ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2023, Vol. 63, No. 1, pp. 48–56. © Pleiades Publishing, Ltd., 2023.
Russian Text © The Author(s), 2023, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2023, Vol. 63, No. 1, pp. 51–60.

GENERAL NUMERICAL METHODS

Generalization of the Subset Sum Problem and Cubic Forms

A. V. Seliverstov*

Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, 127051 Russia

**e-mail: slvstv@iitp.ru*

Received April 19, 2022; revised May 12, 2022; accepted September 10, 2022

Abstract—A new algorithm is proposed for deciding whether a system of linear equations has a binary solution over a field of zero characteristic. The algorithm is efficient under a certain constraint on the system of equations. This is a special case of an integer programming problem. In the extended version of the subset sum problem, the weight can be positive or negative. The problem under consideration is equivalent to the analysis of solution existence for several instances of this problem simultaneously. New sufficient conditions are found under which the computational complexity of almost all instances of this problem is polynomial. In fact, the algorithm checks the existence of a cubic hypersurface that passes through each vertex of the unit cube, but does not intersect a given affine subspace. Several heuristic algorithms for solving this problem have been known previously. However, the new methods expand the solution possibilities. Although only the solution existence problem is considered in detail, binary search allows one to find a solution, if any.

Keywords: integer programming, linear equation system, sum of subsets, average-case complexity

DOI: 10.1134/S0965542523010116

1. INTRODUCTION

The subset sum problem is to decide whether or not a linear equation has a $\{0,1\}$-solution. It is a well-known example of NP-complete problems. The problem with n variables is solvable in $O(2^{n/2})$ arithmetic operations (see [1]). A generally accepted conjecture states that this problem is solvable at worst in exponential time. This lower bound was checked using some computational models with constraints, including linear machines [2, 3] and additive machines [4], as well as by applying methods for checking the solvability of systems of algebraic equations based on the theorem about polynomials over a real closed field (Positivstellensatz) [5] or on Hilbert's Nullstellenzatz [6]. On the other hand, this problem can be solved in pseudopolynomial time by applying the dynamic programming method (see [7, 8]). Heuristic algorithms are available in the low-density case (see [9, 10]). This problem can be generalized to residue rings (modular case) [11] and multiplicative semigroups of matrices (see [12]).

We consider the problem of the existence of a $\{0,1\}$-solution to a system of linear equations. The worst-case computational complexity of this problem is the same as for the subset sum problem with a single equation. However, fast heuristic algorithms are known for systems of sufficiently many equations with a constraint on the coefficient signs (see [13, 14]), with coefficients of any sign (see [15]), or under other constraints (see [16, 17]). Some algorithms are surveyed in the next section.

Heuristic solution methods or approximations are known for close optimization problems, including the multidimensional knapsack problem [18, 19], the modular variant of this problem [20], and mixed knapsack and covering problems [21]. They can also be solved by applying general optimization methods (see [22, 23]). Heuristic local search algorithms are also available for verifying the satisfiability of a propositional conjunctive normal form (CNF) [24, 25], specifically, 3-CNF [26]. The same method can be used to solve the corresponding optimization problem (see [27]). On the other hand, only the exponential-time upper bound $O^*(1.2989^m)$ is known for the worst-case complexity of the maximum satisfiability problem for a CNF formula with m clauses (see [28]). The CNF satisfiability problem is equivalent to the existence of a $\{0,1\}$-solution to a system of linear inequalities with the same number of variables over an ordered ring of integers. Therefore, the worst-case complexity bounds for these problems are related to each other.

Concerning the average-case complexity, we assume that a probability measure is defined on a set of inputs of arbitrarily fixed length. Each such set is usually finite, so all inputs in it can be equiprobable. The average-case complexity is defined as the expectation of the computational complexity on inputs of the given length (see [29]).

The main result of this paper is a heuristic algorithm proposed for the problem of recognizing systems of linear equations that do not have a {0,1}-solution. The algorithm is applicable under weaker constraints than earlier algorithms. Additionally, it can be used in parallel with other methods. The results are valid over an arbitrary field of characteristic zero with efficiently computable operations. To simplify the exposition, the computations are usually considered over the field of rational numbers or over a purely transcendental extension of this field. However, computations with algebraic numbers (see [30]) can also be efficiently performed using algorithms for treating polynomials (see [31]).

2. PRELIMINARIES

Consider a system of m linear equations in $n > m$ variables. By eliminating variables, it is easy to derive a new equation in $n - m$ variables. If $n - m = O(\log_3 n)$, then the problem of searching for a {0,1}-solution to the original system of equations is solvable in polynomial time. It reduces to the search for each {0,1}-solution to an equation in $O(\log_2 n)$ variables and to the check of whether this solution can be extended to a {0,1}-solution to the original system. In what follows, we consider less trivial cases when this problem is easily solvable.

Let $\mathbf{b}$ denote a column and A denote a rectangular $m \times n$ matrix with nonnegative entries. All {0,1}-solutions to the system of inequalities $A\mathbf{x} \leqslant \mathbf{b}$ can be found by applying the dynamic programming method, the running time of which is low if the total number of such solutions is small. In [13, 14] Kuzyurin showed that, for $m > 9\log_2 n$ under some assumptions on the entry distributions in A and $\mathbf{b}$, the average number of such solutions is bounded above by a polynomial in n. Therefore, the average running time of the algorithm is also polynomial. The proof is based on estimating the tail of the binomial distribution. Given all solutions to the system of inequalities, among them we can easily choose a solution to the system of equations, if it exists. An important limitation to the applicability of this method is the requirement that the entries of A be nonnegative. By making changes of variables of the type $y_k = 1 - x_k$, a system of equations with arbitrary coefficients can easily be reduced to a system of equations with nonnegative coefficients at the linear terms, so that the new and original systems have an identical number of {0,1}-solutions. However, the original distribution of coefficients is then distorted.

Given a system of m linear equations in n variables with positive integer coefficients a_k, the density is defined by the formula

$$\varrho = \frac{n}{m\log_2 \max\limits_k a_k}.$$

For $m = 1$, assuming that the density is low, i.e., $\varrho < 0.9408$, almost all instances of the subset sum problem are solvable in polynomial time by searching for the shortest nonzero vector in a certain lattice (see [9, 10]). This method can be generalized to systems of many equations with the same density constraint (see [16]).

Following [15, 32], the fraction of inputs of given size on which the algorithm quickly makes the correct decision can be estimated using the Schwartz–Zippel lemma (see [33]).

Lemma 1. *Given a nonconstant polynomial $f(x_1,\ldots,x_n)$ of degree d over a field K, if ξ_1, ..., ξ_n are independent uniformly distributed random variables on a finite set $S \subseteq K$ of cardinality $|S|$, then*

$$\Pr[f(\xi_1,\ldots,\xi_n) = 0] \leqslant \frac{d}{|S|},$$

where $\Pr[\cdot]$ *denotes the probability that the condition in square brackets holds.*

Recently, this result has been strengthened to univariate polynomials over the field of complex numbers. There exists a set of $2d + 1$ numbers such that any two polynomials f and g of degree d are equal to each other if the range of f is embedded in the range of g on this set (see [34]).

The rank of a matrix over a field can be computed by applying a polynomial number of processors, each executing only $O(\log_2^2 n)$ operations over this field (see [35, 36]). On the other hand, upper bounds for the

complexity of computing the rank of a matrix are close to the complexity of matrix multiplication (see [37, 38]). In this paper, examples are computed in the MathPartner computer algebra system (see [39]). The determinant of a matrix is calculated by the \det() command, and the rank, by the \rank() command. In Maple, the determinant, rank, and permanent of a matrix can be computed, for example, in the LinearAlgebra package.

3. RESULTS

The geometric interpretation of the problem is to check whether the affine subspace defined by the system of linear equations passes through a vertex of the unit n-cube. Each such vertex corresponds to a $\{0,1\}$-solution. According to the applied method, an attempt is made to construct an algebraic hypersurface of low degree that passes through each vertex of the unit cube, but does not intersect the given affine subspace. Here, the hypersurface can be reducible. The existence of such a hypersurface implies that the original system of equations has no $\{0,1\}$-solution. The computational complexity of this method depends on the degree of the hypersurface. The application of quadrics in the case of subspaces of low dimension was studied in [15]. On the other hand, for a general hyperplane, the union of 2^n parallel hyperplanes containing all vertices of the unit n-cube is the desired hypersurface of degree at most 2^n. However, the degree can be lower. Specifically, if the hyperplane is defined by an equation with bounded above positive integer coefficients that has no $\{0,1\}$-solutions, then a small number of parallel hyperplanes is sufficient. This circumstance underlies the dynamic programming method for solving the subset sum problem. However, we consider arbitrary hypersurfaces that are not necessarily a union of hyperplanes.

3.1. Straight Lines in the Projective Plane

First, we consider a straight line in the plane, which corresponds to the search for a $\{0,1\}$-solution to an inhomogeneous linear equation in two variables x_1 and x_2. Its homogenization is an equation in three variables x_0, x_1, and x_2. Each $\{0,1\}$-solution corresponds to a vertex of the unit square. In the projective plane, these vertices have the homogeneous coordinates $(1:0:0)$, $(1:0:1)$, $(1:1:0)$, and $(1:1:1)$, respectively. In these coordinates, the straight line at infinity is given by the equation $x_0 = 0$. The projective curve is defined by a ternary form (homogeneous polynomial) vanishing at points of the curve.

Theorem 1. *Given a square in the projective plane, a straight line L passes through none of its vertices if and only if there exists a cubic curve passing through each vertex of the square, but crossing L only at a point in the straight line at infinity.*

Proof. Suppose that the vertices of the square have homogeneous coordinates $(1:0:0)$, $(1:0:1)$, $(1:1:0)$, and $(1:1:1)$. The projective cubic curve is defined by a ternary cubic form. Vanishing at each vertex of the square, this form is equal to a linear combination of six forms $x_k x_j(x_j - x_0)$, where $k \in \{0,1,2\}$ and $j \in \{1,2\}$. These forms are linearly independent. Therefore, the dimension of the linear space of cubic forms vanishing at each vertex of the square is 6 (see [40]). The dimension of the space of all binary cubic forms is equal to 4. The restriction of a form to the line L defines a linear mapping π from the space of ternary forms vanishing at each vertex of the square to the space of all binary forms. The kernel (null space) of π consists of forms vanishing identically on L. Each form from the kernel is reducible and is divisible by the linear form ℓ defining L. Since, by assumption, L passes through none of the vertices of the square, the dimension of the kernel (nullity) is equal to the dimension of ternary quadratic forms vanishing at each vertex of the square. According to [40], these forms are of the type $\lambda_1 x_1(x_1 - x_0) + \lambda_2 x_2(x_2 - x_0)$. Therefore, the nullity of π is equal to 2. Hence, the dimension of the image of π coincides with the dimension of all binary forms. The mapping π is surjective. Specifically, the image of a cubic form $c(x_0, x_1, x_2)$ from the domain of π is the form x_0^3. The equation $c(x_0, x_1, x_2) = 0$ defines a curve that crosses L only at a point at infinity. Moreover, since the nullity is equal to 2, there exists a one-parameter family of such curves.

This argument relies heavily on the assumption that the straight line L passes through none of the square vertices. Otherwise, the kernel of π would contain the product of the linear form ℓ and a quadratic form vanishing only at three vertices of the square. Then the nullity is higher than 2 and the mapping π is not surjective. Obviously, in this case, the straight line L crosses each curve from the domain of π at some vertex of the square. The theorem is proved.

In Theorem 1, the cubic curve cannot be replaced by a conic. Indeed, ternary quadratic forms vanishing at each vertex of the square span a two-dimensional linear space, while binary quadratic forms span a three-dimensional space. Therefore, the analogue of π from the proof of Theorem 1 is no longer surjective.

3.2. Linear Spaces of Forms

The dimension of the linear space of forms of degree d in variables $x_0, \ldots, x_s$ is equal to the binomial coefficient $\binom{s+d}{d}$. Consider linear combinations of cubic forms of the type $x_k(x_k - x_0)x_t$. Each such form vanishes at every point with coordinates $x_0 = 1$ and $x_k \in \{0,1\}$, where $1 \leqslant k \leqslant n$. The restriction of these forms to the linear subspace defined by the system of equations $x_j = \ell_j(x_0,\ldots,x_s)$, where $s < j \leqslant n$, is a linear combination of forms of two types: either $x_k(x_k - x_0)x_t$ or $\ell_j(\ell_j - x_0)x_t$, where $1 \leqslant k \leqslant s$ and $0 \leqslant t \leqslant s$.

Lemma 2. *Given a set of linear forms $\ell_j(x_0,\ldots,x_s) = a_{j0}x_0 + \cdots + a_{js}x_s$, where $s < j \leqslant n$ and all the coefficients a_{ji} are algebraically independent elements of a purely transcendental extension of the field of rational numbers of transcendence degree $(n-s)(s+1)$, if $(s+3)(s+2) \leqslant 6(n-s)$, then each cubic form in the variables $x_0, \ldots, x_s$ is a linear combination of forms of the type $\ell_j^2 x_t$, where $s < j \leqslant n$ and $0 \leqslant t \leqslant s$.*

Proof. Let M denote the matrix composed of the coefficients of forms of the type $\ell_j^2 x_t$, where the columns correspond to monomials, and the rows, to forms. The total number of forms of the type $\ell_j^2 x_t$ must be at least the number of monomials of the third degree, i.e.,

$$\frac{(s+3)!}{s!3!} \leqslant (n-s)(s+1),$$

where the left-hand side is the total number of third-degree monomials and the right-hand side is the number of forms of the type $\ell_j^2 x_t$. This relation is equivalent to the inequality $(s+3)(s+2) \leqslant 6(n-s)$ from the condition of the lemma. Under this inequality, it suffices to show that the matrix M has a full rank equal to the number of columns.

Consider a square submatrix M' of M with the same number of columns and with a set of nonzero entries, one in each row and each column. The determinant of M' is equal to the alternating sum of products of entries from the sets with one entry from each row and each column. The purely transcendental field extension is isomorphic to the field of rational functions. The entries of M' can be treated as polynomials in the variables a_{jk} ordered according to the order on indices:

$$a_{(s+1)0} > a_{(s+1)1} > \cdots > a_{j0} > a_{j1} > \cdots > a_{js} > \cdots > a_{ns}.$$

This defines a monomial ordering. In M' we choose a set of nonzero entries, one in each row and each column, for which the product is maximal under this monomial ordering. This maximum is attained for a unique set of entries of M'. Indeed, if two distinct sets of this type have equal products of entries, then we can find a third set with a larger product of entries.

Since a_{jk} are algebraically independent, none of their products is equal to a linear combination of other products. Therefore, the determinant of M' is nonzero. Hence, the matrix M is of full rank. The lemma is proved.

The elements of the purely transcendental field extension can be identified with functions of independent variables a_{ji}. Lemma 2 holds for some sparse matrices obtained by substituting integer values for a_{ji}.

Let $n = 3$ and $s = 1$. We choose $\ell_2 = x_0$ and $\ell_3 = x_1$. The coefficient matrix of cubic forms of the type $\ell_j^2 x_t$ is nonsingular, since it is equal to the 4×4 identity matrix. Since the linear space of binary cubic forms is four-dimensional, this space is spanned by forms of the type $\ell_j^2 x_t$.

Let $n = 6$ and $s = 2$. We choose $\ell_3 = x_0$, $\ell_4 = x_1$, $\ell_5 = x_2$, and $\ell_6 = x_1 + x_2$. The coefficient matrix M of cubic forms of the type $\ell_j^2 x_t$ consists of 12 rows and 10 columns. The rank of M is 10, which coincides with the dimension of the linear space of ternary cubic forms.

Let $n = 8$ and $s = 3$. We choose $\ell_4 = x_0$, $\ell_5 = x_1$, $\ell_6 = x_2$, $\ell_7 = x_3$, and $\ell_8 = x_0 + x_1 + x_2 + x_3$. The cubic forms of the type $\ell_j^2 x_t$ have a square coefficient matrix M. Its rank is 20, which coincides with the dimension of the linear space of quaternary cubic forms.

Lemma 3. *Given a set of linear forms $\ell_j(x_0,\ldots,x_s) = a_{j0}x_0 + \cdots + a_{js}x_s$, where $s < j \leqslant n$ and all the coefficients a_{ji} are algebraically independent elements of a purely transcendental extension of the field of rational numbers of transcendence degree $(n-s)(s+1)$, if $(s+3)(s+2) \leqslant 6(n-s)$, then each cubic form in $x_0, \ldots, x_s$ is a linear combination of forms of the type $\ell_j(\ell_j - x_0)x_t$, where $s < j \leqslant n$ and $0 \leqslant t \leqslant s$.*

Proof. By analogy with the proof of Lemma 2, we show that the matrix N consisting of the coefficients of forms of the type $\ell_j(\ell_j - x_0)x_t$ is of full rank. The matrix N is the sum of the matrix M consisting of the coefficients of forms of the type $\ell_j^2 x_t$ and the matrix B consisting of the coefficients of forms of the type $-x_0\ell_j x_t$. However, the entries of B treated as polynomials in a_{ji} have a lower degree than the entries of M. Therefore, the rank of the matrix $N = M + B$ is at least the rank of M. By Lemma 2, the matrix M is of full rank. Therefore, the matrix N is also of full rank. The lemma is proved.

Algorithm Discussed in the Proofs of Theorems 2 and 3

Input: integers $0 < m < n$ and linear forms $\ell_j(x_0,\ldots,x_{n-m})$, where $n - m < j \leqslant n$.

1: The entries of the matrix A are the coefficients of the form

$$\sum_{t=0}^{n-m}\left(\sum_{k=1}^{n-m}\lambda_{tk}x_k(x_k - x_0) + \sum_{j=n-m+1}^{n}\lambda_{tj}\ell_j(\ell_j - x_0)\right)x_t,$$

where the rows of A correspond to monomials of the third degree in $x_0, \ldots, x_{n-m}$ and the columns of A correspond to the variables λ_{tk} and λ_{tj}.

2: The augmented matrix B is obtained from A by adding a column with 1 in the row corresponding to the monomial x_0^3 and with the other entries being zero.

3: **if** $\operatorname{rank}(A) = \operatorname{rank}(B)$, **then** the input is rejected,

4: **else** the algorithm announces that the choice is uncertain.

3.3. Algorithm and Complexity Bounds

Speaking about recognition problems, we propose three possible answers: an input can be accepted or rejected and, additionally, a message is also possible that the choice is uncertain. The answer has to be obtained in finite time without errors, and an uncertainty message can be returned only for a small fraction of inputs (see [12, 15]).

By the computational complexity, we mean the algebraic complexity, which is equal to the number of arithmetic operations with numbers and comparisons of numbers (see [29]). The complexity of executing an individual operation is not taken into account.

Theorem 2. *There exists a recognition algorithm with three possible answers and with its input consisting of positive integers n and $m < n$ and a system of m linear forms $\ell_j(x_0,\ldots,x_{n-m})$, where $n - m < j \leqslant n$, that satisfies the following conditions if $(n-m+3)(n-m+2) \leqslant 6m$:*

- *The algebraic complexity of the algorithm is bounded above by a polynomial in n.*
- *If the input is rejected, then the system of equations $x_j = \ell_j(1, x_1,\ldots,x_{n-m})$ for indices $n - m < j \leqslant n$ has no $\{0,1\}$-solution.*
- *If the input is accepted, then the system of equations $x_j = \ell_j(1, x_1,\ldots,x_{n-m})$ for indices $n - m < j \leqslant n$ has a $\{0,1\}$-solution.*

• *For any admissible set of values n and m, there exists an identically nonzero polynomial of degree at most* $n^2(n-m+1)^2$ *in the coefficients of all linear forms* ℓ_j *such that if for some input the algorithm announces that the choice is uncertain, then this polynomial vanishes for the corresponding set of coefficient values.*

Proof. In fact, the algorithm described above checks whether there exist sets of numbers λ_{tk} and λ_{tj} for which the following two cubic forms are equal to each other:

$$\sum_{t=0}^{n-m}\left(\sum_{k=1}^{n-m}\lambda_{tk}x_k(x_k-x_0)+\sum_{j=n-m+1}^{n}\lambda_{tj}\ell_j(\ell_j-x_0)\right)x_t=x_0^3.$$

Here, $0\leqslant t\leqslant n-m$. This check is reduced to solving a system of linear equations for λ_{tk} and λ_{tj}. The total number of variables in the system is at most $n(n-m+1)$. Each equation in the system corresponds to the current monomial of the third degree in the variables $x_0, \ldots, x_{n-m}$. Therefore, the number of these equations is equal to $(n-m+1)(n-m+2)(n-m+3)/6$. Under the inequality from the condition of the theorem, the number of equations in this system does not exceed the number of variables λ_{tk} and λ_{tj}. By the Kronecker–Capelli theorem, a solution exists if the ranks of the two matrices coincide. The augmented matrix is obtained by adding a column with all entries being zero, except for a single one. The rank of the matrix is easy to calculate. Entries of these matrices are polynomials of degree at most 2 in the coefficients of the linear forms ℓ_j. A sufficient condition for the existence of a solution is that the minor equal to a polynomial of degree at most $n^2(n-m+1)^2$ in the coefficients of the linear forms ℓ_j is nonzero. By Lemma 3, this polynomial does not vanish identically. The theorem is proved.

If the algorithm from Theorem 2 rejects a system of equations S, then it also rejects any system of equations S' obtained by adding new equations to S.

Comparing Theorems 1 and 2 shows that the condition in Theorem 2 is nonoptimal for $n=2$. In particular, this is associated with the rough estimate of the matrix rank in the proof of Lemma 3.

Theorem 3. *There exists a recognition algorithm with three possible answers and with its input consisting of positive integers n* and $m<n$ *and a system of m linear forms* $\ell_j(x_0,\ldots,x_{n-m})$, *where* $n-m<j\leqslant n$, *that satisfies the following conditions if* $(n-m+3)(n-m+2)\leqslant 6m$:

• *The algebraic complexity of the algorithm is bounded above by a polynomial in n.*

• *If the input is rejected, then the system of equations* $x_j=\ell_j(1,x_1,\ldots,x_{n-m})$ *for indices* $n-m<j\leqslant n$ *has no* $\{0,1\}$*-solution.*

• *If the input is accepted, then the system of equations* $x_j=\ell_j(1,x_1,\ldots,x_{n-m})$ *for indices* $n-m<j\leqslant n$ *has a* $\{0,1\}$*-solution.*

• *For any rational number* ε *from the interval* $0<\varepsilon<1$ *and for any admissible values of n* and *m, if all the coefficients of the forms* ℓ_j *are independently and uniformly distributed on a set of cardinality at least* $(1/\varepsilon)n^2(n-m+1)^2$, *then the probability of returning an uncertainty announcement is at most* ε.

Proof. As in the proof of Theorem 2, we apply the algorithm presented in this paper. There exists an identically nonzero polynomial of degree at most $n^2(n-m+1)^2$ in the coefficients of all the linear forms ℓ_j that vanishes on sets for which an uncertainty announcement is returned. By the Schwartz–Zippel lemma (Lemma 1), the probability that this polynomial vanishes is at most ε. The theorem is proved.

4. CONCLUSIONS

The inequality from Theorems 2 and 3 has the form $m\geqslant n-\sqrt{6n-o(n)}$. In the algorithm published in [15], which makes use of quadratic, rather than cubic forms, the counterpart inequality has the form $m\geqslant n-\sqrt{2n-o(n)}$. Therefore, the new algorithm is applicable under weaker constraints. On the other hand, since the algorithm is based on the computation of a matrix rank, it can be efficiently implemented on multiprocessor computers.

The proposed algorithm fails to ensure a polynomial worst-case computational complexity. However, for many cases, it is much more efficient than exhaustive search. If the coefficients of the linear terms of the equations are positive, then the algorithm proposed by Kuzyurin [13, 14] is more efficient on average. However, the new algorithm can be more efficient in the case of oppositely signed coefficients.

There are various methods for reducing a given system of linear equations to another one having the same number of {0,1}-solutions. This opens up additional opportunities to overcome the uncertainty of the answer. On the other hand, although we considered only the solution existence problem, binary search can be used to find a {0,1}-solution, if any. Specifically, the proof that there is no {0,1}-solution corresponding to the vertex of some facet of the unit cube makes it possible to reduce the number of variables in the original problem. The same is true if the facet is replaced by a pair of symmetrically arranged faces of codimension two lying in a single hyperplane. Following this approach, a combination of different methods can be used at different steps.

The algorithm considered in this paper never accepts its input. However, the number of rejections can be reduced by applying other well-known methods, including Kuzyurin's algorithm. After this, the new algorithm can accept some inputs.

A well-known closely related problem is the NP-complete 3-CNF satisfiability, for which heuristic algorithms and their software implementations are available. Although this problem is also reduced to the subset sum one in polynomial time, the new algorithm probably does not provide any gain for its practically applicable input sizes. However, the new algorithm is applicable to a wider set of its instances. This increases interest in generalizations of the subset sum problem, together with the 3-CNF satisfiability problem.

ACKNOWLEDGMENTS

The author is grateful to M.D. Malykh and the anonymous reviewer for helpful comments.

CONFLICT OF INTEREST

The author declares that he has no conflicts of interest.

REFERENCES

1. E. Horowitz and S. Sahni, "Computing partitions with applications to the knapsack problem," J. ACM **21** (2), 277–292 (1974). https://doi.org/10.1145/321812.321823
2. K. Meer, "A note on a $P \neq NP$ result for a restricted class of real machines," J. Complexity **8** (4), 451–453 (1992). https://doi.org/10.1016/0885-064X(92)90007-X
3. P. Koiran, "Computing over the reals with addition and order," Theor. Comput. Sci. **133** (1), 35–47 (1994). https://doi.org/10.1016/0304-3975(93)00063-B
4. F. Cucker and M. Matamala, "On digital nondeterminism," Math. Syst. Theory **29**, 635–647 (1996). https://doi.org/10.1007/BF01301968
5. D. Grigoriev, "Complexity of Positivstellensatz proofs for the knapsack," Comput. Complexity **10**, 139–154 (2001). https://doi.org/10.1007/s00037-001-8192-0
6. S. Margulies, S. Onn, and D. V. Pasechnik, "On the complexity of Hilbert refutations for partition," J. Symb. Comput. **66**, 70–83 (2015). https://doi.org/10.1016/j.jsc.2013.06.005
7. K. Koiliaris and C. Xu, "Faster pseudopolynomial time algorithms for subset sum," ACM Trans. Algorithms **15** (3), 40 (2019). https://doi.org/10.1145/3329863
8. A. Polak, L. Rohwedder, and K. Wegrzycki, "Knapsack and subset sum with small items," in *48th International Colloquium on Automata, Languages, and Programming* (*ICALP 2021*), Ed. by N. Bansal, E. Merelli, and J. Worrell, Dagstuhl, Leibniz Int. Proc. Inf. **198** (106), 1–19 (2021). https://doi.org/10.4230/LIPIcs.ICALP.2021.106
9. J. C. Lagarias and A. M. Odlyzko, "Solving low-density subset sum problems," J. ACM **32** (1), 229–246 (1985). https://doi.org/10.1145/2455.2461
10. M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C. P. Schnorr, and J. Stern, "Improved low-density subset sum algorithms," Comput. Complexity **2** (2), 111–128 (1992). https://doi.org/10.1007/BF01201999
11. A. May, "Solving subset sum with small space—Handling cryptanalytic Big Data," Inf. Technol. **62** (3–4), 181–187 (2020). https://doi.org/10.1515/itit-2019-0038

12. A. N. Rybalov, "On generic complexity of the subset sum problem for semigroups of integer matrices," Prikl. Diskret. Mat., No. 50, 118–126 (2020). http://mi.mathnet.ru/eng/pdm/y2020/i4/p118
13. N. N. Kuzyurin, "An integer linear programming algorithm that is polynomial in the average case," Sib. Zh. Issled. Oper. **1** (3), 38–48 (1994).
14. N. N. Kuzyurin, "An integer linear programming algorithm polynomial in the average case," in *Discrete Analysis and Operations Research: Mathematics and Its Applications*, Ed. by A. D. Korshunov (Springer, Dordrecht, 1996), Vol. 355, pp. 143–152. https://doi.org/10.1007/978-94-009-1606-7
15. A. V. Seliverstov, "Binary solutions to large systems of linear equations," Prikl. Diskret. Mat., No. 52, 5–15 (2021). http://mi.mathnet.ru/eng/pdm/y2021/i2/p5
16. Y. Pan and F. Zhang, "Solving low-density multiple subset sum problems with SVP oracle," J. Syst. Sci. Complexity **29**, 228–242 (2016). https://doi.org/10.1007/s11424-015-3324-9
17. A. V. Seliverstov, "On binary solutions to systems of equations," Prikl. Diskret. Mat., No. 45, 26–32 (2019). http://mi.mathnet.ru/eng/pdm/y2019/i3/p26
18. J. P. Martins and B. C. Ribas, "A randomized heuristic repair for the multidimensional knapsack problem," Optim. Lett. **15**, 337–355 (2021). https://doi.org/10.1007/s11590-020-01611-1
19. V. Cacchiani, M. Iori, A. Locatelli, and S. Martello, "Knapsack problems—An overview of recent advances: Part II. Multiple, multidimensional, and quadratic knapsack problems," Comput. Oper. Res. **143**, 105693 (2022). https://doi.org/10.1016/j.cor.2021.105693
20. D. V. Gribanov and N. Yu. Zolotykh, "On lattice point counting in Δ-modular polyhedra," Optim. Lett. **16**, 1991–2018 (2022). https://doi.org/10.1007/s11590-021-01744-x
21. S. Al-Shihabi, "A novel core-based optimization framework for binary integer programs—the multidemand multidimensional knapsack problem as a test problem," Oper. Res. Perspect. **8**, 100182 (2021). https://doi.org/10.1016/j.orp.2021.100182
22. A. V. Lotov and A. I. Ryabikov, "Extended launch pad method for the Pareto frontier approximation in multiextremal multiobjective optimization problems," Comput. Math. Math. Phys. **61** (10), 1700–1710 (2021). https://doi.org/10.1134/S0965542521100080
23. V. G. Zhadan, "Primal–dual Newton method with steepest descent for the linear semidefinite programming problem: Newton's system of equations," Comput. Math. Math. Phys. **62** (2), 232–247 (2022). https://doi.org/10.1134/S0965542522020129
24. H. Fu, Y. Xu, G. Wu, J. Liu, S. Chen, and X. He, "Emphasis on the flipping variable: Towards effective local search for hard random satisfiability," Inf. Sci. **566**, 118–139 (2021). https://doi.org/10.1016/j.ins.2021.03.009
25. H. Fu, J. Liu, G. Wu, Y. Xu, and G. Sutcliffe, "Improving probability selection based weights for satisfiability problems," Knowl.-Based Syst. **245**, 108572 (2022). https://doi.org/10.1016/j.knosys.2022.108572
26. P. Guo and Y. Zhang, "ISSATA: An algorithm for solving the 3-satisfiability problem based on improved strategy," Appl. Intell. **52**, 1740–1751 (2022). https://doi.org/10.1007/s10489-021-02493-1
27. S. Cai and Z. Lei, "Old techniques in new ways: Clause weighting, unit propagation, and hybridization for maximum satisfiability," Artif. Intell. **287**, 103354 (2020). https://doi.org/10.1016/j.artint.2020.103354
28. W. Li, C. Xu, Y. Yang, J. Chen, and J. Wang, "A refined branching algorithm for the maximum satisfiability problem," Algorithmica **84**, 982–1006 (2022). https://doi.org/10.1007/s00453-022-00938-8
29. S. A. Abramov, *Lectures on Complexity of Algorithms* (Mosk. Tsentr Neprer. Mat. Obrazovan., Moscow, 2010) [in Russian].
30. P. E. Alaev and V. L. Selivanov, "Fields of algebraic numbers computable in polynomial time I," Algebra Logic **58** (6), 447–469 (2020). https://doi.org/10.1007/s10469-020-09565-0
31. A. B. Batkhin, "Parameterization of the discriminant set of a polynomial," Program. Comput. Software **42** (2), 65–76 (2016). https://doi.org/10.1134/S0361768816020031
32. A. V. Seliverstov, "Heuristic algorithms for recognition of some cubic hypersurfaces," Program. Comput. Software **47** (1), 50–55 (2021). https://doi.org/10.1134/S0361768821010096

33. J. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," J. ACM **27** (4), 701–717 (1980).
https://doi.org/10.1145/322217.322225
34. L. Halbeisen, N. Hungerbühler, and S. Schumacher, "Magic sets for polynomials of degree *n*," Linear Algebra Appl. **609**, 413–441 (2021).
https://doi.org/10.1016/j.laa.2020.09.026
35. A. L. Chistov, "Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic," in *Fundamentals of Computation Theory FCT'85*, Ed. by L. Budach, *Lecture Notes in Computer Science* (Springer-Verlag, Berlin, 1985), Vol. 199, pp. 63–69.
https://doi.org/10.1007/BFb0028792
36. K. Mulmuley, "A fast parallel algorithm to compute the rank of a matrix over an arbitrary field," Combinatorica **7** (1), 101–104 (1987).
https://doi.org/10.1007/BF02579205
37. O. N. Pereslavtseva, "Calculation of the characteristic polynomial of a matrix," Discrete Math. Appl. **21** (1), 109–129 (2011).
https://doi.org/10.1515/dma.2011.008
38. H. Y. Cheung, T. C. Kwok, and L. C. Lau, "Fast matrix rank algorithms and applications," J. ACM **60** (5), 31 (2013).
https://doi.org/10.1145/2528404
39. G. I. Malaschonok and A. V. Seliverstov, "Calculation of integrals in MathPartner," Discrete Continuous Model. Appl. Comput. Sci. **29** (4), 337–346 (2021).
https://doi.org/10.22363/2658-4670-2021-29-4-337-346
40. A. V. Seliverstov and V. A. Lyubetsky, "About forms equal to zero at each vertex of a cube," J. Commun. Tech. Electron. **57** (8), 892–895 (2012).
https://doi.org/10.1134/S1064226912080049

Translated by I. Ruzanova