1 \), where \( e_1 = (1, 0, \ldots, 0) \), we have
\[
\delta_1 = \text{Bez}_{n_1, \ldots, n_r}(d_1, n_1; d_2, n_2; \ldots; d_r, n_r) + \text{Bez}_{n_1, \ldots, n_r}(e_1, 1; d_1, n_1 - 1; d_2, n_2; \ldots; d_r, n_r) = \delta + \text{Bez}_{n_1, \ldots, n_r}(e_1, 1; d_1, n_1 - 1; d_2, n_2; \ldots; d_r, n_r).
\]

Now, identity (1) implies that \( \text{Bez}_{n_1, \ldots, n_r}(e_1, 1; d_1, n_1 - 1; d_2, n_2; \ldots; d_r, n_r) = \#J_1 \), where \( J_1 = \{ (j_1, \ldots, j_m) / j_{11} = 1, j_{ik} \neq i \forall (i, k) \neq (1, 1) \} \) and \( \#J_{ab} / J_{ab} = i \) = \( n_1 \) \( 1 \leq i \leq r \).

In order to finish the proof, we will show that \( \#J_1 \leq n_1 \delta \). Since \( \delta \) equals the cardinality of the set \( \mathcal{Z}_0 \) introduced in (2), we will compare the cardinalities of both sets \( J_1 \) and \( \mathcal{Z}_0 \). To this end, we define the following map from \( J_1 \) to \( \mathcal{Z}_0 \) with a given \( n \)-tuple \( j = (1, j_2, \ldots, j_{n_1}, 1, j_{n_2}, \ldots, j_{n_1 n_2}, \ldots, j_{n_1 n_2 \ldots n_r}) \in J_1 \) we associate the \( n \)-tuple \( j' \in \mathcal{Z}_0 \) which is obtained by exchanging the first coordinate of \( j \) (which equals 1) with the first one which is different from 1 and is located beyond the \( n_1 \)th coordinate. Note that a necessary condition for two distinct \( n \)-tuples in \( J_1 \) to lead to the same \( n \)-tuple in \( \mathcal{Z}_0 \) by means of this assignment is that they coincide in all of their coordinates except for two of them located among the \( n_1 \) coordinates, \( n_1 + 1, \ldots, 2n_1 \). Moreover, the vector consisting of these \( n_1 \) coordinates must be of the form \((1, \ldots, 1, j_{ab}, \ldots)\) for both of them (possibly with no 1 at the beginning) and so, they can only differ in the length of the string of 1s in this vector, which ranges between 0 and \( n_1 - 1 \). We conclude that each element of \( \mathcal{Z}_0 \) is the image of at most \( n_1 \) elements of \( J_1 \). It follows that \( \#J_1 \leq n_1 \#\mathcal{Z}_0 \) as we wanted to prove. \( \square \)

References

[1] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behaviour, Princeton University Press, Princeton, NJ, 1944.
[2] J.F. Nash, Non-cooperative games, Ann. of Math. (2) 54 (1951) 286–295.
[3] D. Scott, Systems of polynomial equations, in: CBMS Regional Conference Series in Mathematics, 97. Published for the Conference Board of the Mathematical Sciences, Washington, DC, American Mathematical Society, Providence, RI, 2002.
[4] C.E. Lemke, J.T. Howson Jr., Equilibrium points of bimatrix games, J. Soc. Indust. Appl. Math. 12 (1964) 413–423.
[5] B. von Stengel, Computing equilibria for two person games, in: R.J. Aumann, S. Hars (Eds.), Handbook of Game Theory with Economic Applications, Handb. of Comput. Econom. 1996 413–423.
[6] H. Scarf, The approximation of fixed points of a continuous mapping, SIAM J. Appl. Math. 15 (1967) 1328–1343.
[7] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, Princeton, NJ, 1944.
[8] R. McKelvey, A. McLennan, Gambit: Software Tools for Game Theory, Version 0.2007.01.30, http://gambit.sourceforge.net.
[9] M.J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, Cambridge MA, 1994.
[10] M. Aumann, M. Rubinstein, A Course in Game Theory, MIT Press, Cambridge, MA, 1994.
[11] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behaviour, Princeton University Press, Princeton, NJ, 1944.
[12] B. von Stengel, Computing equilibria for two person games, in: R.J. Aumann, S. Hars (Eds.), Handbook of Game Theory with Economic Applications, Handb. of Comput. Econom. 1996 413–423.
[13] H. Scarf, The approximation of fixed points of a continuous mapping, SIAM J. Appl. Math. 15 (1967) 1328–1343.
[14] M. Aumann, M. Rubinstein, A Course in Game Theory, MIT Press, Cambridge MA, 1994.
[15] M. Aumann, M. Rubinstein, A Course in Game Theory, MIT Press, Cambridge, MA, 1994.
[16] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behaviour, Princeton University Press, Princeton, NJ, 1944.
[40] L. Kronecker, Grundzüge einer arithmetischen Theorie de algebraischen Grössen, J. Reine Angew. Math. 92 (1882) 1–122.
[41] J. König, Einleitung in die allgemeine Theorie der algebraischen Grössen, Druck und Verlag von B.G. Teubner, Leipzig, 1903.
[42] C. Durvy, G. Lecerf, A concise proof of the Kronecker polynomial system solver from scratch, Expo. Math. 26 (2) (2008) 101–139.
[43] M. Minimair, Sparse resultant under vanishing coefficients, J. Algebraic Combin. 18 (1) (2003) 53–73.
[44] L. González Vega, H. Lombardi, T. Recio, M.-F. Roy, Spécialisation de la suite de Sturm. RAIRO, Inform. Théor. Appl. 28 (1) (1994) 1–24.
[45] T. Lickteig, M.-F. Roy, Sylvester–Habicht sequences and fast Cauchy index computation, J. Symbolic Comput. 31 (3) (2001) 315–341.
[46] S.J. Berkowitz, On computing the determinant in small parallel time using a small number of processors, Inform. Process. Lett. 18 (3) (1984) 147–150.
[47] V. Strassen, Vermeidung von divisionen, J. Reine Angew. Math. 264 (1973) 184–202.
[48] W. Brown, J. Traub, On Euclid’s algorithm and the theory of subresultants, J. Assoc. Comput. Mach. 18 (1971) 505–514.
[49] M. Ben-Or, D. Kozen, J. Reif, The complexity of elementary algebra and geometry, J. Comput. System Sci. 32 (2) (1986) 251–264.
[50] J. Canny, Improved algorithms for sign determination and existential quantifier elimination, Comput. J. 36 (5) (1993) 409–418.
[51] J. von zur Gathen, J. Gerhard, Modern Computer Algebra, Cambridge University Press, New York, 1999.
[52] M. Coste, M.-F. Roy, Thom’s lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets, J. Symbolic Comput. 5 (1–2) (1988) 121–129.
[53] R. Zippel, Probabilistic algorithms for sparse polynomials, in: Symbolic and algebraic computation (EUROSAM ’79, Internat. Sympos., Marseille, 1979), in: Lecture Notes in Comput. Sci., vol. 72, 1979, pp. 216–226.
[54] M. Giusti, K. Hägele, G. Lecerf, J. Marchand, B. Salvy, The projective Noether Maple package: Computing the dimension of a projective variety, J. Symbolic Comput. 30 (3) (2000) 291–307.
[55] B. Sturmfels, On the Newton polytope of the resultant, J. Algebraic Combin. 3 (2) (1994) 207–236.
[56] P. Pedersen, B. Sturmfels, Product formulas for resultants and Chow forms, Math. Z. 214 (1993) 377–396.