A parametric representation of totally mixed Nash equilibria

Gabriela Jeronimo[a,c,*†] Daniel Perrucci[a,*] Juan Sabia[a,b,c,*]

[a] Departamento de Matemática, FCEN,
Universidad de Buenos Aires, Ciudad Universitaria, 1428 Buenos Aires, Argentina

[a] Departamento de Ciencias Exactas, CBC,
Universidad de Buenos Aires, Ciudad Universitaria, 1428 Buenos Aires, Argentina

[c] CONICET - Argentina

Abstract

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.

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.

*Partially supported by the following Argentinian research grants: UBACyT X847 (2006-2009) and CONICET PIP 5852/05.

†Partially supported by the Argentinian research grant ANPCyT PICT 2005 17-33018.

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 polynomials 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 to 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], [15], [16], [17], [18], [19], [20], [21], [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 Subsection 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 $n_1+1,\dots,n_r+1$ pure strategies respectively, within complexity $O(\delta^3 n^8(\log(n)+\log(\delta))r^2 n_1\dots n_r(n^3+r^2\prod_{1\le i\le r}(n_i+1)))$, where δ is the number of totally mixed Nash equilibria of a generic game with the given structure and $n=\sum_{1\le i\le r} n_i$.*

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 I, 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 I*, there is a family of $n\delta+1$ polynomials with rational coefficients S_0, $S_{ij}^{(h)}$, $1\le i\le r$, $1\le j\le n_i$ and $1\le h\le\delta$, in the payoff values of a game with r players with $n_1+1,\dots,n_r+1$ pure strategies such that for every payoff vector c satisfying the conditions*

$$S_0(c)\neq 0,\ S_{ij}^{(h)}(c)>0 \quad (1\le i\le r, 1\le j\le n_i, 1\le h\le\delta),$$

the associated game has δ totally mixed Nash equilibria. The polynomials S_0 and $S_{ij}^{(h)}$ have degrees bounded by $4\delta^2n^2$ 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 $\deg_{x_i} f$ 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 $\mathbb{A}^n(k)$ and $\mathbb{P}^n(k)$ (or simply by $\mathbb{A}^n$ or $\mathbb{P}^n$ 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] and [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 non-cooperative 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 $n_1 + 1, \dots, n_r + 1$ different available pure strategies respectively ($n_1, \dots, n_r \in \mathbb{N}$).

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

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

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 $X_1, \dots, X_r$ satisfying $\pi_i(X_1, \dots, X_{i-1}, X_i, X_{i+1}, \dots, X_r)$ $\pi_i(X_1, \dots, X_{i-1}, X_i', X_{i+1}, \dots, X_r)$ for every $1 \le i \le r$ 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 $1 \le i \le r$, $0 \le j \le n_i$.

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, Sec. 6.3]). They are the real vectors $(X_1, \dots, X_r)$ with $X_i := (x_{i0}, \dots, x_{in_i})$ for every $1 \le i \le r$ satisfying:

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

(b) $\displaystyle\sum_{0 \le j \le n_i} x_{ij} = 1$ for $i = 1, \dots, r$,

(c) $\displaystyle\sum_{J_{-i}} \left(c^{(i)}_{j_1 \dots j_{i-1} k j_{i+1} \dots j_r} - c^{(i)}_{j_1 \dots j_{i-1} 0 j_{i+1} \dots j_r} \right) x_{1j_1} \dots x_{i-1 j_{i-1}} x_{i+1 j_{i+1}} \dots x_{rj_r} = 0$ for $i = 1, \dots, r$, $k = 1, \dots, n_i$, where the sum runs over all $J_{-i} := j_1 \dots j_{i-1} j_{i+1} \dots j_r$ with $0 \le j_t \le n_t$ for every $t \ne i$.

Observe that (c) is a system of $n := n_1 + \cdots + n_r$ multihomogeneous polynomial equations in the r groups of variables $X_1, \dots, X_r$ with $n_1 + 1, \dots, n_r + 1$ 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 $\{x_{i0} = 0\}$ $(1 \le i \le r)$ will be called *affine quasi-equilibria* of the game.

Every quasi-equilibrium $\xi := (\xi_1, \dots, \xi_r)$ determines at most one totally mixed Nash equilibrium of the game: for every $1 \le i \le r$, let $s_{\xi_i} := \sum_{0 \le j \le n_i} \xi_{ij}$ be the sum of the coordinates of ξ_i. If $s_{\xi_i} \neq 0$ for every $1 \le i \le r$, the unique associated representation of ξ whose coordinates satisfy condition (b) is $(\xi_1/s_{\xi_1}, \dots, \xi_r/s_{\xi_r})$, 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 $n_1, \dots, n_r$ and consider r groups of variables $X_j := (x_{j0}, \dots, x_{j\,n_j})$, $j = 1, \dots, r$. We say that $F \in K[X_1, \dots, X_r]$ is *multihomogeneous* of *multidegree* $v := (v_1, \dots, v_r)$, where $(v_1, \dots, v_r) \in \mathbb{N}_0^r$, if F is homogeneous of degree v_j in the group of variables X_j for every $1 \le j \le r$.

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 $F_1, \dots, F_n$ with multidegrees $\nu_i := (\nu_{i1}, \dots, \nu_{ir})$ for $i = 1, \dots, n$ is a zero-dimensional variety with

$$\mathrm{Bez}_{n_1\dots,n_r}(\nu_1; \dots; \nu_n) := \sum_{(j_1,\dots,j_n)\in\mathfrak{J}} \Big(\prod_{i=1}^{n} \nu_{ij_i}\Big) \tag{1}$$

points, where $\mathfrak{J} := \{(j_1, \dots, j_n) \ / \ \#\{k : j_k = i\} = n_i \ \forall\, 1 \le i \le r\}$. 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 $\mathrm{Bez}_{n_1\dots,n_r}(\nu_1; \dots; \nu_n)$ is called the *Bézout number* of the generic multihomogeneous polynomial system. If $k_1, \dots, k_t$ are positive integers with $\sum_{i=1}^t k_i = n$, $\mathrm{Bez}_{n_1,\dots,n_r}(\nu_1, k_1; \dots; \nu_t, k_t)$ will denote the Bézout number of a multihomogeneous system with k_i polynomials of multidegree ν_i for every $1 \le i \le t$.

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 $n_1+1, \dots, n_r+1$ pure strategies respectively, we have a system of $n = \sum_{j=1}^r n_j$ polynomial equations consisting of exactly n_i polynomials of multidegrees equal to $d_i := (1, \dots, 1, 0, 1, \dots, 1) \in (\mathbb{N}_0)^r$ (where the 0 lies in the ith coordinate of d_i) for every $1 \le i \le r$. The multihomogeneous Bézout number associated to this system will be denoted by δ. In fact, for a "generic" game, this is the number of totally mixed Nash equilibria (see [27]).

Taking into account that $d_{ii} = 0$ for every $1 \le i \le r$, it is straightforward to see that δ equals the cardinality of the set

$$\mathfrak{J}_0 = \{(j_{11}, \dots, j_{rn_r}) \,/\, j_{ik} \neq i \,\forall\, 1 \le k \le n_i \text{ and } \#\{j_{hk} \,/\, j_{hk} = i\} = n_i \ \forall\, 1 \le i \le r\}. \tag{2}$$

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 $n_j \le \sum_{1\le k \le r,\ k\neq j} n_k = n - n_j$ for every $1 \le j \le r$. 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 $V = \{\xi^{(1)}, \dots, \xi^{(\delta)}\} \subset \mathbb{A}^n$ be a zero-dimensional variety defined by polynomials in $K[x_1, \dots, x_n]$. Given a *separating* linear form $\ell = u_1x_1 + \cdots + u_nx_n \in K[x_1, \dots, x_n]$ for V (that is, a linear form ℓ such that $\ell(\xi^{(i)}) \neq \ell(\xi^{(k)})$ if $i \neq k$), the following polynomials completely characterize the variety V:

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

$$V = \big\{ \big(\frac{w_1}{p'}(\eta), \dots, \frac{w_n}{p'}(\eta)\big) \in \overline{K}^n \,/\, \eta \in \overline{K},\ p(\eta) = 0 \big\}.$$

The family of univariate polynomials $p, w_1, \dots, w_n \in K[T]$ is called the *geometric resolution* of V (associated with the linear form ℓ).

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 $\xi := (\xi_1, \dots, \xi_r)$ with $\xi_i := (\xi_{i0} : \cdots : \xi_{in_i})$ $(1 \le i \le r)$ to denote a point in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$. Assume that $V \subset \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$ is a zero-dimensional variety defined by multihomogeneous polynomials in $K[X_1, \dots, X_r]$ such that $\xi_{i0} \neq 0$ $(1 \le i \le r)$ holds for every point $\xi \in V$. Then, we may associate with V the following zero-dimensional variety in $\mathbb{A}^n$, where $n := n_1 + \cdots + n_r$:

$$\{(\xi'_1, \dots, \xi'_r) \in \mathbb{A}^n \,/\, \xi'_i = (\xi_{i1}/\xi_{i0}, \dots, \xi_{in_i}/\xi_{i0})\ \forall\, 1 \le i \le r,\ \xi \in V\}.$$

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

$$V = \big\{ \big((p'(\eta) : w_{11}(\eta) : \cdots : w_{1n_1}(\eta)), \dots, (p'(\eta) : w_{r1}(\eta) : \cdots : w_{rn_r}(\eta))\big) / \, \eta \in \overline{K},\ p(\eta) = 0 \big\}.$$

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 $n_1+1,\dots,n_r+1$ pure strategies respectively, where $n_i \in \mathbb{N}$ for every $1 \le i \le r$.

For $1 \le i \le r$, $1 \le k \le n_i$, let $A^{(ik)} := (A^{(ik)}_{j_1\dots j_{i-1}j_{i+1}\dots j_r})_{0\le j_t\le n_t}$ be a set of new indeterminates and

$$F^{(i)}_k := \sum_{J_{-i}} A^{(ik)}_{J_{-i}}\, x_{1j_1}\dots x_{i-1j_{i-1}}x_{i+1j_{i+1}}\dots x_{rj_r}, \tag{3}$$

where the sum runs over all $J_{-i} := j_1\dots j_{i-1}j_{i+1}\dots j_r$ with $0 \le j_t \le n_t$ for every $t \ne i$; that is, $F^{(i)}_k$ is a generic multihomogeneous polynomial of multidegree $d_i := (1,\dots,1,0,1,\dots,1) \in (\mathbb{N}_0)^r$ (where 0 is in the i-th coordinate). We also introduce a set of new indeterminates $A^{(0)} := (A^{(0)}_0, A^{(0)}_{ij} : 1 \le i \le r,\ 1 \le j \le n_i)$ which stand for the coefficients of a generic affine linear form $A^{(0)}_0 + \sum_{\substack{1\le i\le r\\1\le j\le n_i}} A^{(0)}_{ij}\, x_{ij}$ in the $n = n_1+\dots+n_r$ variables x_{ij} $(1 \le i \le r, 1 \le j \le n_i)$, and we consider the multilinear polynomial

$$F_0 := A^{(0)}_0 x_{10}\dots x_{r0} + \sum_{\substack{1\le i\le r\\1\le j\le n_i}} A^{(0)}_{ij}\, x_{10}\dots x_{i-1\,0}x_{ij}x_{i+1\,0}\dots x_{r0},$$

which is obtained by homogenizing the generic affine linear form with respect to each group of variables $X_1,\dots,X_r$.

Algorithm GenericGame

Input: The number of players r and the number of pure strategies $n_1+1,\dots,n_r+1$ to each player.
Output: A (parametric) geometric resolution $\{P, W_{ij};\ 1 \le i \le r, 1 \le j \le n_i\} \subset \mathbb{Q}[C^{(i)}_{j_1,\dots,j_r}][T]$ of the set of quasi-equilibria of a generic game with the input structure.
Procedure:

1. Compute an slp encoding the resultant $\mathcal{R} = \mathrm{Res}(F_0, F^{(1)}_1,\dots,F^{(1)}_{n_1},\dots,F^{(r)}_1,\dots,F^{(r)}_{n_r})$.

2. Compute the partial derivatives $\mathcal{R}_0 = \partial\mathcal{R}/\partial A^{(0)}_0$ and $\mathcal{R}_{ij} = \partial\mathcal{R}/\partial A^{(0)}_{ij}$ for $1 \le i \le r$, $1 \le j \le n_i$.

3. Specialize $A^{(0)}_0 = T$, $A^{(0)}_{i1} = -1$ for $1 \le i \le r$, $A^{(0)}_{ij} = 0$ for $1 \le i \le r, 2 \le j \le n_i$, and $A^{(ik)}_{J_{-i}} = C^{(i)}_{j_1\dots j_{i-1}kj_{i+1}\dots j_r} - C^{(i)}_{j_1\dots j_{i-1}0j_{i+1}\dots j_r}$ for $1 \le i \le r$, $1 \le k \le n_i$ and each $J_{-i} = j_1\dots j_{i-1}j_{i+1}\dots j_r$ in the polynomials $\mathcal{R}$, $\mathcal{R}_0$ and $\mathcal{R}_{ij}$ to obtain $P, \partial P/\partial T$ and W_{ij} for $1 \le i \le r$, $1 \le j \le n_i$.

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

$$\begin{aligned}
D &:= \sum_{0\le i\le r} n_i \mathrm{Bez}_{n_1,\dots,n_r}(d_0,n_0;d_1,n_1;\dots;d_i,n_i-1;\dots;d_r,n_r) \qquad (n_0:=1, d_0:=(1,\dots,1)),\\
\delta &:= \mathrm{Bez}_{n_1,\dots,n_r}(d_1,n_1;\dots;d_r,n_r),\\
n &:= \sum_{1\le i\le r} n_i,\\
N &:= \prod_{1\le i\le r}(n_i+1)+\sum_{1\le i\le r} n_i(n_1+1)\dots(n_{i-1}+1)(n_{i+1}+1)\dots(n_r+1).
\end{aligned}$$

The algorithm obtains polynomials $P(T)$, $W_{ij}(T)\in\mathbb{Q}[C^{(i)}_{j_1\dots j_r}][T]$ *giving the geometric resolution with* $\deg_T P=\delta$, $\deg_T W_{ij}<\delta$ *and degrees bounded by* D *in the parameters* $C^{(i)}_{j_1\dots j_r}$, *which are encoded by a straight-line program of length* $O(D^2(D+n_1\dots n_r\,\delta\log(D)r^2n^4(n^3+rN)))$.

T he first step of the algorithm consists in the computation of the polynomial $\mathcal{R}$ which is the specialization of the resultant of a system of multihomogeneous polynomials of respective multidegrees $d_0=(1,\dots,1), d^{(1)}_j=(0,1,\dots,1)$ for $1\le j\le n_1,\dots,d^{(r)}_j=(1,\dots,1,0)$ for $1\le j\le n_r$, 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 Subsection A). The complexity of this algorithm is of order $O(D^2(D+n_1\dots n_r\delta\log(D)r^2n^4(n^3+rN)))$ and it computes an slp for $\mathcal{R}$ whose length is of the same order.

The algorithm to compute a geometric resolution of the zero-dimensional variety defined by the system $F^{(i)}_k=0$ $(1\le i\le r,\ 1\le k\le n_i)$ from $\mathcal{R}$ is standard: the parametrizations of the points in the variety are obtained from the partial derivatives $\mathcal{R}_0=\partial\mathcal{R}/\partial A^{(0)}_0$ and $\mathcal{R}_{ij}=\partial\mathcal{R}/\partial A^{(0)}_{ij}$ for $1\le i\le r$, $1\le j\le n_i$. An slp which encode these derivatives can be computed from the slp representing $\mathcal{R}$ within the same complexity order and length of the same order as the slp which encodes $\mathcal{R}$ (see [33]). Note that $\deg_{A^{(0)}_0}\mathcal{R}_{ij}<\delta$ for every $1\le i\le r, 1\le j\le n_i$.

Let $L:=\sum_{\substack{1\le i\le r\\ 1\le j\le n_i}} L_{ij}x_{ij}$ be a generic linear form in the variables x_{ij}, where L_{ij} are new variables, and let T be another new variable. Let $P_L\in\mathbb{Q}[A^{(ik)},L_{ij}][T]$ be the polynomial obtained by specializing

$$A^{(0)}_0\mapsto T,\ A^{(0)}_{ij}\mapsto -L_{ij}x_{i0}\ (1\le i\le r, 1\le j\le n_i) \tag{4}$$

in $\mathcal{R}$. Since $\mathcal{R}\in(F_0,F^{(i)}_k : 1\le k\le r, 1\le i\le n_k)$, substituting L for T in P_L, we obtain a polynomial $\mathcal{P}\in(F^{(i)}_k : 1\le k\le r, 1\le i\le n_k)$. As $\deg_T(P_L)=\delta$, P_L must be a multiple by a nonzero factor in $\mathbb{Q}[A^{(ik)}]$ of the minimal polynomial of L. On the other hand,

$$\frac{\partial\mathcal{P}}{\partial L_{ij}}=-\frac{\partial\mathcal{R}}{\partial A^{(0)}_{ij}}(L,-L_{ij},A^{(ik)})x_{i0}+\frac{\partial\mathcal{R}}{\partial A^{(0)}_0}(L,-L_{ij},A^{(ik)})x_{ij}$$

belongs to the ideal $(F_k^{(i)} : 1 \leq k \leq r, 1 \leq i \leq n_k)$. We conclude that making the substitution (4) in $\mathcal{R}_0$ and $\mathcal{R}_{ij}$ $(1 \leq i \leq r, 1 \leq j \leq n_i)$, polynomials which complete the geometric resolution of the variety defined by $F_k^{(i)} = 0$ with respect to the generic linear form $L := \sum_{\substack{1\leq i\leq r\\ 1\leq j\leq n_i}} L_{ij}x_{ij}$ can be obtained.

Now, we choose a separating linear form and we substitute its coefficients for the parameters L_{ij}. As the multihomogeneous system $F_k^{(i)} = 0$ is generic, it has no zeroes in the hyperplanes $x_{i0} = 0$ $(1 \leq i \leq r)$ and we can consider its zeroes as affine points by setting $x_{i0} = 1$ $(1 \leq i \leq r)$. The linear form $\ell := \sum_{1\leq i\leq r} x_{i1}$ 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 $l \in \mathbb{Q}[x_{ij}; 1 \leq i \leq r, 1 \leq j \leq n_i]$ separating these solutions. Now, making a linear change of variables in each group X_i $(1 \leq i \leq r)$, the linear form l maps to ℓ 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 ℓ. As ℓ is a separating linear form for a specific system, it is also separating for the generic one.

Hence, specializing

$$A_0^{(0)} \mapsto T,\ A_{i1}^{(0)} \mapsto -1\ (1 \leq i \leq r),\ A_{ij}^{(0)} \mapsto 0\ (1 \leq i \leq r, 2 \leq j \leq n_i)$$

in the polynomials $\mathcal{R}$, $\mathcal{R}_0$ and $\mathcal{R}_{ij}$ $(1 \leq i \leq r,\ 1 \leq j \leq n_i)$, new polynomials giving a geometric resolution of the set of common zeros of the polynomials $F_k^{(i)}$ in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$ are obtained. Finally, substituting

$$A_{J_{-i}}^{(ik)} = C^{(i)}_{j_1\dots j_{i-1}kj_{i+1}\dots j_r} - C^{(i)}_{j_1\dots j_{i-1}0j_{i+1}\dots j_r} \tag{5}$$

for every $1 \leq i \leq r$, $1 \leq k \leq n_i$ and each $J_{-i} = j_1 \dots j_{i-1}j_{i+1} \dots j_r$ with $0 \leq j_t \leq n_t$ for every $t \neq i$, the algorithm obtains polynomials $P(T)$, $\partial P/\partial T(T)$ and $W_{ij}(T)$ in $\mathbb{Q}[C^{(i)}_{j_1\dots j_r}][T]$ such that the set of quasi-equilibria of the generic game in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$ is represented as follows:

$$\Big\{\big((\frac{\partial P}{\partial T}(t) : W_{11}(t) : \cdots : W_{1n_1}(t)), \dots, (\frac{\partial P}{\partial T}(t) : W_{r1}(t) : \cdots : W_{rn_r}(t))\big) \,/\, t \in \overline{\mathbb{K}},\ P(t) = 0\Big\}, \tag{6}$$

where $\mathbb{K} := \mathbb{Q}(C^{(i)}_{j_1\dots j_r})$ (note that the linear form ℓ still separates the quasi-equilibria of the generic game). The polynomials P, $\partial P/\partial T$ and W_{ij} are encoded by an slp of length $O(D^2(D + n_1 \dots n_r\delta \log(D)r^2n^4(n^3 + rN)))$ over $\mathbb{Q}$, which is also the order of complexity of the whole computation. The upper bounds $\deg_T P \leq \delta$ and $\deg_T W_{ij} < \delta$ follow from the stated upper bounds for the degrees of $\mathcal{R}$ and $\mathcal{R}_{ij}$ in the variable $A_0^{(0)}$. □ □

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 δ 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 satisfying those conditions has δ 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 slp'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 $\mathrm{TaQ}(Q,P) = \#\{t \in \mathbb{R} : P(t) = 0,\, Q(t) > 0\} - \#\{t \in \mathbb{R} : P(t) = 0,\, Q(t) < 0\}$ 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 $0 \le h \le q$, the h-th *Sylvester-Habitch matrix* of P and Q, $\mathrm{SyHa}_h(P,Q)$, is the $(p+q-2h)\times(p+q-h)$-matrix of the polynomials $T^{q-h-1}P, \ldots, P, Q, \ldots, T^{p-h-1}Q$ in the basis $\{T^{p+q-h-1}, \ldots, T, 1\}$ and the h-th *signed subresultant coefficient*, $\mathrm{sRes}_h(P,Q)$, is the determinant of the square matrix $\widehat{\mathrm{SyHa}}_h(P,Q)$ obtained from $\mathrm{SyHa}_h(P,Q)$ by deleting the last h columns. Besides, for P and Q in $\mathbb{R}[T]$, $\mathrm{sRes}_p(P,Q)$ 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, $S_{ij}^{(h)}$, $1 \le i \le r$, $1 \le j \le n_i$, $1 \le h \le \delta$, in $\mathbb{Q}[C^{(i)}_{j_1\ldots j_r}]_{1\le i\le r,\, 0\le j_t\le n_t} \setminus \{0\}$ with total degrees bounded by $4\delta D$, such that for every vector $c := (c^{(i)}_{j_1\ldots j_r})_{1\le i\le r,\, 0\le j_t\le n_t}$ with real coordinates satisfying the conditions*

$$S_0(c) \neq 0,\ S_{ij}^{(h)}(c) > 0 \quad (1 \le i \le r,\, 1 \le j \le n_i, 1 \le h \le \delta),$$

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

The polynomials S_0 and $S_{ij}^{(h)}$ can be computed within complexity $O(\delta^2(n\delta^2+L))$ from a straight-line program of length L encoding polynomials P, W_{ij} as in 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 $c := (c^{(i)}_{j_1\ldots j_r})_{1\le i\le r,\, 0\le j_t\le n_t}$ over $\mathbb{R}$ and assume that the polynomials $P(c)(T)$ and $W_{ij}(c)(T)$ obtained from $P(T)$ and $W_{ij}(T)$ 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

$$\xi_i = \Big(\frac{P'(c)(t)}{S_i(c)(t)}, \frac{W_{i1}(c)(t)}{S_i(c)(t)}, \ldots, \frac{W_{i\,n_i}(c)(t)}{S_i(c)(t)}\Big) \qquad (1 \le i \le r),$$

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

$$\bigcap_{1\le i\le r,\, 1\le j\le n_i} \{t \in \mathbb{R} : P(c)(t) = 0,\, (P'(c)W_{ij}(c))(t) > 0\}.$$

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

$$S_0 := \mathrm{Res}_{\delta,\delta-1}(P(T), P'(T)) \in \mathbb{Q}[C_{j_1\dots j_r}^{(i)}]$$

of $P(T)$ and $P'(T)$ regarded as polynomials in the variable T with coefficients in $\mathbb{Q}[C_{j_1\dots j_r}^{(i)}]$, where the subindices indicate the degrees in T of P and P' respectively. Observe that, for every real vector $c = (c_{j_1\dots j_r}^{(i)})_{1\leq i\leq r,\, 0\leq j_t\leq n_t}$ with $S_0(c) \neq 0$, $P(c)(T)$ is a nonzero square-free polynomial of degree δ. Furthermore, the solution set of the system $G_k^{(i)}(c) = 0$, $1 \leq i \leq r, 1 \leq k \leq n_i$, is a zero-dimensional sub-variety of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$ with δ distinct points and $P(c)(T)$, $W_{ij}(c)(T)$ $(1 \leq i \leq r,\ 1 \leq j \leq n_i)$ 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 δ different totally mixed Nash equilibria if and only if, for every $1 \leq i \leq r$, $1 \leq j \leq n_i$,

$$\#\{t \in \mathbb{R} : P(c)(t) = 0,\ (P' W_{ij}(c))(t) > 0\} = \delta. \tag{7}$$

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 δ, we have that (7) is equivalent to $\mathrm{TaQ}(P'(c)W_{ij}(c), P(c)) = \delta$. By [11, Proposition 2.57], this equality is the same as $I((P'(c))^2 W_{ij}(c)/P(c)) = \delta$, which is equivalent to $I(W_{ij}(c)/P(c)) = \delta$, since $P(c)$ is a square-free polynomial. By [11, Remark 2.55], the last identity implies that $\deg_T(W_{ij}(c)) = \delta - 1$. Moreover, provided that this degree condition is met, by [11, Theorem 9.12], identity (7) is equivalent to the fact that there are no sign changes in the sequence $\mathrm{sRes}_\delta(P(c), W_{ij}(c))$, $\mathrm{sRes}_{\delta-1}(P(c), W_{ij}(c))$, $\dots$, $\mathrm{sRes}_0(P(c), W_{ij}(c))$. Thus, we define

$$S_{ij}^{(h)} = \mathrm{sRes}_h(P, W_{ij})\, \mathrm{sRes}_{h-1}(P, W_{ij}) \quad \text{for } 1 \leq h \leq \delta - 1 \quad \text{and} \quad S_{ij}^{(\delta)} = p_\delta\, \mathrm{sRes}_{\delta-1}(P, W_{ij}),$$

where P, W_{ij} are regarded as polynomials in the variable T with coefficients in $\mathbb{Q}[C_{j_1\dots j_r}^{(i)}]$ and p_δ 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 $S_{ij}^{(h)}(c) > 0$ for $1 \leq h \leq \delta$ (since $\mathrm{sRes}_{\delta-1}(P(c), W_{ij}(c))$ equals the coefficient of degree $\delta - 1$ of $W_{ij}(c)$, the condition on the degree of $W_{ij}(c)$ is ensured by $S_{ij}^{(\delta)}(c)$ being positive).

Note that each of the polynomials S_0 and $\mathrm{sRes}_h(P, W_{ij})$ 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 $\mathbb{Q}[C_{j_1\dots j_r}^{(i)}]$ of total degrees bounded by D and so, their degrees are bounded by $2\delta D$. Therefore, the degree of $S_{ij}^{(h)}$ $(1 \leq i \leq r,\ 1 \leq j \leq n_i, 1 \leq h \leq \delta)$ 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_{ij} $(1 \leq i \leq r,\ 1 \leq j \leq n_i)$, we obtain an slp of length $O(\delta^2 L)$ within the same complexity for the coefficients of P, P' and W_{ij} $(1 \leq i \leq r,\ 1 \leq j \leq n_i)$ 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 resultant S_0 and the signed subresultants required. This algorithm computes, in fact, all the coefficients of 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 $h = 1, \dots, \delta - 1$, the matrix $\widehat{\mathrm{SyHa}}_h(P, W_{ij})$ may be obtained from $\widehat{\mathrm{SyHa}}_{h-1}(P, W_{ij})$ by deleting the first and last row and the last two columns. Thus, all the signed subresultants $\mathrm{sRes}_h(P, W_{ij})$ $(1 \leq h \leq \delta - 1)$ are obtained as intermediate results in the computation of $\mathrm{sRes}_0(P, W_{ij})$ 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 slp's of length $O(\delta^2(\delta^2 + L))$ encoding the polynomials $S_{ij}^{(h)}$ within complexity $O(\delta^4)$. 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 $O(\delta^2(n\delta^2 + L))$. □ □

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 $S_{ij}^{(h)}(c) > 0$ 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 multihomogenous bound for the number of equilibra is small.

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

As before, consider an r-person non-cooperative game in normal form in which the players have $n_1 + 1, \dots, n_r + 1$ distinct available pure strategies respectively. Let $c^{(i)} := (c^{(i)}_{j_1 \cdots j_r})_{0 \leq j_k \leq n_k}$ 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 $a^{(ik)} := (a^{(ik)}_{j_1 \cdots j_{i-1} j_{i+1} \cdots j_r})_{0 \leq j_t \leq n_t}$ defined as follows $a^{(ik)}_{j_1 \cdots j_{i-1} j_{i+1} \cdots j_r} := c^{(i)}_{j_1 \cdots j_{i-1} k j_{i+1} \cdots j_r} - c^{(i)}_{j_1 \cdots j_{i-1} 0 j_{i+1} \cdots j_r}$. Thus, if $a := (a^{(ik)})_{1 \leq i \leq r,\, 1 \leq k \leq n_i}$, the set of quasi-equilibria of the game is

$$V_a := \{\xi := (\xi_1, \dots, \xi_r) \in \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r} \;/\; F_k^{(i)}(a, \xi) = 0 \;\forall 1 \leq i \leq r,\, 1 \leq k \leq n_i\}.$$

Lemma 3 *Under the previous notation, there exists an algorithm which decides whether V_a has only finitely many points within complexity $O(D^2(n_1\dots n_r)^2\,\delta(D+n_1\dots n_r\delta\log(D)r^2n^4(n^3+rN)))$.*

Proof. Consider a generic polynomial of multidegree $d_0 := (1,\dots,1)$ in the groups of variables $X_1,\dots,X_r$,

$$F_0 = \sum_{\substack{1\le i\le r\\ 0\le j_i\le n_i}} A^{(0)}_{j_1\dots j_r} x_{1j_1}\dots x_{rj_r}$$

and the resultant $R := \mathrm{Res}(F_0, F_1^{(1)},\dots,F_{n_1}^{(1)},\dots,F_1^{(r)},\dots,F_{n_r}^{(r)})$. Let $R_a(A^{(0)}_{j_1\dots j_r})$ be the polynomial obtained by substituting the coordinates of $a=(a^{(ik)})_{1\le i\le r,\,1\le k\le n_i}$ for $A^{(ik)}$ 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 $f\in\mathbb{Q}[X_1,\dots,X_r]$ 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 $\mathcal{L} := O(D^2(D+n_1\dots n_r\delta\log(D)r^2n^4(n^3+rN)))$ which encodes the multihomogeneous resultant R by using the algorithm described in [24], adapted according to Subsection A below, within complexity of the same order as $\mathcal{L}$. The specialization to obtain R_a does not modify this complexity order.

Let $f_0\in\mathbb{Q}[t][X_1,\dots,X_r]$ be the polynomial obtained by specializing the variables $A^{(0)}_{j_1\dots j_r}$ in F_0 into successive powers of a new variable t:

$$A^{(0)}_{j_1\dots j_r} = t^{j_1+(n_1+1)j_2+(n_1+1)(n_2+1)j_3+\dots+(\prod_{j=0}^{r-1}(n_j+1))j_r}. \tag{8}$$

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

$$f_0(\xi)(t) = \sum_{j_1\dots j_r}\xi_{1j_1}\dots\xi_{rj_r}t^{j_1+(n_1+1)j_2+(n_1+1)(n_2+1)j_3+\dots+(\prod_{j=0}^{r-1}(n_j+1))j_r}\in\mathbb{C}[t],$$

which is a nonzero polynomial due to the fact that there exists at least one choice of $j_1,\dots,j_r$ for which the product $\xi_{1j_1}\dots\xi_{rj_r}$ 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 $\Delta(t_0)\ne 0$. Then $f_0(t_0)\in\mathbb{Q}[X_1,\dots,X_r]$ 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 $R_a\in\mathbb{Q}[A^{(0)}_{j_1\dots j_r}]$ is zero or not, it suffices to decide whether $R_{a,t}\in\mathbb{Q}[t]$ is zero or not. Taking into account that $\deg(R_{a,t})\le(\prod_{1\le i\le r}(n_i+1)-1)\delta$, this task can be achieved by evaluating $R_{a,t}$ at $(\prod_{1\le i\le r}(n_i+1))\delta$ different values of t, which is done by substituting the powers of these values for $A^{(0)}_{j_1\dots j_r}$ as in (8) in the slp for R_a. The overall complexity of this procedure is $\Big((\prod_{1\le i\le r}(n_i+1))+\mathcal{L}\Big)(\prod_{1\le i\le r}(n_i+1))\delta$. □ □

Once we know 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

$$V_a^{\mathrm{aff}} := \{\xi\in V_a : \xi_{i0}\ne 0\ \forall\, 1\le i\le r\}$$

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 $F \in \mathbb{Q}[A_0^{(0)}, A_{ij}^{(0)}]$, $F^{(t)} \in \mathbb{Q}[t][A_0^{(0)}]$ will denote the polynomial obtained from F by specializing, as before, all the variables, except for $A_0^{(0)}$, into successive powers of a variable t.

Algorithm AffineQuasiEquilibria

Input: A family of payoff matrices $c^{(i)} := (c^{(i)}_{j_1\dots j_r})_{0\le j_k\le n_k}$ for $1 \le i \le r$ 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 $d_0 = (1,\dots,1)$, $d_j^{(i)} = (1,\dots,0,\dots,1)$ (where the 0 is in the ith coordinate) for $1 \le i \le r$ and $1 \le j \le n_i$.
2. Compute $a^{(ik)}_{j_1\dots j_{i-1}j_{i+1}\dots j_r} := c^{(i)}_{j_1\dots j_{i-1}kj_{i+1}\dots j_r} - c^{(i)}_{j_1\dots j_{i-1}0j_{i+1}\dots j_r}$ and set a for the vector with these entries.
3. Specialize $A^{(ik)} = a^{(ik)}$ in R to obtain the polynomial $R_a(A^{(0)}_{j_1\dots j_r})$.
4. Compute the leading coefficient of R_a in the variable $A^{(0)}_{0\dots 0}$ and divide R_a by this coefficient, obtaining a polynomial R_a^{aff}.
5. Specialize $A^{(0)}_{0\dots0} = A_0^{(0)}$, $A^{(0)}_{0\dots0\,j\,0\dots0} = A_{ij}^{(0)}$ (where the index j is in the ith place) for $1 \le j \le n_i$ and $A^{(0)}_{j_1\dots j_r} = 0$ otherwise in R_a^{aff}, obtaining a polynomial $\widetilde{P}_a$.
6. Compute the first nonzero subresultant of $\widetilde{P}_a^{(t)}, \frac{\partial \widetilde{P}_a^{(t)}}{\partial A_0^{(0)}}$ and an element $\tau \in \mathbb{Q}$ at which this subresultant does not vanish.
7. Compute $P_a = \widetilde{P}_a / \gcd(\widetilde{P}_a, \frac{\partial \widetilde{P}_a}{\partial A_0^{(0)}})$ and its partial derivatives.
8. Evaluate the variables $A_{ij}^{(0)}$ for $1 \le i \le r$, $1 \le j \le n_i$, at $(-\tau^k)_{0\le k\le n-1}$ 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 $n_1+1,\dots,n_r+1$ pure strategies respectively and payoff matrices $c^{(1)},\dots,c^{(r)}$ within complexity $O(\delta^8 D^2(D + n_1\dots n_r \delta \log(D) r^2 n^5 (n^3 + rN)))$ provided that the associated set of quasi-equilibria is zero-dimensional.*

S et $\mathcal{F}_a := \prod_{\xi \in V_a} F_0(\xi) \in \mathbb{Q}[A^{(0)}]$. 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 $F_0(a^{(0)})$, $F_1^{(1)}(a^{(11)}), \ldots, F_{n_r}^{(r)}(a^{(rn_r)})$ have a common root in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$. But this is equivalent to the fact that $\mathrm{Res}(F_0, F_1^{(1)}, \ldots, F_{n_r}^{(r)})$ vanishes at $a^{(0)}, a^{(11)}, \ldots, a^{(rn_r)}$ or, equivalently, that $R_a(a^{(0)}) = 0$. Then

$$R_a = C_a \prod_{\xi \in V_a} F_0(\xi)^{m_\xi}, \quad \text{with } C_a \in \mathbb{Q},\ m_\xi \in \mathbb{N}. \tag{9}$$

To describe the set V_a^{aff} of affine quasi-equilibria of the game, we compute $R_a^{\mathrm{aff}} := \prod_{\xi \in V_a^{\mathrm{aff}}} F_0(\xi)^{m_\xi}$. Note that R_a^{aff} is monic in the variable $A_{0\ldots0}^{(0)}$ and $\prod_{\xi \in V_a - V_a^{\mathrm{aff}}} F_0(\xi)^{m_\xi}$ does not depend on this variable. Then, $C_a \prod_{\xi \in V_a - V_a^{\mathrm{aff}}} F_0(\xi)^{m_\xi}$ is the leading coefficient of R_a in the variable $A_{0\ldots0}^{(0)}$.

Consider the generic form $f_0 := A_0^{(0)} x_{10} \ldots x_{r0} + \sum_{\substack{1 \le i \le r \\ 1 \le j \le n_i}} A_{ij}^{(0)} x_{10} \ldots x_{i-1\,0} x_{ij} x_{i+1\,0} \ldots x_{r0}$.

After specializing R_a^{aff} as follows:

$$\begin{array}{rcll} A_{0\ldots0}^{(0)} & \mapsto & A_0^{(0)} & \\ A_{0\ldots0\,j\,0\ldots0}^{(0)} & \mapsto & A_{ij}^{(0)} & \text{where the index } j \text{ is in the } i\text{th place } (1 \le j \le n_i) \\ A_{j_1\cdots j_r}^{(0)} & \mapsto & 0 & \text{otherwise} \end{array} \tag{10}$$

we obtain $\widetilde{P}_a := \prod_{\xi \in V_a^{\mathrm{aff}}} f_0(\xi)^{m_\xi}$. The geometric resolution of V_a^{aff} can be obtained from the square-free part $P_a := \prod_{\xi \in V_a^{\mathrm{aff}}} f_0(\xi)$ of $\widetilde{P}_a$ 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 $A_{ij}^{(0)}$.

Complexity and details of the different steps of the algorithm:

Computation of R_a^{aff}: 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 $R_a^{(t_0)} \in \mathbb{Q}[A_{0\ldots0}^{(0)}]$ be the (nonzero) polynomial obtained from R_a after specializing it as in (8) for every $(j_1, \ldots, j_r) \neq (0, \ldots, 0)$ setting $t = t_0$. Because of (9), $\deg(R_a^{(t_0)}) = d_a$. Then, in order to compute $d_a := \deg_{A_{0\ldots0}^{(0)}}(R_a)$, it suffices to compute the coefficients of $R_a^{(t_0)}$ up to degree δ (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 slp of length $O(\delta^2 \mathcal{L})$ for the coefficient of $(A_{0\ldots0}^{(0)})^{d_a}$ of R_a, R_a^{aff} 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 $O(\delta^4 \mathcal{L})$, which produces an slp 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 $G = \gcd(\widetilde{P}_a, \frac{\partial \widetilde{P}_a}{\partial A_0^{(0)}})$. For $\xi_1 \neq \xi_2$, $f_0^{(t)}(\xi_1)$ and $f_0^{(t)}(\xi_2)$ are relatively prime irreducible polynomials in $\overline{\mathbb{Q}}[A_0^{(0)}, t]$, therefore $G^{(t)} = \prod_{\xi \in V_a^{\mathrm{aff}}} f_0^{(t)}(\xi)^{m_\xi - 1} = \gcd(\widetilde{P}_a^{(t)}, \frac{\partial \widetilde{P}_a^{(t)}}{\partial A_0^{(0)}})$. Then, $\tilde{d}_a := \deg(\gcd(\widetilde{P}_a, \frac{\partial \widetilde{P}_a}{\partial A_0^{(0)}}))$ can be obtained as the degree of the gcd of the polynomials $\widetilde{P}_a^{(t)}$ and $\frac{\partial \widetilde{P}_a^{(t)}}{\partial A_0^{(0)}}$. To compute this degree, the algorithm looks for their 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 slp of length $O(\delta^4\mathcal{L}+n)$ for $\widetilde{P}_a^{(t)}$ and then an slp of length $O(\delta^2(\delta^4\mathcal{L}+n))$ 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 $O(\delta^6\mathcal{L}+\delta^2 n)$, and therefore, the whole complexity of this step is bounded by $O(\delta^8\mathcal{L}n+\delta^4 n^2)$. The polynomial $\widetilde{G} = \mathrm{s}_a \cdot \gcd(\widetilde{P}_a, \frac{\partial \widetilde{P}_a}{\partial A_0^{(0)}})$ is obtained as the $\tilde{d}_a$th polynomial subresultant of $\widetilde{P}_a$ and $\frac{\partial \widetilde{P}_a}{\partial A_0^{(0)}}$ within complexity $O(\delta^6\mathcal{L})$ (here $\mathrm{s}_a = \mathrm{sRes}_{\tilde{d}_a}(\widetilde{P}_a, \frac{\partial \widetilde{P}_a}{\partial A_0^{(0)}})$). Finally, P_a is obtained by dividing $\mathrm{s}_a\widetilde{P}_a$ by $\widetilde{G}$. Note that we already know a point τ where s_a does not vanish (the value obtained when computing $\tilde{d}_a$). Then, by evaluating the nonzero polynomial $\widetilde{G}^{(t)}(A_0^{(0)},\tau) \in \mathbb{Q}[A_0^{(0)}]$ of degree $\tilde{d}_a$ in at most $\tilde{d}_a+1$ elements in $\mathbb{Q}$, we obtain $a_0^{(0)} \in \mathbb{Q}$ such that $\widetilde{G}^{(t)}(a_0^{(0)},\tau) \neq 0$. This enables us to compute the quotient P_a by applying the algorithm in [47]. The complexity of this step is of order $O(\delta^8\mathcal{L})$.

Computation of a geometric resolution: Observe that the linear form ℓ whose coefficients are $(-\tau^k)_{0\leq k\leq n-1}$ is a separating linear form for V_a^{aff}. As in the proof of Theorem 1, the algorithm proceeds to compute the partial derivatives of P_a and evaluate P_a and its derivatives at the coefficients of ℓ. This may be done within $O(\delta^8\mathcal{L}n)$ operations.

The overall complexity of the algorithm is $O(\delta^8\mathcal{L}n+\delta^4 n^2) = O(\delta^8 D^2(D+n_1\dots n_r\delta\log(D)r^2n^5(n^3+rN)))$. □ □

Now, using standard methods (see, for instance, [49], [50]), we are able to give a description and to compute the cardinality of the set 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 $n_1+1,\dots,n_r+1$ pure strategies respectively within complexity $O(\delta^9 D^2(D+n_1\dots n_r\delta\log(D)r^2n^5(n^3+rN)))$ provided that the associated set of quasi-equilibria of the game is zero-dimensional.*

U sing 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^{aff}. Then, the totally mixed Nash equilibria of the game are the points $(\xi_1,\dots,\xi_r)$ with $\xi_i = (p'(t)/s_i(t), w_{i1}(t)/s_i(t),\dots,w_{ir}(t)/s_i(t))$, where $s_i = p' + \sum_{j=1}^{n_i} w_{ij}$ for $i=1,\dots,r$, 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 $\{t\in\mathbb{R} : p(t)=0, p'(t)>0, w_{ij}(t)>0\ \forall\, 1\leq i\leq r, 1\leq j\leq n_i\}$ and $\{t\in\mathbb{R} : p(t)=0, p'(t)<0, w_{ij}(t)<0\ \forall\, 1\leq i\leq r, 1\leq j\leq n_i\}$.

Once we have p, p' and w_{ij} $(1\leq i\leq r, 1\leq j\leq n_i\})$ encoded in dense form (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 $O(n\delta^3)$. □ □

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_a 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 $c^{(i)} := (c^{(i)}_{j_1\dots j_r})_{0\le j_k\le n_k}$ for $1 \le i \le r$ 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 $d_0 = (1, \dots, 1)$, $d^{(i)}_j = (1, \dots, 0, \dots, 1)$ (where the 0 is in the ith coordinate) for $1 \le i \le r$ and $1 \le j \le n_i$.
2. Choose a vector $b := (b^{(ik)})_{1\le i\le r, 1\le k\le n_i}$ at random.
3. Compute $a^{(ik)}_{j_1\dots j_{i-1}j_{i+1}\dots j_r} := c^{(i)}_{j_1\dots j_{i-1}kj_{i+1}\dots j_r} - c^{(i)}_{j_1\dots j_{i-1}0j_{i+1}\dots j_r}$ and set a for the vector with these entries.
4. Specialize $A^{(ik)} = a^{(ik)} + u \cdot (b^{(ik)} - a^{(ik)})$, where u is a new variable, $A^{(0)}_{0\dots 0} = A^{(0)}_0$, $A^{(0)}_{0\dots 0\, j\, 0\dots 0} = A^{(0)}_{ij}$, where the index j is in the ith place, and $A^{(0)}_{j_1\cdots j_r} = 0$ otherwise in R, obtaining $\widetilde{P}_{a+u(b-a)}$.
5. Compute by interpolation the coefficients $(p_\ell(u))_{0\le\ell\le\delta} \in \mathbb{Q}[A^{(0)}_{ij}][u]$ of $\widetilde{P}_{a+u(b-a)}(A^{(0)}_0)$.
6. For $0 \le \ell \le \delta$, compute the coefficients $(p_{\ell k})_{0\le k\le D}$ of $p_\ell(u)$ and $\epsilon_\ell = \min\{k : p_{\ell k} \ne 0\}$.
7. Compute $\epsilon = \min\{\epsilon_\ell : 0 \le \ell \le \delta\}$ and $P = \sum_{\ell=0}^{\delta} (-1)^\ell p_{\ell\epsilon} T^\ell$.
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 $n_1+1, \dots, n_r+1$ pure strategies respectively and payoff matrices $c^{(1)}, \dots, c^{(r)}$ within complexity $O(D^4\delta^3(D+n_1\dots n_r\,\delta\log(D)r^2n^4(n^3+rN)))$.*

T he 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 $b := (b^{(ik)})_{1\le i\le r, 1\le k\le n_i}$ such that $R_b \not\equiv 0$ (b can be either chosen at random or effectively constructed as the coefficient vector of a system with δ many common roots). Then, if u is a new variable, $R_{a+u(b-a)}$ is a nonzero polynomial. If $f_k^{(i)}$ is the polynomial obtained from $F_k^{(i)}$ by evaluating $x_{j0} = 1$ $(1 \le j \le r)$, let $\widetilde{f}_k^{(i)} := f_k^{(i)}(a^{(ik)} + u(b^{(ik)} - a^{(ik)}))$ $(1 \le i \le r, 1 \le k \le n_i)$. Let $\widetilde{P}_{a+u(b-a)}$ be the polynomial obtained from $R_{a+u(b-a)}$ by specializing it as in (10) and let $L := \sum_{i,j} A_{ij}^{(0)} x_{ij}$. Let $\epsilon \in \mathbb{N}_0$ such that $\widetilde{P}_{a+u(b-a)} = u^\epsilon \widetilde{P}$ for $\widetilde{P} \in \mathbb{Q}[u, A^{(0)}]$ with $\widetilde{P}(u, A^{(0)}) \mid_{u=0} \not\equiv 0$ and set $P := \widetilde{P} \mid_{u=0, A_0^{(0)}=-T}$.

Since R is a linear combination of $F_0, F_k^{(i)}$ $(1 \le i \le r, 1 \le k \le n_i)$, we have that $\widetilde{P}_{a+u(b-a)} \mid_{A_0^{(0)}=-L} = u^\epsilon \widetilde{P} \mid_{A_0^{(0)}=-L} \in (\widetilde{f}_k^{(i)} : 1 \le i \le r, 1 \le k \le n_i) \in \mathbb{Q}[u, x_{ij} : 1 \le i \le r, 1 \le j \le n_i]$. Now, each irreducible component of the variety V_u defined by the ideal $(\widetilde{f}_k^{(i)} : 1 \le i \le r, 1 \le k \le n_i)$ in $\mathbb{A}^{n+1}$ has dimension at least 1. Then, for each isolated point ξ of V_a^{aff}, $(0, \xi) \in \mathbb{A}^{n+1}$ lies in an irreducible component C of V_u such that $u \notin I(C)$. Therefore $\widetilde{P} \mid_{A_0^{(0)}=-L} \in I(C)$ and $P \mid_{T=L}$ vanishes at ξ. Then P is a multiple of the minimal polynomial of L over the set of isolated points of V_a^{aff}.

Note that, if $\widetilde{P}_{a+u(b-a)} = \sum_{\ell=0}^{\delta} p_\ell(u)(A_0^{(0)})^\ell$, where $p_\ell(u) \in \mathbb{Q}[A_{ij}^{(0)}][u]$ are such that $p_\ell(u) \not\equiv 0$ for some $0 \le \ell \le D$, then $\epsilon := \max\{k : u^k \text{ divides } p_\ell \ \forall\, 0 \le \ell \le D\}$.

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 $\mathcal{L}$ is the length of an slp encoding R, an slp of length $3N + \mathcal{L}$ for $\widetilde{P}_{a+u(b-a)}$ is obtained. Then an slp encoding the coefficients $p_\ell(u)$ $(0 \le \ell \le \delta)$ is obtained by interpolation, which takes $O(\delta^2(N + \mathcal{L}))$ operations. Afterwards, for every $0 \le \ell \le \delta$, an slp which encodes the coefficients of $p_\ell(u) = \sum_{k=0}^{D} p_{\ell k}\, u^k$ is obtained, within complexity $O(D^2\delta^2(N + \mathcal{L}))$, in order to compute $\epsilon_\ell = \min\{k : p_{\ell k} \ne 0\}$. To decide whether each of the multivariate polynomials $p_{\ell k} \in \mathbb{Q}[A_{ij}^{(0)}]$ encoded by an slp is zero or not, we apply the probabilistic Zippel-Schwartz zero test (see [53]). In this way, $\epsilon = \min\{\epsilon_\ell : 0 \le \ell \le \delta\}$ is computed. The overall complexity of this step is $O(D^2\delta^3(N + \mathcal{L}))$. The last step of this procedure obtains an slp for $P = \sum_{\ell=0}^{\delta}(-1)^\ell p_{\ell\epsilon} T^\ell$ from the slp's encoding $p_{\ell\epsilon}$ $(0 \le \ell \le \delta)$ 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_a^{aff} 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 δ 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 δ 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.

$r \diagdown n_i$	2		3		4		5		6	
2	1	1	1	1	1	1	1	1	1	1
3	2	8	10	64	56	512	346	4096	2252	32768
4	9	81	297	6561	13833	531441	748521	4.3×10^7	4.4×10^7	3.4×10^9
5	44	1024	13756	1.0×10^6	6.7×10^6	1.0×10^9	4.0×10^9	1.0×10^{12}	2.7×10^{12}	1.1×10^{15}
6	265	15625	925705	2.4×10^8	5.7×10^9	3.8×10^{12}	4.5×10^{13}	5.9×10^{16}	4.1×10^{17}	9.3×10^{20}

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 a 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. These implementation issues and the design of a deterministic algorithm for arbitrary games is the matter of future research.

A Appendix: 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 $G_0, \dots, G_n$ in r groups of variables $X_j = (x_{j0}, \dots, x_{jn_j})$, with $\sum_{j=1}^{r} n_j = n$, coincides with the sparse resultant of the dehomogenized polynomials $g_0, \dots, g_n$ obtained by setting $x_{j0} = 1$ for every $1 \leq j \leq r$.

Let $\mathcal{A}_0, \dots, \mathcal{A}_n \subset \mathbb{Z}^n$ be finite sets and let $g_0, \dots, g_n$ be polynomials with supports $\mathcal{A}_0, \dots, \mathcal{A}_n$ respectively. For any subset $J \subseteq \{0, \dots, n\}$, let $\mathcal{L}_J$ be the lattice generated by $\sum_{j \in J} \mathcal{A}_j$. Following [55], if $I \subset \{0, \dots, n\}$, the collection of supports $\{\mathcal{A}_i\}_{i \in I}$ is said to be *essential* if $\operatorname{rank}(\mathcal{L}_I) = \#I - 1$ and $\operatorname{rank}(\mathcal{L}_J) \geq \#J$ for each proper subset J of I. If there is a unique subcollection $\{\mathcal{A}_i\}_{i \in I}$ which is essential, the resultant $\operatorname{Res}(g_0, \dots, g_n)$ is not constant and coincides with the resultant $\operatorname{Res}(g_i; i \in I)$ (see [55, Corollary 1.1]).

Proposition 8 *Let $n_1, \dots, n_r$ 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 $F_0, F_1^{(1)}, \dots, F_{n_1}^{(1)}, \dots, F_1^{(r)}, \dots, F_{n_r}^{(r)}$ in r groups of $n_1+1, \dots, n_r+1$ variables respectively, where F_0 has multidegree $d_0 = (1, \dots, 1)$ and, for every $1 \leq i \leq r$, $F_k^{(i)}$ are polynomials of multidegree $d_i = (1, \dots, 0, \dots, 1)$ (with 0 in the ith coordinate), is a non-constant polynomial and can be computed algorithmically within complexity $O(D^2(D + n_1 \dots n_r \delta \log(D) r^2 n^4 (n^3 + rN)))$, where D, δ and N are as in Theorem 1.*

T he 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 $m_1, \dots, m_r \in \mathbb{N}$ and $m := m_1 + \cdots + m_r$:

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

2. $m+1$ multihomogeneous polynomials with multidegrees $(1, \dots, 1)$ in r groups of $m_1+1, \dots, m_r+1$ variables each,

3. m multihomogeneous polynomials in r groups of variables with $m_1, m_2+1, \dots, m_r+1$ 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 $F_0, F_1^{(1)}, \dots, F_{n_1}^{(1)}, \dots, F_1^{(r)}, \dots, F_{n_r}^{(r)}$ we start with satisfy the conditions in (I) with $m_i := n_i$ and $m := n$.

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

$$\mathrm{Res}(G_0^{(0)}, (G_k^{(i)})_{1\le i\le r; 1\le k\le m_i}) = \prod_{\xi \in V} g_0^{(0)}(\xi) \prod_{1\le j\le r} \mathrm{Res}((G_{kj}^{(i)})_{1\le i\le r, 1\le k\le m_i}),$$

where $g_0^{(0)}$ is the dehomogeneized polynomial obtained from $G_0^{(0)}$ by evaluating $x_{\ell\, m_\ell} = 1$ $(1 \le \ell \le r)$, V is the set of common zeros in $\mathbb{A}^m$ of the polynomials $g_k^{(i)}$ obtained in the same way from the $G_k^{(i)}$ $(1 \le i \le r, 1 \le k \le m_i)$, and, for each $1 \le j \le r$, $G_{kj}^{(i)}$ is the polynomial obtained from $G_k^{(i)}$ by setting $x_{j\, m_j} = 0$. (Note that this result, applied to the polynomials $F_0, F_1^{(1)}, \dots, F_{n_1}^{(1)}, \dots, F_1^{(r)}, \dots, F_{n_r}^{(r)}$ implies that the resultant R we want to compute is a non-constant polynomial.)

(I.a) If $m_i \ge 2$ for every $1 \le i \le r$, (up to renaming variables and polynomials) each of the resultants $\mathrm{Res}(G_{kj}^{(i)})$ 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 $\mathrm{Res}((G_{k1}^{(i)})_{1\le i\le r, 1\le k\le m_i})$ we can discard the first group of variables. Then, the resultant involves m polynomials in $r-1$ groups of $m_2+1, \dots, m_r+1$ variables each with m_i polynomials with multidegree $(1, \dots, 0, \dots, 1)$, where the 0 is in the $(i-1)$th coordinate for $2 \le i \le r$, and one with multidegree $(1, \dots, 1)$.

If, for every $2 \le i \le r$, $m_i < \sum_{1\le j\le r, j\ne i} m_j$, since $m_1 = 1$ we deduce that $m_i \le \sum_{2\le j\le r, j\ne i} m_j$ 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 $m_i = 1 + \sum_{2\le j\le r, j\ne i} m_j$ for some $2 \le i \le r$, then $m_j < m_i$ for every $j \ne i$. Therefore, the unique essential subset is $\{(i,k) : 1 \le k \le m_i\}$ and the resultant to be computed is the resultant of the corresponding family of m_i polynomials of multidegree $(1, \dots, 1)$ in $r-2$ groups of $m_2+1, \dots, m_{i-1}+1, m_{i+1}+1, \dots, m_r+1$ 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 $m_i = \sum_{j\neq i} m_j$ may be valid for at most one value i. If, on the contrary, $m_{i_1} = \sum_{j\neq i_1} m_j$ and $m_{i_2} = \sum_{j\neq i_2} m_j$ 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).

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

(III.c) If $m_1 \geq 2$ and $m_i = \sum_{j\neq i} m_j$ 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 $m_1, m_2+1,\ldots,m_{i-1}+1, m_{i+1}+1\ldots,m_r+1$ variables respectively and we are in situation (II).

(III.d) If $m_1 \geq 2$ and $m_i < \sum_{j\neq i} m_j$ for every $2 \leq i \leq r$, then $m_i \leq m_1 - 1 + \sum_{j\neq 1,i} m_j$ 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:

$$\mathrm{Res}((G_k^{(i)})_{1\leq i\leq r;1\leq k\leq m_i}) = \prod_{\xi\in W} g_1^{(1)}(\xi) \prod_{2\leq l\leq r} \mathrm{Res}((G_{kl}^{(1)})_{2\leq k\leq m_1}; (G_{kl}^{(i)})_{2\leq i\leq r,1\leq k\leq m_i}),$$

where $g_1^{(1)}$ is the dehomogeneized of $G_1^{(1)}$ by setting $x_{jm_j} = 1$ for every $1 \leq j \leq r$, W is the set of common zeros in $\mathbb{A}^{m-1}$ of the polynomials $g_k^{(1)}$ $(2 \leq k \leq m_1)$, $g_k^{(i)}$ $(2 \leq i \leq r, 1 \leq k \leq m_i)$ obtained in the same way from the $G_k^{(i)}$, and $G_{kl}^{(i)}$ is the polynomial obtained from $G_k^{(i)}$ by setting $x_{l\,m_l} = 0$.

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 $m_1'+1,\ldots,m_{l-1}'+1, m_l', m_{l+1}'+1,\ldots,m_r'+1$ variables each with m_i' polynomials of multidegree d_i for every $1 \leq i \leq r$. We have $m_1' \leq \sum_{j\neq 1} m_j'$ and, for $i \neq 1$, the condition $m_i < \sum_{j\neq i} m_j$ implies that $m_i' \leq \sum_{j\neq i} m_j'$; therefore, renaming variables and polynomials, we are again under the assumptions of (III). $\square$

$\square$

B Appendix: 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 ith coordinate. Denoting

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

the resultant R is a multihomogeneous polynomial in the coefficients of F_0, $F_k^{(i)}$ of degree δ in the coefficients of F_0 and δ_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,* $D \leq \Big(1+\sum_{1\leq i\leq r} n_i(n_i+1)\Big)\delta \leq n^2\delta.$

L et us prove that $\delta_i \leq (n_i+1)\delta$ 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 $e_1 = (1,0,\dots,0)$, we have

$$\begin{aligned}\delta_1 &= \mathrm{Bez}_{n_1,\dots,n_r}(d_1,n_1;d_2,n_2;\dots;d_r,n_r) + \mathrm{Bez}_{n_1,\dots,n_r}(e_1,1;d_1,n_1-1;d_2,n_2;\dots;d_r,n_r)\\ &= \delta + \mathrm{Bez}_{n_1,\dots,n_r}(e_1,1;d_1,n_1-1;d_2,n_2;\dots;d_r,n_r).\end{aligned}$$

Now, identity (1) implies that $\mathrm{Bez}_{n_1,\dots,n_r}(e_1,1;d_1,n_1-1;d_2,n_2;\dots;d_r,n_r) = \#\mathfrak{J}_1$, where

$$\mathfrak{J}_1 = \{(j_{11},\dots,j_{rn_r}) \,/\, j_{11}=1, j_{ik}\neq i\,\forall\,(i,k)\neq(1,1) \text{ and } \#\{j_{hk}\,/\,j_{hk}=i\} = n_i\,\forall\,1\leq i\leq r\}.$$

In order to finish the proof, we will show that $\#\mathfrak{J}_1 \leq n_1\delta$. Since δ equals the cardinality of the set $\mathfrak{J}_0$ introduced in (2), we will compare the cardinalities of both sets $\mathfrak{J}_1$ and $\mathfrak{J}_0$. To this end, we define the following map from $\mathfrak{J}_1$ to $\mathfrak{J}_0$: with a given n-tuple $j := (1, j_{12},\dots,j_{1n_1},\dots,j_{r1},\dots,j_{rn_r}) \in \mathfrak{J}_1$ we associate the n-tuple $j' \in \mathfrak{J}_0$ 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_1th coordinate. Note that a necessary condition for two distinct n-tuples in $\mathfrak{J}_1$ to lead to the same n-tuple in $\mathfrak{J}_0$ 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 $n_1+1,\dots,2n_1$. Moreover, the vector consisting of these n_1 coordinates must be of the form $(1,\dots,1,j_{hk},\dots)$ for both of them (possibly with no 1 at the beginning) and so, they can only differ in the length of the string of 1's in this vector, which ranges between 0 and n_1-1. We conclude that each element of $\mathfrak{J}_0$ is the image of at most n_1 elements of $\mathfrak{J}_1$. It follows that $\#\mathfrak{J}_1 \leq n_1\#\mathfrak{J}_0$ as we wanted to prove. □ □

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] B. Sturmfels, Solving systems of polynomial equations, CBMS Regional Conference Series in Mathematics, 97. Published for the Conference Board of the Mathematical Sciences, Washington, DC, by the 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, Aumann, R.J., Hart, S., eds., Handbook of Game Theory with Economic Applications 3 (2002) 1723–1759.

[6] H. Scarf, The approximation of fixed points of a continuous mapping, SIAM J. Appl. Math. 15 (1967) 1328–1343.

[7] R. Datta, Finding all Nash equilibria of a finite game using polynomial algebra, Econ. Theory, DOI 10.1007/s00199-009-0447-z.

[8] P. Herings, R. Peeters, A globally convergent algorithm to compute all Nash equilibria for n person games, Ann. Oper. Res. 137 (2005) 349–368.

[9] A. Sommese, C.W. Wampler, The numerical solution of systems of polynomials arising in engineering and science. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[10] R. McKelvey, A. McLennan, T. Turocy, (2007) Gambit: Software Tools for Game Theory, Version 0.2007.01.30, `http://gambit.sourceforge.net.`

[11] S. Basu, R. Pollack, M.-F. Roy, Algorithms in real algebraic geometry, Second edition, Springer Verlag, Berlin, 2006.

[12] R. Lipton, E. Markakis, Nash equilibria via polynomial equations, in: Proc. of the 6th Latin American Symposium on Theoretical Informatics, Lecture Notes in Comput. Sci. 2976, 2004, 413–422.

[13] R. McKelvey, A. McLennan, Computation of equilibria in finite games, Handbook of Computational Economics, Vol. 1 (1996) 87–142.

[14] D. Lazard, Résolution des systèmes d'équations algébriques. Theoret. Comput. Sci. 15 (1981), no. 1, 77–110.

[15] A.L. Chistov, D. Yu. Grigoriev, Subexponential-time solving systems of algebraic equations I, II, Steklov Mathematical Institute, Leningrad department, LOMI Preprints E-9-83, E-10-83, Leningrad (1983).

[16] M. Giusti, J. Heintz, La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. (French) Computational algebraic geometry and commutative algebra (Cortona, 1991), 216–256, Sympos. Math., XXXIV, Cambridge Univ. Press, Cambridge, 1993.

[17] F. Rouillier, Solving zero-dimensional systems through the rational univariate representation. Appl. Algebra Engrg. Comm. Comput. 9 (1999), no. 5, 433–461.

[18] G. Jeronimo, J. Sabia, Effective equidimensional decomposition of affine varieties. J. Pure Appl. Algebra 169 (2002), no. 2-3, 229–248.

[19] M. Giusti, J. Heintz, K. Hägele, J.E. Morais, L.M. Pardo, J.L. Montaña, Lower bounds for Diophantine approximations. Algorithms for algebra (Eindhoven, 1996). J. Pure Appl. Algebra 117/118 (1997), 277–317.

[20] M. Giusti, J. Heintz, J.E. Morais, J. Morgenstern, L.M. Pardo, Straight-line programs in geometric elimination theory. J. Pure Appl. Algebra 124 (1998), no. 1-3, 101–146.

[21] M. Giusti, G. Lecerf, B. Salvy, A Gröbner free alternative for polynomial system solving. J. Complexity 17 (2001), No. 1, 154–211.

[22] G. Lecerf, Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers. J. Complexity 19 (2003), no. 4, 564–596.

[23] D. Cox, J. Little, D. O'Shea, Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra. Third edition. Undergraduate Texts in Mathematics. Springer, New York, 2007.

[24] G. Jeronimo, J. Sabia, Computing multihomogeneous resultants using straight-line programs, J. Symbolic Comput., Volume 42, Issues 1-2 (2007), pp. 218–235.

[25] E. Schost, Computing parametric geometric resolutions. Appl. Algebra Engrg. Comm. Comput. 13 (2003), no. 5, 349–393.

[26] J. Heintz, G. Jeronimo, J. Sabia, P. Solernó, Intersection theory and deformation algorithms: the multi-homogeneous case, manuscript, 2005.

[27] R. McKelvey, A. McLennan, The maximal number of regular totally mixed Nash equilibria, J. Econom. Theory 72 (1997) No. 2, 411–425.

[28] D. Lazard, F. Rouillier, Solving parametric polynomial systems. J. Symbolic Comput. 42 (2007), no. 6, 636–667.

[29] V. Weispfenning, Comprehensive Gröbner bases. J. Symbolic Comput. 14 (1992), no. 1, 1–29.

[30] I. Shafarevich, Basic algebraic geometry, Springer-Verlag, Berlin, 1974.

[31] J, Heintz, Definability and fast quantifier elimination in algebraically closed fields, Theoret. Comput. Sci. 24 (1983) 239–277.

[32] J. von zur Gathen, Parallel arithmetic computations: a survey, in: Proc. 12th FOCS, Bratislava, Lecture Notes in Comput. Sci. 33, 1986, 93–112.

[33] P. Bürgisser, M. Clausen, M.A. Shokrollahi, Algebraic complexity theory. With the collaboration of Thomas Lickteig, Grundlehren der Mathematischen Wissenschaften, 315, Springer-Verlag, Berlin, 1997.

[34] J. Heintz, C.-P. Schnorr, Testing polynomials which are easy to compute, Monographie 30 de l'Enseignement Mathématique (1982) 237–254.

[35] M.J. Osborne, A. Rubinstein, A Course In Game Theory, MIT Press, Cambridge MA, 1994.

[36] A. Morgan, A. Sommese, C. Wampler, A product-decomposition bound for Bézout numbers. SIAM J. Numer. Anal. 32 (1995), No. 4, 1308–1325.

[37] A. McLennan, The maximum number of real roots of a multihomogeneous system of polynomial equations, Beiträge Algebra Geom. 40 (1999) No. 2, 343–350.

[38] D. N. Bernstein, The number of roots of a system of equations, Functional Anal. Appl. 9 (1975) No. 3, 183–185.

[39] M. Oka, Non-degenerate complete intersection singularity, Actualités Mathématiques, Hermann, Paris, 1997.

[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 Gröszen. Druck und Verlag von B.G. Teubner, Leipzig, 1903.

[42] C. Durvye, G. Lecerf, A concise proof of the Kronecker polynomial system solver from scratch, Expo. Math. 26 (2008), No. 2, 101–139.

[43] M. Minimair, Sparse resultant under vanishing coefficients, J. Algebraic Combin. 18 (2003) No. 1, 53–73.

[44] L. González Vega, H. Lombardi, T. Recio, M.-F. Roy, Spécialisation de la suite de Sturm. RAIRO, Inform. Théor. Appl. 28 (1994), No. 1, 1–24.

[45] T. Lickteig, M.-F. Roy, Sylvester-Habicht sequences and fast Cauchy index computation. J. Symbolic Comput. 31 (2001), No. 3, 315–341.

[46] S. J. Berkowitz, On computing the determinant in small parallel time using a small number of processors, Inform. Process. Lett. 18 (1984) No. 3, 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 (1986), no. 2, 251–264.

[50] J. Canny, Improved Algorithms for Sign Determination and Existential Quantifier Elimination, Comput. J. 36 (1993) No. 5, 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 (1988), No. 1-2, 121–129.

[53] R. Zippel, Probabilistic algorithms for sparse polynomials, in: Symbolic and algebraic computation (EUROSAM '79, Internat. Sympos., Marseille, 1979), Lecture Notes in Comput. Sci. 72, 1979, 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 (2000), No. 3, 291–307.

[55] B. Sturmfels, On the Newton polytope of the resultant, J. Algebraic Combin. 3 (1994) No. 2, 207–236.

[56] P. Pedersen, B. Sturmfels, Product formulas for resultants and Chow forms, Math. Z. 214 (1993) 377–396.