Binary Solutions to Some Systems of Linear Equations

Alexandr V. Seliverstov(✉)

Institute for Information Transmission Problems of the Russian Academy of Sciences
(Kharkevich Institute), Bolshoy Karetny per. 19, build.1, Moscow 127051, Russia
slvstv@iitp.ru

Abstract. A point is called binary if its coordinates are equal to either zero or one. It is well known that it is hard to find a binary solution to the system of linear equations whose coefficients are integers with small absolute values. The aim of the article is to propose an effective probabilistic reduction from the system to the unique equation when there is a small difference between the number of binary solutions to the first equation and the number of binary solutions to the system. There exist nontrivial examples of linear equations with small positive coefficients having a small number of binary solutions in high dimensions.

Keywords: Subset sum · Linear equation · Probabilistic algorithm
Computational complexity

1 Introduction

Let us consider a system of linear equations over integers. The problem of the existence of a $(0, 1)$-solution to the system is NP-complete [1]. The $(0, 1)$-solution is also referred to as either binary or Boolean one.

In case of the unique equation, one can either find some $(0, 1)$-solution or prove the absence of such solutions, using dynamic programming [2–6]. Moreover, the number of $(0, 1)$-solutions to the linear equation over integers can be computed in pseudopolynomial time [3]. On the other hand, the problem is to solve the system that consists of the linear equation and the set of quadratic equations $x_1^2 = x_1, \ldots, x_n^2 = x_n$. If there is no solution, then a direct proof of the insolvability of the system by means of Hilbert's Nullstellensatz requires to produce polynomials of high degree [7]. All known methods for solving systems of algebraic equations require at least exponential time in general case [8,9]. There exists a one-to-one correspondence between the $(0, 1)$-solutions and singular points of the effectively computed cubic hypersurface [10]. Some singular points can be found by means of the method described in [11]. Another approach to the problem is based on L-class enumeration algorithm [12].

This research has been supported by the Russian Science Foundation, Project No. 14–50–00150.

A. Eremeev et al. (Eds.): OPTA 2018, CCIS 871, pp. 183–192, 2018.
https://doi.org/10.1007/978-3-319-93800-4_15

There is also the related optimization problem to find the maximum of the linear functional on the set of $(0, 1)$-points satisfying a unique inequality. It is called the knapsack problem. There are well known both fully polynomial time approximation scheme and pseudopolynomial time algorithm for solving the problem. The obstacle for solving the optimization problem is a large number of values of the linear functional at different $(0, 1)$-points. If all coefficients are small positive integers, then the linear functional is bounded. Thus, the set of its values at $(0, 1)$-points is small. Howbeit, the NP-complete problem seems insolvable in polynomial time. Moreover, the polynomial hierarchy is infinite relative to a random oracle with probability one [13].

2 Preliminaries

The running time of the algorithm is the number of arithmetic operations ($+$, $-$, and $\times$) as well as of verifications of two binary predicates $=$ and $<$ over integers. The $\widetilde{O}$ notation suppresses a factor that is polylogarithmic in the input size. The O^* notation suppresses a factor that is polynomial in the input size.

The symbol $\mathbf{x}$ denotes the integer sequence $(x_1, \ldots, x_n)$. Both k and j are integer so that $k \leq m$ and $j \leq n$, where m is an integer.

The number of $(0, 1)$-solutions to the linear equation $\beta + \alpha_1 x_1 + \ldots + \alpha_n x_n = 0$ over integers is equal to the coefficient of the monomial $t^{-\beta}$ of the univariate Laurent polynomial

$$F(t) = \prod_{j=1}^{n} (1 + t^{\alpha_j})$$

In case the j-th coefficient α_j is negative, one can make the linear transformation $x_j \mapsto 1 - x_j$. Thus, without loss of generality, one can assume that all coefficients α_j are positive, that is, the Laurent polynomial $F(t)$ is a polynomial.

Proposition 1 (Smolev [3]). *The number of $(0, 1)$-solutions to the linear equation $\beta + \alpha_1 x_1 + \ldots + \alpha_n x_n = 0$ over integers can be computed in pseudopolynomial time $O(n^3 a)$, where $a = \max_j |\alpha_j|$.*

So, the counting problem seems to be as hard as the recognition problem, that is, whether there exists a $(0, 1)$-solution to the linear equation. If all coefficients α_j are positive, the recognition problem coincides with the subset sum problem.

Remark 1. The subset sum problem can be solved in exponential time $O^*(2^{n/2})$ as well as in exponential space $O^*(2^{n/2})$ according to [14]. On the other hand, it can be solved in probabilistic time $O^*(2^{0.86n})$ and in polynomial space [15].

The running time of the algorithm solving the subset sum problem by means of dynamic programming is bounded by $O(n^2 a \log_2(na))$, where $a = \max_j |\alpha_j|$. Furthermore, in case the coefficients α_k are large, if the difference between $\max_k \alpha_k$ and $\min_k \alpha_k$ is bounded by a polynomial in n, then the subset sum problem can be solved in polynomial time [3].

There are some ways to improve the upper bound. In accordance with [5], in case $n < |\beta|$, the problem can be solved in pseudopolynomial time $\tilde{O}(\sqrt{n}|\beta|)$. In accordance with [6], the problem can be solved by a probabilistic algorithm in pseudopolynomial time $O(n + |\beta| \log_2 |\beta| \log_2^3(n/\varepsilon) \log_2 n)$ with error probability at most ε. On the other hand, the subset sum problem can be solved in polynomial space $\tilde{O}(n^2)$ and in pseudopolynomial time $\tilde{O}(n^3|\beta| \log_2 |\beta|)$ according to [16]. There exists another space-efficient algorithm [17].

Let us consider linear forms $\alpha_1 x_1 + \cdots + \alpha_n x_n$ over integers, where the greatest common divisor $\mathrm{GCD}(\alpha_1, \ldots, \alpha_n) = 1$. The greatest coefficient that appears in such linear forms vanishing on a set of $n-1$ linearly independent $(0,1)$-points is at most $2^{-n}(\sqrt{n+1})^{n+1}$. The upper bound is based on the inequality for determinants [18]. It is almost tight [19,20]. So, the distribution of the number of $(0,1)$-solutions as a function in β is complicated [21].

Proposition 2 [1,22]**.** *Given the system of m linear equations $\beta_k + \alpha_{k1}x_1 + \cdots + \alpha_{kn}x_n = 0$ over integers. The set of $(0,1)$-solutions to the system coincides with the set of $(0,1)$-solutions to the unique equation*

$$\sum_{k=1}^{m} \gamma_k \left(\beta_k + \sum_{j=1}^{n} \alpha_{kj} x_j \right) = 0,$$

where integers $\gamma_k = (an + b + 1)^{k-1}$, $a = \max_{k,j} |\alpha_{kj}|$, and $b = \max_k |\beta_k|$.

Remark 2. On the other hand, in accordance with Proposition 1 as well as Remark 1, if all coefficients of the unique linear equation belong to a small segment near zero, then the equation can be solved by dynamic programming. Therefore, it is important to look for the coefficients γ_k as small as possible.

Propositions 2 and 1 together provide an algorithm whose running time is exponential in the number m. Let us compare the algorithm with what is obtained as a result of elimination of m variables. In this case, the absolute values of the coefficients of the resulting linear equation can rapidly increase during the process of elimination. This method allows to quickly find all $(0,1)$-solutions to the system only under the condition $n - m = O(\log_2 n)$.

Proposition 3. *There exists an algorithm that accepts the system of m independent linear equations $\beta_k + \alpha_{k1}x_1 + \cdots + \alpha_{kn}x_n = 0$ if and only if it has some $(0,1)$-solution. The running time of the algorithm is bounded by $O(nm^2 + nm2^{n-m})$.*

Proof. Elimination of m variables produces a linear equation that depends on at most $n - m$ variables. Therefore, it suffices to go over all $(0,1)$-points of $(n-m)$-dimensional space and to verify for each of them whether it corresponds to the $(0,1)$-solution to the input system. □

So, the most difficult case is when $n \approx 2m$.

Remark 3. If some linear equation of the system has small coefficients and a small number of $(0,1)$-solutions, then one can compute the list of all $(0,1)$-solutions to the equation by means of a binary search tree. Next, one can check step by step whether a $(0,1)$-solution from the list is the solution to the system. But the task is more difficult, when there are sufficiently many $(0,1)$-solutions to each linear equation.

3 Main Results

Theorem 1. *Given the positive number ε and the system of $m \geq 2$ linear equations $\ell_k(\mathbf{x}) = 0$ over integers, where $\ell_k(\mathbf{x}) = \beta_k + \alpha_{k1}x_1 + \cdots + \alpha_{kn}x_n$. Assume the first linear equation has at most μ redundant $(0,1)$-solutions, which do not satisfy the system. If all random integers $\eta_2,\ldots,\eta_m$ are independent and uniformly distributed over the set from zero up to the number $N = \lceil \mu/\varepsilon \rceil$, then the probability that each $(0,1)$-solution to the linear equation*

$$(Nm(an+b)+1)\ell_1(\mathbf{x}) + \sum_{k=2}^{m} \eta_k \ell_k(\mathbf{x}) = 0$$

satisfies the system is at least $1-\varepsilon$, where $a = \max_{k,j} |\alpha_{kj}|$ and $b = \max_k |\beta_k|$.

Proof. If either the first equation has no $(0,1)$-solution or each $(0,1)$-solution to the first equation satisfies the whole system, then the desired result is obvious. Else let us define a subset of the set of all $(0,1)$-points

$$\mathcal{S} = \{\mathbf{x} \in \{0,1\}^n : \ell_1(\mathbf{x}) = 0 \wedge (\exists k \leq m)\ell_k(\mathbf{x}) \neq 0\}.$$

The cardinality of the set $\mathcal{S}$ is at most μ. Let us define the polynomial

$$f(y_2,\ldots,y_m) = \prod_{\mathbf{x}\in\mathcal{S}} \left(\sum_{k=2}^{m} \ell_k(\mathbf{x}) y_k \right)$$

In particular, if the set $\mathcal{S}$ is empty, then one can set $f = 1$. If a sequence $\gamma_2,\ldots,\gamma_m$ increases sufficiently fast, then $f(\gamma_2,\ldots,\gamma_m)$ does not vanish, consequently, the polynomial f does not vanish identically. Note that $\deg f \leq \mu$.

Let random integers η_k be independent and each η_k is uniformly distributed over the set $\{0,\ldots,N\}$. In accordance with the Schwartz–Zippel lemma [23], the probability of vanishing $f(\eta_2,\ldots,\eta_m)$ is at most ε.

In case $f(\eta_2,\ldots,\eta_m) \neq 0$, to prove that the system has no redundant $(0,1)$-solution, it is sufficient to prove that there exists no redundant $(0,1)$-solution to the following system of two linear equations

$$\begin{cases} \ell_1(\mathbf{x}) = 0 \\ \eta_2\ell_2(\mathbf{x}) + \cdots + \eta_m\ell_m(\mathbf{x}) = 0 \end{cases}$$

In turn, a $(0,1)$-point is the solution to the system if and only if it satisfies the unique linear equation

$$(Nm(an+b)+1)\ell_1(\mathbf{x}) + \sum_{k=2}^{m} \eta_k \ell_k(\mathbf{x}) = 0.$$

In particular, if $\mu = 0$, then the equation coincides with the first equation. □

Remark 4. The number N can be replaced by another large number. So, without loss of generality one can assume $N = 2^\nu - 1$, where ν is integer. In this case, random numbers can be identified with sequences of independent random bits. There exist other methods for calculating random variables by coin tossing, cf. [24].

Remark 5. Of course, instead of the first equation one can use the sum $h(\mathbf{x})$ of both the first equation and a linear combination of all other equations having small coefficients. But this $h(\mathbf{x})$ must be explicitly defined.

Next, let us consider a Las Vegas algorithm what uses random integers while it is running, but always either returns the correct answer or never halts.

Theorem 2. *There exists a zero-error probabilistic algorithm such that for each integer $\mu \geq 0$ and for each system of $m \geq 2$ linear equations $\ell_k(\mathbf{x}) = 0$ over integers, where $\ell_k(\mathbf{x}) = \beta_k + \alpha_{k1}x_1 + \cdots + \alpha_{kn}x_n$, if the first linear equation has at most μ redundant $(0,1)$-solutions, which do not satisfy the system, then the algorithm returns the linear equation $h(\mathbf{x}) = 0$ over integers, where*

$$\begin{aligned} h(\mathbf{x}) &= (2\mu m(an+b)+1)\ell_1(\mathbf{x}) + \gamma_2\ell_2(\mathbf{x}) + \cdots + \gamma_m\ell_m(\mathbf{x}) \\ a &= \max\nolimits_{k,j} |\alpha_{kj}| \\ b &= \max\nolimits_{k} |\beta_k| \\ (\forall k)\gamma_k &\leq 2\mu \end{aligned}$$

so that each $(0,1)$-solution to the equation $h(\mathbf{x}) = 0$ is the solution to the system. In the case, the running time of the algorithm is pseudopolynomial in expectation. If the condition for μ is false, then the algorithm either returns an equation $h(\mathbf{x}) = 0$ or never halts.

Proof. Let $\eta_2, \ldots, \eta_m$ be independent random integers from zero to 2μ.

At first, the algorithm chooses these random integers, sets

$$h(\mathbf{x}) = (2\mu m(an+b)+1)\ell_1(\mathbf{x}) + \eta_2\ell_2(\mathbf{x}) + \cdots + \eta_m\ell_m(\mathbf{x}),$$

and computes the number λ_0 of $(0,1)$-solutions to the equation $h(\mathbf{x}) = 0$ by means of Proposition 1. For each $1 \leq k \leq m$, it computes the number λ_k of $(0,1)$-solutions to the system of two equations $h(\mathbf{x}) = 0$ and $\ell_k(\mathbf{x}) = 0$ by means of Propositions 2 and 1.

If for all k the equation $\lambda_0 = \lambda_k$ holds, then the algorithm returns the current equation $h(\mathbf{x})$, where for all k the coefficients $\gamma_k = \eta_k$.

Otherwise the algorithm repeats the same computation with new choice of random integers $\eta_2,\dots,\ \eta_m$.

If the number μ satisfies the condition, then the probability that the algorithm returns a correct answer is at least $\frac{1}{2}$ at each round according to Theorem 1. The probability of there is no correct answer in a long series of repeats is small. Thus, the expected running time is almost as small as the running time of one round of the algorithm. □

Theorem 3. *Given the system of m linear equations $\ell_k(\mathbf{x}) = 0$ over integers, where $m > r > 0$. Assume the subsystem of equations $\ell_1(\mathbf{x}) = 0,\dots,\ \ell_r(\mathbf{x}) = 0$ has at most μ redundant $(0,1)$-solutions, which do not satisfy the system. There exist integers $\gamma_{r+1},\dots,\ \gamma_m$ belonging to the segment from zero up to the integer μ such that each $(0,1)$-solution to the new system of linear equations $\ell_1(\mathbf{x}) = 0,\dots,$ $\ell_r(\mathbf{x}) = 0$, and $\gamma_{r+1}\ell_{r+1}(\mathbf{x}) + \cdots + \gamma_m\ell_m(\mathbf{x}) = 0$ satisfies the initial system.*

Proof. If either the considered subsystem has no $(0,1)$-solution or each $(0,1)$-solution to the subsystem satisfies the whole system, then the desired result is obvious. Else let us define a subset of the set of all $(0,1)$-points

$$\mathcal{S} = \{\mathbf{x} \in \{0,1\}^n : \ell_1(\mathbf{x}) = 0 \wedge \cdots \wedge \ell_r(\mathbf{x}) = 0 \wedge (\exists k \le m)\ell_k(\mathbf{x}) \ne 0\}.$$

The cardinality of the set $\mathcal{S}$ is at most μ. Let us define the polynomial

$$f(y_{r+1},\dots,y_m) = \prod_{\mathbf{x}\in\mathcal{S}} \left(\sum_{k=r+1}^{m} \ell_k(\mathbf{x}) y_k \right)$$

In particular, if the set $\mathcal{S}$ is empty, then one can set $f = 1$. If a sequence $\gamma_{r+1},\dots,\gamma_m$ increases sufficiently fast, then $f(\gamma_{r+1},\dots,\gamma_m)$ does not vanish, consequently, the polynomial f does not vanish identically. On the other hand, the inequality $\deg f \le \mu$ holds. In accordance with the Schwartz–Zippel lemma [23], there exist desired integers $\gamma_{r+1},\dots,\ \gamma_m$ belonging to the segment from zero up to the integer μ. □

4 Discussion

In case a correct value for μ is known, either Theorems 1 or 2 together with Proposition 1 provide the probabilistic algorithm to enumerate $(0,1)$-solutions to the system of linear equations over integers because each solution to the system satisfies all linear combinations of the equations. The first algorithm halts in one-sided error polynomial time. The second algorithm does not make errors.

If a $(0,1)$-solution exists, then it can be found by binary search. Moreover, all $(0,1)$-solutions can be listed in this way. Any substitution for a variable by either zero or one does not increase the number of solutions. Thus, the reduction of dimension require at most $2n$ steps. If all coefficients α_{kj} are nonnegative, then the search of $(0,1)$-solutions to the system can be improved by means of new algorithms for the subset sum problem, which are listed in Remark 1.

The algorithms can be useful for small both a and μ. The restriction on both values $a = \max_{k,j} |\alpha_{kj}|$ and $b = \max_k |\beta_k|$ is not crucial. The recognition problem of the existence of a $(0, 1)$-solution to the system is NP-complete in case $a = 1$ without any restriction on the number of solutions, that is, $\mu = 2^n$. The reduction is obvious [1]. Thus, the linear system in n variables can be reduced to the another linear system in $O(n \log_2(ab))$ variables such that new coefficients have small absolute values. Furthermore, in case a is small, the running time of the algorithms depends weakly on b because without loss of generality one can assume the inequality $b \leq an$ holds. Otherwise the system has no $(0, 1)$-solution. But the upper bound on the value μ is crucial.

If the first equation of the system has a small number of $(0, 1)$-solutions, then μ can be chosen small too. But in the case, one can to check all these $(0, 1)$-solutions by means of the deterministic algorithm. Nontrivial case is when each equation has many $(0, 1)$-solutions, but there is a small difference between the number of $(0, 1)$-solutions to the first equation and the number of $(0, 1)$-solutions to the system.

There exist at most 2^{n-m} binary solutions to the system of m linearly independent linear equations in n variables. In accordance with Proposition 1, the number λ_1 of $(0, 1)$-solutions to the first equation can be found in pseudopolynomial time. So, there is the lower bound on the value $\mu \geq \lambda_1 - 2^{n-m}$. Another way to obtain the lower bound on the value μ is to compute the upper bound on the dimension of the affine hull of $(0, 1)$-solutions to the first equation of the system.

Note that μ can be a rough upper bound on the difference between the total number of $(0, 1)$-solutions to the first equation and the number of $(0, 1)$-solutions to the system. On the other hand, it is hard to compute this difference. Otherwise, it would be easy to calculate the number of $(0, 1)$-solutions to the system, that is, to solve the hard counting problem.

Of course, if all absolute values of the coefficients α_j are small integers, then there exists a number β such that the linear equation $\beta + \alpha_1 x_1 + \cdots + \alpha_n x_n = 0$ has at least $2^n/(1 + na)$ binary solutions, where $a = \max_j |\alpha_j|$. Let us consider examples of linear equations with small positive coefficients having a few $(0, 1)$-solutions. In particular, if the first equation of the system coincides with one of exemplified equations, then one can use a small value of μ.

Example 1. If all the coefficients α_k are strictly positive, then there exists exactly one $(0, 1)$-solution to the equation

$$\sum_{j=1}^{n} \alpha_j x_j = \sum_{j=1}^{n} \alpha_j,$$

that is, $(1, \dots, 1)$. Moreover, if the inequality $\alpha_1 + \cdots + \alpha_{n-1} < \alpha_n$ holds, then the equation

$$\sum_{j=1}^{n} \alpha_j x_j = \sum_{j=1}^{n-1} \alpha_j$$

has exactly one $(0, 1)$-solution, that is, $(1, \dots, 1, 0)$.

Example 2. If $n = p + q$, where $p \neq q$ and both numbers p and q are prime, then there exist exactly two $(0,1)$-solutions to the equation

$$\sum_{j=1}^{p} qx_j + \sum_{j=p+1}^{p+q} px_j = pq.$$

These antipodal points are $(1,\dots,1,0\dots,0)$ and $(0,\dots,0,1,\dots,1)$, where the number of zeros is equal to either p or q. The equations $x_1 = x_2 = \dots = x_p$ hold because

$$q\sum_{j=1}^{p} x_j \equiv 0 \pmod{p}.$$

The equations $x_{p+1} = x_{p+2} = \dots = x_n$ hold because

$$p\sum_{j=p+1}^{p+q} x_j \equiv 0 \pmod{q}.$$

The maximum of the linear form over the set $\{0,1\}^n$ is equal to $2pq$.

Example 3. If $n = p + q + 1$, where $p \neq q$ and both numbers p and q are prime, then there are exactly three $(0,1)$-solutions to the equation

$$\sum_{j=1}^{p} qx_j + \sum_{j=p+1}^{p+q} px_j + pqx_n = pq.$$

These points are $(1,\dots,1,0\dots,0,0)$, $(0,\dots,0,1,\dots,1,0)$, and $(0,\dots,0,1)$. The maximum of the linear form over the set $\{0,1\}^n$ is equal to $3pq$.

Example 4. If $n = p + q + r$, where $p < q < r$ and the numbers p, q, and r are prime, then there are exactly three $(0,1)$-solutions to the equation

$$\sum_{j=1}^{p} qrx_j + \sum_{j=p+1}^{p+q} prx_j + \sum_{j=p+q+1}^{p+q+r} pqx_j = pqr.$$

The maximum of the linear form over the set $\{0,1\}^n$ is equal to $3pqr$.

In this way, one can construct other examples with arbitrary given number of $(0,1)$-solutions for almost all n. Linear transformations of coordinates of the type $x_j \mapsto 1 - x_j$ allow constructing other examples with coefficients of different signs.

The abundance of such examples allows to hope that the discussed algorithm can find practical application, in particular, in bioinformatics and economics [25].

Theorem 3 provides an improvement of the Proposition 2. If there exists a subsystem with a small number of redundant $(0,1)$-solutions, which do not satisfy the system, then one can reduce the number of equations without a considerable increment of absolute values of its coefficients. Unfortunately, it requires guessing this subsystem. Assume the initial system has no $(0, 1)$-solution. At first, it can be reduced to the new system according to Theorem 3. Next, it can be reduced to the unique equation according to Proposition 2. At last, one can count the number of $(0, 1)$-solution according to Proposition 1. So, this particular instance of the $coNP$-complete problem can be solved by the non-deterministic algorithm. Of course, if the hypothesis $NP \neq coNP$ holds, then the running time of the algorithm must be sufficiently large in some cases.

The same result is also applicable to the case of $(-1, 1)$-solutions, that is, solutions to the set partition problem.

Acknowledgements. The author would like to thank the anonymous reviewers for useful comments.

References

1. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1986)
2. Dantzig, G.B.: Discrete-variable extremum problems. Oper. Res. **5**(2), 266–277 (1957)
3. Smolev, V.V.: On an approach to the solution of a Boolean linear equation with positive integer coefficients. Discrete Math. Appl. **3**(5), 523–530 (1993). https://doi.org/10.1515/dma.1993.3.5.523
4. Tamir, A.: New pseudopolynomial complexity bounds for the bounded and other integer Knapsack related problems. Oper. Res. Lett. **37**(5), 303–306 (2009). https://doi.org/10.1016/j.orl.2009.05.003
5. Koiliaris, K., Xu, C.: A faster pseudopolynomial time algorithm for subset sum. In: SODA 2017 Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1062–1072. Society for Industrial and Applied Mathematics, Philadelphia (2017)
6. Bringmann, K.: A near-linear pseudopolynomial time algorithm for subset sum. In: SODA 2017 Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1073–1084. Society for Industrial and Applied Mathematics, Philadelphia (2017)
7. Margulies, S., Onn, S., Pasechnik, D.V.: On the complexity of Hilbert refutations for partition. J. Symbolic Comput. **66**, 70–83 (2015). https://doi.org/10.1016/j.jsc.2013.06.005
8. Chistov, A.L.: An improvement of the complexity bound for solving systems of polynomial equations. J. Math. Sci. **181**(6), 921–924 (2012). https://doi.org/10.1007/s10958-012-0724-4
9. Jeronimo, G., Sabia, J.: Sparse resultants and straight-line programs. J. Symbolic Comput. **87**, 14–27 (2018). https://doi.org/10.1016/j.jsc.2017.05.005
10. Latkin, I.V., Seliverstov, A.V.: Computational complexity of fragments of the theory of complex numbers. Bull. Karaganda Univ. Math. **1**, 47–55 (2015). (In Russian). http://vestnik.ksu.kz
11. Seliverstov, A.V.: On tangent lines to affine hypersurfaces. Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki **27**(2), 248–256 (2017). (In Russian). https://doi.org/10.20537/vm170208

12. Kolokolov, A.A., Zaozerskaya, L.A.: Finding and analysis of estimation of the number of iterations in integer programming algorithms using the regular partitioning method. Russian Math. (Iz. VUZ) **58**(1), 35–46 (2014). https://doi.org/10.3103/S1066369X14010046
13. Håstad, J., Rossman, B., Servedio, R.A., Tan, L.-Y.: An average-case depth hierarchy theorem for Boolean circuits. J. ACM **64**(5), 35 (2017). https://doi.org/10.1145/3095799
14. Horowitz, E., Sahni, S.: Computing partitions with applications to the knapsack problem. J. ACM **21**(2), 277–292 (1974). https://doi.org/10.1145/321812.321823
15. Bansal, N., Garg, S., Nederlof, J., Vyas, N.: Faster space-efficient algorithms for subset sum and k-sum. In: STOC 2017 Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pp. 198–209 ACM, New York (2017). https://doi.org/10.1145/3055399.3055467
16. Lokshtanov, D., Nederlof, J.: Saving space by algebraization. In: STOC 2010 Proceedings of the Forty-second ACM Symposium on Theory of Computing, pp. 321–330. ACM, New York (2010). https://doi.org/10.1145/1806689.1806735
17. Gál, A., Jang, J.-T., Limaye, N., Mahajan, M., Sreenivasaiah, K.: Space-efficient approximations for Subset Sum. ACM Trans. Comput. Theory **8**(4), 16 (2016). https://doi.org/10.1145/2894843
18. Williamson, J.: Determinants whose elements are 0 and 1. Am. Math. Monthly **53**(8), 427–434 (1946)
19. Alon, N., Vũ, V.H.: Anti-Hadamard matrices, coin weighing, threshold gates and indecomposable hypergraphs. J. Comb. Theory A **79**(1), 133–160 (1997). https://doi.org/10.1006/jcta.1997.2780
20. Babai, L., Hansen, K.A., Podolskii, V.V., Sun, X.: Weights of exact threshold functions. In: Hliněný, P., Kučera, A. (eds.) MFCS 2010. LNCS, vol. 6281, pp. 66–77. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15155-2_8
21. Gorbunov, K.Yu., Seliverstov, A.V., Lyubetsky, V.A.: Geometric relationship between parallel hyperplanes, quadrics, and vertices of a hypercube. Probl. Inform. Transm. **48**(2), 185–192 (2012). https://doi.org/10.1134/S0032946012020081
22. Williams, R.: New algorithms and lower bounds for circuits with linear threshold gates. In: STOC 2014 Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing, pp. 194–202. ACM, New York (2014). https://doi.org/10.1145/2591796.2591858
23. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. J. ACM **27**(4), 701–717 (1980). https://doi.org/10.1145/322217.322225
24. Bacher, A., Bodini, O., Hwang, H.-K., Tsai, T.-H.: Generating random permutations by coin-tossing: classical algorithms, new analysis, and modern implementation. ACM Trans. Algorithms **13**(2), 24 (2017). https://doi.org/10.1145/3009909
25. Beresnev, V.L., Melnikov, A.A.: An upper bound for the competitive location and capacity choice problem with multiple demand scenarios. J. Appl. Ind. Math. **11**(4), 472–480 (2017). https://doi.org/10.1134/S1990478917040020