On binary solutions to systems of linear equations over a field of characteristic zero

Alexandr V. Seliverstov

Abstract. Let us consider the generic-case complexity. The machine halts at every input and gives a meaningful answer at almost every input, but it can abandon the calculation using explicit notification, that is, there exists the vague halting state. A generic-case polynomial-time algorithm is proposed to recognize systems of linear equations without any binary solution, when the number of equations is close to the number of unknowns.

A sequence is called binary when it belongs to the set $\{0,1\}^*$. Let us consider the problem whether there exists a binary solution to a system of inhomogeneous linear equations with integer coefficients. The problem is NP-complete and can be reduced to its particular case containing only one linear equation [1]. Furthermore, a binary solution to one linear equation can be found using a pseudopolynomial-time algorithm [1, 2]. Without any restriction on the coefficients, Horowitz and Sahni [3] had introduced the meet-in-the-middle approach and gave an exact $O^*(2^{n/2})$ time and space algorithm. A few years later, Schroeppel and Shamir [4] improved the space complexity to $O^*(2^{n/4})$. There is also known a polynomial upper bound on the average-case complexity of the multidimensional knapsack problem [5].

By means of eliminating variables, searching for a binary solution to a system of m linearly independent linear equations in n unknowns is reduced to a parallel check whether a binary solution to a subsystem in $n-m$ unknowns can be extended to a binary solution to the whole system of equations in n unknowns. Hence, the initial problem is polynomial-time solvable when the difference between the number of unknowns and the number of linearly independent equations is bounded by a function of the type $n - m = O(\log n)$. Let us consider the case when the difference between the number of unknowns n and the number of equations m is bounded by a function of the type $n - m = O(\sqrt{n})$. So, the previously obtained estimates are improved, although the proposed method is generally useless for one equation.

An easy generalization of this problem is searching for binary solutions to a system of linear equations over an arbitrary field $(K, 0, 1, +, -, \times, ()^{-1}, =)$ of characteristic zero. Let us define $0^{-1} = 0$. In contrast to previous works [5, 6], we consider not only ordered fields but also arbitrary fields of characteristic zero, including the field of complex numbers. Let us use either generalized register machines [7] or BSS-machines over reals [8]. These machines over an algebraic extension of the field of rational numbers naturally correspond to the idea of symbolic computations. Every register contains an element of K. The machine also has index registers containing non-negative integers. The running time is polynomial when the total number of operations performed by the machine is bounded by a polynomial in the number of registers containing the input. Initially, this number is written in the zeroth index register.

A predicate holds almost everywhere when it holds on every instance x satisfying an inequality of the type $f(x) \neq 0$, where f denotes a nonzero polynomial. This restriction is more rigorous than any upper bound on the measure. Let us consider so-called generic generalized register machines over K. The machine halts at every input and gives a meaningful answer at almost every input, but it can abandon the calculation using explicit notification, that is, there exists the vague halting state [6]. More precisely, a generalized register machine over K is called generic when two conditions hold: (1) the machine halts at every input and (2) for every positive integer k and for almost all inputs, each of which occupies exactly k registers, the machine accepts or rejects the input, but does not halt in the vague state. Generic machines that compute non-trivial output in registers are defined similarly. If the machine halts in the vague state, then the output recorded in the registers is considered meaningless. Note that the machine does not make any error. For detailed description of generic computation on classical computational models refer to [9, 10].

Without loss of generality, let us consider systems of linear equations of the type $x_j = \ell_j(1, x_1, \cdots, x_{n-m})$, where $j > n - m$ and every $\ell_j(x_0, x_1, \cdots, x_{n-m})$ denotes a linear form over K.

Theorem 1. *Given two positive integers n and m satisfying the inequality*

$$2n \geq (n - m + 1)(n - m + 2).$$

For almost every m-tuple of linear forms $\ell_j(x_0, \cdots, x_{n-m})$, where $j > n-m$, there exist a set of coefficients λ_k such that the equality

$$\sum_{k=1}^{n-m} \lambda_k x_k (x_k - x_0) + \sum_{j=n-m+1}^{n} \lambda_j \ell_j (\ell_j - x_0) = x_0^2$$

holds. Moreover, for every n there exists a polynomial of degree at most $2n$ in coefficients of all the linear forms ℓ_j such that if the set of coefficients λ_j does not exist, then the polynomial vanishes.

Proof. The quest is a solution to an inhomogeneous system of linear equations in n unknowns $\lambda_1, \ldots, \lambda_n$. The system contains only one inhomogeneous equation. Let

us denote by r the number of all the equations, i.e., $r = \frac{1}{2}(n-m+1)(n-m+2) \leq n$. The sufficient condition for the solvability is the full rank of a $r \times n$ matrix.

If $r = n$, then it is sufficient that the determinant does not vanish. If $r < n$, then it is sufficient that some $r \times r$ minor does not vanish. For example, let us pick up the leading principal minor. In any case, it is a polynomial of degree r in matrix entries. Every entry is a polynomial of degree at most two in coefficients of some ℓ_j. Thus, the minor is a polynomial of degree at most $2r \leq 2n$. To complete the proof, we only need to show that this polynomial does not vanish identically, cf. [6]. □

Theorem 2. *There exists a polynomial time generic generalized register machine over K such that for all positive integers n and m satisfying the inequality*

$$2n \geq (n - m + 1)(n - m + 2),$$

and for almost every m-tuple of linear forms $\ell_j(x_0, \cdots, x_{n-m})$, where $j > n - m$, if the machine accepts the input, then there exists no binary solution to the system of all equations of the type $x_j = \ell_j(1, x_1, \cdots, x_{n-m})$. Moreover, for every n there exists a polynomial of degree at most $2n$ in coefficients of all the linear forms ℓ_j such that if the machine halts in vagues halting state, then the polynomial vanishes.

Proof. If $2n < (n - m + 1)(n - m + 2)$, then the machine rejects the input. Else, in accordance with Theorem 1, some polynomial time generic machine calculates numbers $\lambda_1, \ldots, \lambda_n$ such that the equality

$$\sum_{k=1}^{n-m} \lambda_k x_k(x_k - 1) + \sum_{j=n-m+1}^{n} \lambda_j \ell_j(\ell_j - 1) = 1$$

holds. On the other hand, if there exists a binary solution to the system of all the equations $x_j = \ell_j(1, x_1, \cdots, x_{n-m})$, then the left-hand polynomial vanishes at the binary solution. Therefore, an affirmative answer confirms that there is no binary solution to the system. Otherwise, the machine halts in the vague halting state. The estimate for the degree of a polynomial that vanishes in these cases coincides with the estimate from Theorem 1. □

Remark. Over the field of rational numbers, not only the arithmetic complexity but also the bit complexity is polynomial because the rank can be easily computed [1]. So, there is a polynomial-time generic-case algorithm. Moreover, the rank of a sparse matrix can be computed faster [11]. On the other hand, the rank can be computed in $O(\log^2 n)$ operations over an arbitrary field using a polynomial number of processors [12].

Acknowledgments. The reported study was funded by RFBR according to the research project no. 18-29-13037.

References

[1] Schrijver A. *Theory of linear and integer programming.* John Wiley & Sons, New York, 1986.

[2] Koiliaris K., Xu C. Faster pseudopolynomial time algorithms for subset sum. *ACM Transactions on Computation Theory*, 2019, vol. 15, no. 3, article 40. https://doi.org/10.1145/3329863

[3] Horowitz E., Sahni S. Computing partitions with applications to the knapsack problem. *Journal of the Association for Computing Machinery*, 1974, vol. 21, no. 2, pp. 277–292. https://doi.org/10.1145/321812.321823

[4] Schroeppel R., Shamir A. A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems. *SIAM Journal on Computing*, 1981, vol. 10, no. 3, pp. 456–464. https://doi.org/10.1137/0210033

[5] Kuzyurin N.N. An algorithm that is polynomial in the mean in integer linear programming. *Sibirskii Zhurnal Issledovaniya Operatsii*, 1994, vol. 1, no. 3, pp. 38–48 (in Russian). http://mi.mathnet.ru/da493

[6] Seliverstov A.V. Binary solutions to large systems of linear equations. Submitted to *Prikladnaya Diskretnaya Matematika* (in Russian).

[7] Neumann E., Pauly A. A topological view on algebraic computation models. *Journal of Complexity*, 2018, vol. 44, pp. 1–22. https://doi.org/10.1016/j.jco.2017.08.003

[8] Blum L., Shub M., Smale S. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society* (N.S.), 1989, vol. 21, no. 1, pp. 1–46. https://doi.org/10.1090/S0273-0979-1989-15750-9

[9] Miasnikov A., Ushakov A. Generic case completeness. *Journal of Computer and System Sciences*, 2016, vol. 82, no. 8, pp. 1268–1282. https://doi.org/10.1016/j.jcss.2016.05.002

[10] Rybalov A.N. On generic complexity of the subset sum problem for semigroups of integer matrices. *Prikladnaya Diskretnaya Matematika*, 2020, no. 50, pp. 118–126. https://doi.org/10.17223/20710410/50/9

[11] Cheung H.Y., Kwok T.C., Lau L.C. Fast matrix rank algorithms and applications. *Journal of the ACM*, 2013, vol. 60, no. 5, article 31. https://doi.org/10.1145/2528404

[12] Chistov A.L. Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic. In: Budach L. (eds) *Fundamentals of Computation Theory. FCT 1985. Lecture Notes in Computer Science*, vol. 199. Springer, Berlin, Heidelberg, 1985. https://doi.org/10.1007/BFb0028792

Alexandr V. Seliverstov
Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute)
Moscow, Russia
e-mail: slvstv@iitp.ru