International Journal of Mathematics Trends and Technology
ISSN: 2231 – 5373 /doi:10.14445/22315373/IJMTT-V67I5P503
Volume 67 Issue 5, 27-32, May, 2021

Complexity of Solution of Simultaneous Multivariate Polynomial Equations

Duggirala Meher Krishna[#1], Duggirala Ravi[#2]

[#] *Department of Computer Science and Engineering, Gayatri Vidya, Parishad College of Engineering (Autonomous), Madhurawada, Visakhapatnam – 530048, Andhra Pradesh, India*

Abstract *— In this paper, an original reduction algorithm for solving simultaneous multivariate polynomial equations is presented. The algorithm is exponential in complexity, but the well-known algorithms, such as the extended Euclidean algorithm and Buchberger's algorithm, are superexponential. The superexponential complexity of the well-known algorithms is due to their not being "minimal" in a certain sense. Buchberger's algorithm produces a Gröbner basis. The proposed original reduction algorithm achieves the required task via computation of determinants of parametric Sylvester matrices, and produces a Rabin basis, which is shown to be minimal, when two multivariate polynomials are reduced at a time. The minimality of Rabin basis allows us to prove exponential lower bounds for the space complexity of an algebraic proof of certification, for a specific computational problem in the computational complexity class* PSPACE*, showing that the complexity classes* PSPACE *and* P *cannot be the same. It is also shown that the class of languages decidable by probabilistic algorithms with (probabilistic) polynomial time proofs for the membership of input words is not the same as the complexity class* P.

Keywords *— Extended Euclidean algorithm; Buchberger's algorithm and Gröbner basis; Parametric Sylvester matrices, Minimality and Rabin basis; Algebraic complexity theory.*

I. INTRODUCTION

Two prominent methods for reducing simultaneous multivariate polynomial equations are the extended Euclidean algorithm and Buchberger's algorithm. The reduction can be performed eliminating one variable at a time. If two multivariate polynomials vanish simultaneously, then so does their parametric gcd, where the parametric gcd is an element in the integral domain or the field of rational functions in the remaining variables. However, in general, if the parametric gcd vanishes, for a particular interpretation of variables from the algebraic closure of the ground field without any free variables, then the same cannot be guaranteed for the two multivariate polynomials, whose gcd has vanished under the chosen interpretation. The failure of the converse to hold true in general contributes to the superexponential complexity of these two well-known algorithms (Section 2).

For the two multivariate polynomials in discussion to vanish simultaneously, it is both necessary and sufficient that the determinant of the parametric Sylvester matrix, called their *resultant*, with respect to the variable being eliminated, vanishes, for any or some values in the ground field, in which the zeros are being searched for, assuming that neither of the two multivariate polynomials vanishes identically, under the chosen interpretation nullifying the resultant. The equivalence of the vanishing resultant — except when at least one of the two multivariate polynomials identically vanishes, under the interpretation of values to the variables other than the variable being eliminated in the current reduction step — to the sharing of a common zero in the algebraic closure of the ground field without parameters lends us the minimality criterion, for the reduction of the two multivariate polynomials. The entries of the Sylvester matrix being multivariate polynomials themselves, albeit without the variable being eliminated in the current step, the Gaussian elimination procedure cannot be applied, even though the final resultant, which is the determinant of the Sylvester matrix, is the same. The reason for inapplicability of the Gaussian elimination procedure for the determinant of the parametric matrices is that the computations in the intermediate stages might become excessively large, causing superexponential space and time for the completion of its computations, as can be achieved by simplification of intermediate results. The overall performance of the Gaussian elimination procedure for computation of the parametric resultant may be worse than the extended Euclidean algorithm or Buchberger's algorithm. A step-by-step simplified computation of the resultant that does not run into the space or time explosion problem, which is experienced with the Gaussian elimination method for the parametric matrices, is also presented. The reduced multivariate polynomial basis obtained by taking the resultant for each reduction step is called a *Rabin basis*, in honour of Professor Michael Oser Rabin, for his profound contributions to computer science (Section 3).

The minimality criterion allows us to derive exponential lower bounds for space requirement for an algebraic proof of certification, for a specific computational problem in two variables with undetermined coefficients over any finite field. The specific computational problem is shown to exist in PSPACE, by exhibiting an algorithm for solving it, requiring space bounded by a linear expression in the sum of the degrees of the two independent variables and logarithm of the cardinality of

the finite field, excluding the space required for the finite field arithmetic operations. It is customarily acknowledged that, for a computational problem to be in P, a polynomial time algorithm, together with a proof of certification — the proof being bounded in space by a polynomial in appropriate parameter values for the instance — must exist for its correctness of operation. The class of nondeterministic computational or decision problems, for which it is possible to produce machine checkable deterministic proofs, *bounded in space by a polynomial for each such specific computational problem*, denoted by NPSPACE-with-proof-in-PSPACE, relative to any particular fixed system of deductive or symbolic logic, equipped with rules of inference, that might be extensible, is included in NP. From this fact, it follows that the complexity classes NPSPACE-with-proof-in-PSPACE and NP are one and the same. The machine checkable proofs may include references to external facts, the rules of inference may be specialized to a specific computational or decision problem, and the extensibility is the system's or users' ability to add more rules perhaps adaptively and / or interactively. The nonexistence of a polynomial time deterministic verification algorithm for a computational problem can be inferred from the nonexistence of a proof of correctness for any such algorithm, for its solution, that is bounded in space by a polynomial in the acceptable parameter values for its instances. A discernment of Herbrand's theorem, as applied to multivariate polynomials, shows that there cannot be a shorter form for the algebraic proof of certification, for the specific computational problem in PSPACE, because an immediate reflection shows that the degree of the resultant for an instance to the computational problem under investigation, even when the number variables is only two, is exponential, which must be combined with the fact that further mod and gcd operations may have to be performed, for the completion of the proof. In particular, the algebraic form of the resultant needs to hold for all undetermined coefficients, degrees and field characteristics, for the application of Herbrand's theorem. Moreover, it is easily possible to assume multivariate polynomials with undetermined coefficients (with more than just two independent variables and undetermined coefficients) as instances for the specific computational problem in PSPACE, and again invoking Herbrand's theorem, recursion can be applied, to produce an algebraic proof of correctness for the specific computational problem in discussion. By restricting interpretation of variables to small dimension extension fields, a deterministic algorithm running in linear space (possibly excluding the space required for the finite field arithmetic operations) can be exhibited, for the specific computational problem with generalization to multiple variables. The occurrence of recursion effectively annihilates any little hope of finding a proof of certification bounded in space by a polynomial, for any deterministic algorithm for the specific computational problem in its most generality, even when the interpretation is restricted to small dimension extension fields. In summary, we have to become contented in accepting that PSPACE cannot be P. In fact, an almighty can be assumed to be capable of guessing the correct answer to the question posed as part of the specific computational problem, but the impossibility of producing a polynomial space proof of certification shows that PSPACE cannot be NP, either. In addition, to these results, it is also shown that the class of languages acceptable by probabilistic algorithms with probabilistic polynomial time proofs for the membership of an input word is not the same as the complexity class NP (Section 4).

II. EXTENDED EUCLIDEAN ALGORITHM, BUCHBERGER'S ALGORITHM AND GRÖBNER BASIS

Let $\mathbb{F}$ be a field and $\mathbb{F}[x_1, \dots, x_n]$, for some positive integer $n \geq 2$, be the integral domain of polynomials in n independent variables $x_1, \dots, x_n$, with coefficients in $\mathbb{F}$. Let $\boldsymbol{x}$ be a short notation for $\mathbf{x} = (x_1, \dots, x_n)$, $\alpha(\mathbf{x}) = \sum_{i=0}^{d} a_i(x_1, \dots, x_{n-1}) x_n^i$ and $\beta(\mathbf{x}) = \sum_{i=0}^{d} b_i(x_1, \dots, x_{n-1}) x_n^i$ be two polynomials in $\mathbb{F}[x_1, \dots, x_n]$, both of degree $d \geq 1$. It is further assumed that the polynomials $a_i(x_1, \dots, x_{n-1})$ and $b_i(x_1, \dots, x_{n-1})$ in $\mathbb{F}[x_1, \dots, x_{n-1}]$ are all nonzero, and that each requires at least $L_{\min}$ units of space, for $0 \leq i \leq d$, such that they could include more than $L_{\min} \geq 2$ terms with very diverse exponent vectors, so that their products after expansion may contain only insignificantly small number of collision terms, for applying cancellations or simplification of terms, or they may admit succinct representations requiring at least $L_{\min}$ units of space, when their products are not expanded.

The operation of the extended Euclidean algorithm for computation of the parametric gcd is explained in the sequel. Since the two input polynomials are of the same degree d in x_n, an application of two consecutive steps to eliminate the highest degree term x_n^d results in two multivariate polynomials of degree $d-1$ each, such that their coefficients would need $L_{\min}$ units of space. Now, by induction, an application of two consecutive steps to eliminate x_n^{d-i}, from the two multivariate polynomials obtained as the result of the last consecutive pair of steps by eliminating x_n^{d-i+1}, for $i = 1, 2, \dots, d$, would result in $2^i L_{\min}$ units of space, without expansion. Thus, when the products are not expanded, the overall space requirement for the elimination of x_n is at least $\boldsymbol{\mathcal{O}}(2^d L_{\min})$. One more insight is concerning the final degree of any of the variables x_i, for $1 \leq i \leq n-1$. For simplicity, let the degree of occurrence of the variable x_i, for some fixed index i, where $1 \leq i \leq n-1$, be $\delta_i \geq 2$, for each term occurring as the coefficient of x_i in either input polynomial. The elimination procedure produces coefficients as multivariate polynomials in $\mathbb{F}[x_1, \dots, x_{n-1}]$. Assuming that the occurrence of cancellations while simplifying the computations is a rare event, the degree of occurrence of the variable x_i in the parametric gcd can be lower bounded by $\boldsymbol{\mathcal{O}}(2^d \delta_i)$, for $1 \leq i \leq n-1$.

Expansion and simplification of the products formed in the intermediate steps might not produce a lot of cancellations, and would only be expected to further blow up the space requirement. A consecutive pair steps, eliminating the term x_n^d, would take at least $\boldsymbol{\mathcal{O}}(L_{\min}^2)$ space, after expansion, and by induction, a consecutive pair of steps, for eliminating x_n^{d-i+1}, by a

consecutive pair steps, would need at least space $\boldsymbol{\mathcal{O}}\left(L_{\min}^{2^i}\right)$, after expansion, for $i = 1, 2, \ldots, d$, resulting in the overall space requirement of at least $\boldsymbol{\mathcal{O}}\left(L_{\min}^{2^i}\right)$ space, after expansion. This is the problem that causes the space explosion when the extended Euclidean algorithm is applied, for eliminating a single variable, from the two input multivariate polynomials. Most of the zeros of the parametric gcd would not lead to the common zeros of the two input multivariate polynomials.

Buchberger's algorithm follows closely the operational principle of the extended Euclidean algorithm and, in effect, emulates the latter by considering the exponent vector as a whole, in the sum of terms form. The multivariate polynomials so produced are collected in the Gröbner basis [1], named after the Ph D advisor of the author, presumably connoting Hilbert's basis theorem.

III. PARAMETRIC SYLVESTER MATRICES, PARAMETRIC RESULTANT AND RABIN BASIS

Let $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ be multivariate be two polynomials in $\mathbb{F}[x_1, \ldots, x_n]$, both of degrees $d_\alpha \geq 1$ and $d_\beta \geq 1$. The Sylvester matrix corresponding to the polynomials $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$, for the elimination of the variable x_n, is a $D \times D$ matrix, where $D = d_\alpha + d_\beta$, with nonzero entries the multivariate polynomials $a_i(x_1, \ldots, x_{n-1})$, for $0 \leq i \leq d_\alpha$, and $b_j(x_1, \ldots, x_{n-1})$, for $0 \leq j \leq d_\beta$. It is assumed that the number of terms in the sum-of-terms form of expansion of the entries is at most $L_{\max}$ each for these polynomials. The resultant, denoted by $\mathrm{Res}\big(\alpha(\mathbf{x}), \beta(\mathbf{x})\big)$, with respect to the variable x_n, is the determinant of the $D \times D$ Sylvester matrix. The expansion of the determinant form as the sum of $D!$ many product terms shows that the number of terms in the resultant can be at most $D! L_{\max}^D < (DL_{\max})^D$, which is much smaller than $L_{\min}^{2^d}$, when $d_\alpha = d_\beta = d, D = 2d, L_{\min} \geq 2$ and $L_{\max}$ not too large. The observation holds even when $d_\alpha \neq d_\beta$. On the other hand, if the degree of occurrence of a variable x_i is at most Δ_i, for a fixed index i, where $1 \leq i \leq n-1$, in the multivariate polynomial coefficients of the input polynomials, then the degree of occurrence of the variable x_i in $\mathrm{Res}\big(\alpha(\mathbf{x}), \beta(\mathbf{x})\big)$, with respect to the variable x_n, is at most $D\Delta_i$, which is much smaller than $2^d \delta_i$, as found in the previous section.

However, the straightforward expansion of the determinant form results in exponential time complexity, owing to the $D!$ many terms in the sum. Similarly, the Gaussian elimination method could deliver a worse performance than the extended Euclidean algorithm, because the intermediate results may not collapse into a small number of terms until the final result.

The following propagation of computations of the determinants of smaller dimension square submatrices to larger square submatrices is useful. For the computation of the determinant of a $D \times D$ matrix, for a large dimension $D > 1$, let the determinant and inverse of a $k \times k$ submatrix, S_k, be found, where S_k is a submatrix of the $k \times D$ matrix P_k, obtained by collecting the first k rows of the matrix, inductively, for some $k \geq 2$, but $k \leq D - 1$. Let P_{k+1} be the $(k+1) \times D$ matrix obtained by adjoining the next row in the $D \times D$ matrix to P_k. Assuming that the determinant of the given $D \times D$ matrix, which is $\mathrm{Res}\big(\alpha(\mathbf{x}), \beta(\mathbf{x})\big)$, with respect to the variable x_n, for the Sylvester matrix, does not identically vanish, as an element in the integral domain $\mathbb{F}[x_1, \ldots, x_{n-1}]$, the rows of the matrix P_{k+1} are linearly independent over the field of fractions of the integral domain $\mathbb{F}[x_1, \ldots, x_{n-1}]$. By the equality of the row rank to the column rank, there are $k+1$ linearly independent columns of P_{k+1}. Now, of these linearly independent columns, k of the columns can be chosen to be those corresponding to the columns of S_k, because the columns corresponding to S_k are linearly independent, themselves, by its invertibility, and if every other column of P_{k+1} were a linear combination of the k columns of P_{k+1}, of dimensions $(k+1) \times 1$ each, corresponding to those of S_k, then the column rank P_{k+1} itself would be k. For the Sylvester matrix with respect to the variable x_n, the linear combination is taken over the field of fractions of the integral domain $\mathbb{F}[x_1, \ldots, x_{n-1}]$. Thus, at any point, if it is not possible to propagate the computation of the determinant from a $k \times k$ submatrix to $(k+1) \times (k+1)$ submatrix, for the reason that the column rank cannot increase, after adjoining any of the remaining $D - k$ rows to P_k, then the determinant of the given matrix itself vanishes, and, for the Sylvester matrix, $\mathrm{Res}\big(\alpha(\mathbf{x}), \beta(\mathbf{x})\big)$, with respect to the variable x_n, itself identically vanishes. Given S_k, $\det(S_k)$ and S_k^{-1}, the computations required for identifying an appropriate column in P_{k+1}, in order to form the $(k+1) \times (k+1)$ matrix S_{k+1}, its determinant $\det(S_{k+1})$ and its S_{k+1}^{-1} can be computed using standard formulas from matrix algebra. This method of computation of the parametric resultant avoids needless space and time explosion, that can be observed in the Gaussian elimination method.

If $\mathrm{Res}\big(\alpha(\mathbf{x}), \beta(\mathbf{x})\big) = 0$, for some interpretation of the variables $x_i = \xi_i$ in the algebraic closure of $\mathbb{F}$, for $1 \leq i \leq n-1$, and neither of $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ vanishes identically as respective single variable polynomials in x_n, for the ground instances of $x_i = \xi_i$, for $1 \leq i \leq n-1$, , then $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ share a common zero in the algebraic closure of $\mathbb{F}$. This property is called the *minimality* of the reduction step, for the elimination of the variable x_n, from the two participating multivariate polynomials. The collection of multivariate polynomials as obtained by computing the parametric resultant of two multivariate polynomials, at a time, with respect to any of the independent variables, is called a Rabin basis.

IV. PROOF SHOWING THAT PSPACE CAN BE NEITHER NP NOR P

A. *Proof Showing that* NPSPACE-with-proof-in-PSPACE = NP

For the definitions of the computational complexity classes denoted by P, NP, NPSPACE and PSPACE, the readers are referred to [5], where NPSPACE appears as NPS and PSPACE as PS. As an additional complexity class, let NPSPACE-with-proof-in-PSPACE be the collection of languages over an alphabet containing at least two symbols, such that, for the acceptability of an input word, for each language independently, a nondeterministic algorithm requiring space bounded by a polynomial (specific to the particular language) in the string length of the input word, as with NPSPACE, exists, but with an additional property that a proof of acceptance can be automatically generated with respect to any particular system of logic with its own rules of inference. Some of the rules of inference may be specialized to the particular language. The size of the proof must be bounded by some polynomial in the string length of the input word, the proof itself is assumed to be machine checkable for its validity, and the corresponding decision problem of proof checking is required to belong to P. It is easy to see that the complexity class NP is included in NPSPACE-with-proof-in-PSPACE, because of the deterministic polynomial time verification condition for the languages in NP.

For the converse inclusion, let Σ be an alphabet of at least two distinct symbols, and let $\mathcal{L} \subseteq \Sigma^*$ be a language in NPSPACE-with-proof-in-PSPACE. By assumption, there is a polynomial $p_{\mathcal{L}}(|\omega|)$, for every word $\omega \in \Sigma^*$, where $|\omega|$ is the string length of ω, such that whenever $\omega \in \mathcal{L}$, there is a proof attesting to this fact, of at most $p_{\mathcal{L}}(|\omega|)$ bits of information, relative to a particular fixed system of logic, together with the rules of inference, perhaps specialized for the language $\mathcal{L}$. The time required to check the validity of each step in the proof is taken to be bounded by a fixed, but sufficiently large, constant. The syntax checking of the proof is also assumed to require time bounded by a polynomial in the size of the proof. Thus, for every $\omega \in \mathcal{L}$, the proof that ω indeed belongs to $\mathcal{L}$ can be guessed, checked for syntactic correctness of the proof, and finally checked for the validity of the proof itself, in overall time bounded by some polynomial in $|\omega|$. For an external user, $p_{\mathcal{L}}(|\omega|)$ may still remain oblivious, as the membership of $\mathcal{L}$ to NPSPACE-with-proof-in-PSPACE requires only its existence.

B. *A Specific Computational Problem in* PSPACE *Belonging to Neither* NP *Nor* P

Let p be a large prime number, and $\mathbb{Z}_p$ be the finite field of integers with arithmetic operations mod p. Let $m,\ n \geq 5$ be positive integers, and $f(t,\ x) = \sum_{i=0}^{n-1} a_i(t)x^i + x^n \in \mathbb{Z}_p[t,\ x]$, where $a_i(t) \in \mathbb{Z}_p[t]$ are nonzero polynomials, with undetermined coefficients for the purpose of description of the computational problem, for $0 \leq i \leq n-1$. Let the degree of occurrence of t among all $a_i(t)$, for $0 \leq i \leq n-1$, be at most m, and it is assumed, for convenience, that there is exactly one polynomial of degree m, and that all the remaining $n-1$ polynomials are of degree at most $m-1$.

The computational question is, "what is the number of values of t in the algebraic closure of $\mathbb{Z}_p$, for which there can be a solution, for $x \in \mathbb{Z}_p$, such that $f(t,\ x) = 0$?" This question is akin to the problems studied along the lines of [7]. Since the polynomials $a_i(t)$, for $0 \leq i \leq n-1$, are all of degree at most m, each, it suffices to search for values of t in extension fields of $\mathbb{Z}_p$ of degree at most m. As a refinement based on this observation, the computational task is to enumerate (produce as output) the number of solutions for t in the extension field of degree d over $\mathbb{Z}_p$, for each degree d, where $1 \leq d \leq m$, such that $f(t,\ x) = 0$, for some $x \in \mathbb{Z}_p$. Obviously, the computational problem is in PSPACE. In [2], the authors discuss another similar computational problem, but do not assert whether the problem they study indeed belongs to PSPACE. Instead, their contention is restricted to the hardness of solving simultaneous multivariate polynomial equations, in general.

It is easily observed that $g(t) = \text{Res}\big(f(t,\ x), (x^p - x)\big)$ is a polynomial of degree exactly mp, and does not depend on n, by the convenient assumption made. Also, if $g(t) = 0$, nether of $f(t,\ x)$ and $(x^p - x)$ vanishes, since the coefficients of the leading degree term in both the polynomials are equal to the constant 1. However, the multiplicity of occurrence of a root of $g(t)$ must be accurately accounted for : for example, the two polynomials $\left(\phi(t) + \left(\psi(t)\right)^2 + x\right)$ and $(\phi(t) + x)$ share a common zero in the algebraic closure of $\mathbb{Z}_p$, exactly when $\left(\psi(t)\right)^2 = 0$, but each such value of t in the algebraic closure of $\mathbb{Z}_p$ must be taken into account as occurring with only multiplicity one. Thus, $h(t) = \gcd(\ g(t),\ g'(t))$, where $g'(t)$ is the formal derivative of the polynomial $g(t)$, must be computed, and finally, the degree of t in the polynomial $\frac{g(t)}{h(t)}$ yields the answer to the first question. As to the second question, the degree of gcd of $\frac{g(t)}{h(t)}$ with $\left(x^{p^d} - x\right)$ yields the number of values of t in the degree d extension of $\mathbb{Z}_p$, such that $f(t,\ x) = 0$, for some $x \in \mathbb{Z}_p$, for $1 \leq d \leq m$.

It is quite a simple matter to generalize the problem to higher dimensions. Let $f(\mathbf{x}) \in \mathbb{Z}_p[x_1, \ldots,\ x_n]$, for some integer $n \geq 3$, but requiring the number of zeros in the extensions of degree at most m_i larger than 2, for $1 \leq i \leq n$, including zeros in $\prod_{i=1}^{n} GF(\ p\ ,\ d_i)$, for all possible index vectors $(d_1, \ldots\ ,\ d_n)$\$, where $1 \leq d_i \leq m_i$ and $GF(\ p\ ,\ d_i)$ is the degree d_i extension of $\mathbb{Z}_p$. It is worth noting that a comparison of m_i to the degree of occurrence of x_i is omitted, for the

purpose of stating the problem in its most generality. The enumeration problem can be easily shown to be in PSPACE, because m_i , for $1 \leq i \leq n$, are fixed inputs to the instance. If it is required to consider values for x_i in degree d_i extensions of $\mathbb{Z}_p$, that are not in any smaller dimension extension, then the condition as to whether $\left(x^{p^j} - x\right) \neq 0$ holds, for $1 \leq j \leq d_i - 1$, must be checked for. Of course, a primitive element can be chosen in degree d_i extensions of $\mathbb{Z}_p$, and the primitive element can be rised to integer powers, such that the exponents are relatively prime to $(p^{d_i} - 1)$. Now, in the algebraic proof, if such a condition must be expressed, then the principle of inclusion and exclusion must also be applied, in addition to the division by gcd with derivatives, as may be required, for example, in the expression for the Möbius inversion formula. An algebraic proof of validity of the output of the PSPACE algorithm for this problem --- *i.e*., proof of certification for the validity of its output as a specific indicator of an algebraic expression --- would be definitely enormously large in its size.

Accommodating more equations and rising the same question concerning the number of solutions to the systems of simultaneous multivariate equations, in general, the contention that PSPACE $\neq$ NP can be more aptly testified, because the solution space cannot be bounded by a polynomial, disallowing any claim of producing a direct polynomial time algebraic proof, for the validation of the answer produced by a nondeterministic algorithm.

The philosophic question under investigation is whether there can be a shorter form of solution for the computational problem exhibited to be in PSPACE. The answer to the question is that the algebraic form, depicted as the solution to the problem, holds for all the prime numbers occurring in place of the field characteristics and for all the other indeterminate parameters as part of the problem instances — where it may be recalled that the degree of the resultant of $f(x, t)$ and $(x^p - x)$ in the first problem in two variables, x and t, did not depend on the degree n of x — and hence there cannot be a shorter form, by Herbrand's theorem. This situation should not be confused with the way the determinant is computed. In the case of the determinant, there was an easier way to compute it, and gradually, it is shown to be equal to the corresponding multilinear alternating form, and in contrast, in the context of the specific computational problem shown to belong to PSPACE, there is simply no alternative algebraic proof attesting the validity of the solution produced by the algorithm requiring space bounded by a polynomial in appropriate values of the parameters of the problem instances.

C. Implications of the Fact that IP = PSPACE

In [6], it is shown that **IP = PSPACE** , where **IP** is the class of computational problems that can be probabilistically proved for the validation of an assumed solutions, interactively. Savitch's theorem [5] shows that NPSPACE = PSPACE. Allowing for the nondeterministic choices of space bounded by a polynomial in the string length of the input word, the class of languages that admit probabilistic polynomial time proofs, by nondeterminstic polynomial space algorithms, is exactly IP. Let Probabilistic-P be the class of languages acceptable by probabilistic algorithms with probabilistic polynomial time proofs attesting the membership of an input word each such language. Tentatively, if it is assumed that Probabilistic-P = P, then clearly, it must be the case that IP = NPSPACE = PSPACE = NP. However, PSPACE $\neq$ NP, by the discussion of the last subsection, Thus, Probabilistic-P $\neq$ P.

V. CONCLUSIONS

This paper presents an original reduction method for solving simultaneous multivariate polynomial equations, by eliminating one variable, from two equations, taken at a time. The reduction method is shown to satisfy a certain minimality criterion, and hence becomes optimal in respect of the constraints stated. A mathematical problem that is in PSPACE but that which cannot be in either of NP and P is also presented. As an afterthought, the trace of execution of an algorithm, even allowing for nondeterministic choices of sizes bounded by some polynomial in the appropriate values of the input parameters for the instances, must also be bounded in size by some polynomial in those parameter values. The trace may be supplied as the input to a debugger program for validation of its operation. This part may be included in the polynomial time certification for the algorithm, for each given input instance. The contentments assert that PSPACE $\neq$ NP and PSPACE $\neq$ P, affirmatively. For probabilistic algorithms, the class of languages decidable by deterministic algorithms with probabilistic polynomial time proofs for the membership of an input word is not the same as the complexity class P.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge fruitful discussions with Professor Michael Oser Rabin, Professor Ronald Linn Rivest and Professor Adi Shamir. The second author received introductions about machine checkable proofs and polynomial time verification certifications during a stint of postdoctoral position under the supervision of Professor Amir Pnueli, at the Weizmann Institute of Science, during the year 2002.

REFERENCES

[1] Bruno Buchberger, An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal, Ph. D. Thesis, University of Innsbruck (1965), English translation by M. Abramson in Journal of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions, 41(3) (2006) 475-511.

[2] D. Castro, M. Giusti, J. Heintz, G. Matera, and L. M. Pardo, The Hardness of Polynomial Equation Solving, Foundations of Computational Mathematics, 3(4) (2003), 347-420.

[3] J.-C. Faugère, A New Efficient Algorithm for Computing Gröbner Bases (F4), Journal of Pure and Applied Algebra, 139(1) (1999), 61-88.

[4] J.-C. Faugère, A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5), Proc. International Symposium on Symbolic and Algebraic Computation, ACM Press (2002) 75-83.

[5] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Education (2007).

[6] Adi Shamir, IP = PSPACE, Journal of the ACM, 39(4) (1992), 869-877.

[7] André Weil, Number of Solutions of Equations in Finite Fields, Bulletin of the American Mathematical Society, 55(5) (1949), 497-508.