Generic-Case Complexity of the Multiple Subset Sum Problem

Alexandr Seliverstov

Abstract. We consider generic-case complexity of the multidimensional subset sum problem. Several heuristic algorithms have been known. So, in 1994, Nikolai Kuzyurin published such an algorithm. Nevertheless, the more methods are known, the more opportunities exist for solving certain problems. We propose a sub-exponential algorithm to verify that there is no binary solution to a general system of sufficiently many linear equations with integer coefficients. Roughly speaking, the algorithm checks whether there exists a low-degree algebraic hypersurface passing through each point with binary coordinates but not intersecting the given affine subspace.

Introduction

The subset sum problem is NP-complete. A commonly held view was that its worst-case complexity cannot be sub-exponential [1, 2]. Moreover, if we restrict our computations by so-called linear machines, then the problem is proved hard [3].

Generic-case complexity of a decision problem is sub-exponential when the set of hard inputs is negligible (or empty), but almost all inputs can be solved in sub-exponential time. Moreover, the negligible set containing hard inputs can be discerned explicitly. Such algorithms are also known as deterministic errorless heuristics. An example of fast generic-case algorithm is the condensation method for computing determinants [4]. For general matrices, the method is very nice. Nevertheless, if some intermediate matrix contains a zero entry, then the algorithm can fails.

By means of variable elimination, searching for a $\{0,1\}$ solution to a system of m linearly independent linear equations in n variables is reduced to a parallel check whether a binary solution to a subsystem in $n-m$ variables can be extended to a $\{0,1\}$ solution to the whole system of equations in n variables. Hence, the initial problem is polynomial-time solvable when the difference between the number of variables and the number of linearly independent equations is bounded by a function of the type $n - m = O(\log_2 n)$. In this work, we consider generic-case complexity when both m and difference $n - m$ are sufficiently large.

The Kuzyurin Algorithm

Let us denote by A a $m \times n$ matrix with nonnegative entries and by $\mathbf{b}$ a column. One can enumerate all $\{0, 1\}$ solutions to the system of inequalities $A\mathbf{x} \leq \mathbf{b}$ using dynamic programming. If $m > 9 \log_2 n$ and some assumptions about the distribution of the entry values hold, then the average number of $\{0, 1\}$ solutions is polynomially bounded. Therefore, all solutions can be found in average polynomial time [5]. The proof is based on the tail bounds of the binomial distribution. Next, one can verify whether a $\{0, 1\}$ solution to the system of equations $A\mathbf{x} = \mathbf{b}$ exists. The crucial limitation on the applicability of the Kuzyurin algorithm is the requirement of nonnegativity of the matrix entries. Of course, any system of linear equations can be reduced to another system with nonnegative coefficients, but the distribution is warped.

Low-density Problems

Let the density of an instance of the subset sum problem with positive integer coefficients a_k be defined by $\frac{n}{\log_2 \max_k a_k}$. A polynomial-time algorithm is known for solving almost all instances of sufficiently low density using a subroutine for finding the shortest nonzero vector in a lattice [6, 7]. The multiple low-density problems are considered too [8].

1. Our Main Algorithm

Within the context of the generic-case complexity, we consider machines having three halting states. So, the machine not only rejects or accepts an input, but it can also halt in the vague halting state. The latter means denial of response. But such a failure is possible only on a small fraction of inputs.

Theorem 1. *There exist both constant c and machine with the vague halting state such that for all positive integers $d \geq 2$, n, and $m < n$ satisfying the inequality $(n - m + d)(n - m + d - 1) \leq md(d - 1)$ and for every m-tuple of linear forms $\ell_j(x_0, \ldots, x_{n-m})$, where $n - m < j \leq n$, the machine either rejects the input or halts in the vague halting state in $O(n^{cd})$ arithmetic operations. If the machine rejects the input, then there is no $\{0, 1\}$ solution to the system of all inhomogeneous equations of the type $x_j = \ell_j(1, x_1, \ldots, x_{n-m})$. Moreover, for every applicable integers d, n, and m, there exists a nonzero polynomial of degree at most $n^2(n - m + 1)^{2d-4}$ in coefficients of all the linear forms ℓ_j such that if the machine halts in the vague halting state, then the polynomial vanishes.*

Proof. The algorithm (Fig. 1) verifies whether there exists a solution to a system of linear equations in at most $n(n - m + 1)^{d-2}$ unknowns λ_{tk} and λ_{tj}. The sufficient condition for the solvability is the full rank of a matrix. Of course, the rank can

FIGURE 1. Checking whether the system has no $\{0,1\}$ solution.

Input: Both integer $d \geq 2$ and set of m linear forms ℓ_j in $n-m+1$ variables.
if there exist numbers λ_{ik} and λ_{ij} such that

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

where g_t is the t-th monomial of degree $d-2$ in variables $x_0, \ldots, x_{n-m}$
  then the machine **rejects** the input
  else the machine halts in the **vague** halting state.

be calculated easily. But a weaker sufficient condition is that a largest minor does not vanish. The minor is a polynomial in matrix entries. An entry is a polynomial of degree at most two in coefficients of linear forms ℓ_j. Thus, the minor is a polynomial of degree at most $n^2(n-m+1)^{2d-4}$. This polynomial does not vanish identically. □

Roughly speaking, the algorithm checks whether there exists a hypersurface passing through each $\{0,1\}$ point but not intersecting the given affine subspace. Therefore, for given d, if the algorithm rejects a subsystem, then it rejects the whole system too.

If $n - m = O(\sqrt{n})$, then one can use a constant degree d. Thus, generic-case complexity is polynomial, cf. [9].

Theorem 2. *There exist both constant c and machine with the vague halting state such that for all positive integers $n > m \geq 4 \log_2^4 n$ and for every m-tuple of linear forms $\ell_j(x_0, \ldots, x_{n-m})$, where $n - m < j \leq n$, the machine either rejects the input or halts in the vague halting state in $O\left(2^{cn/\log n}\right)$ arithmetic operations. If the machine rejects the input, then there is no $\{0,1\}$ solution to the system of all inhomogeneous equations of the type $x_j = \ell_j(1, x_1, \ldots, x_{n-m})$. Moreover, for every applicable integers n and m, if all coefficients of forms ℓ_j picked independently and uniformly at random from a set of cardinality $(1/\varepsilon)4^{\lceil n/\log_2 n \rceil}$, then the machine halts in the vague halting state with probability at most ε.*

Proof. Let us use Theorem 1 with parameter $d = \lceil n/\log_2^2 n \rceil$. There exists a nonzero polynomial of degree at most $4^{\lceil n/\log_2 n \rceil}$ in coefficients of all the linear forms ℓ_j such that if the machine halts in the vague halting state, then the polynomial vanishes. In accordance with the Schwartz–Zippel lemma, the machine halts in the vague halting state with probability at most ε. □

Conclusion

If all coefficients are nonnegative integers from a large set and picked independently at random, then the Kuzyurin algorithm has the advantage with high probability [5]. But our algorithm works over all integers. Moreover, for nonnegative coefficients, it can also give a quick answer when the Kuzyurin algorithm requires a long running time.

References

[1] D. Grigoriev, *Complexity of Positivstellensatz proofs for the knapsack*, Computational Complexity **10** (2001), 139–154. `https://doi.org/10.1007/s00037-001-8192-0`

[2] S. Margulies, S. Onn, D.V. Pasechnik, *On the complexity of Hilbert refutations for partition*, Journal of Symbolic Computation **66** (2015), 70–83. `https://doi.org/10.1016/j.jsc.2013.06.005`

[3] K. Meer, *A note on a $P \neq NP$ result for a restricted class of real machines*, Journal of Complexity **8**:4 (1992), 451–453. `https://doi.org/10.1016/0885-064X(92)90007-X`

[4] C.L. Dodgson, *Condensation of determinants, being a new and brief method for computing their arithmetical values*, Proceedings of the Royal Society of London **15** (1866), 150–155.

[5] N.N. Kuzyurin, *An integer linear programming algorithm polynomial in the average case.* In: A.D. Korshunov (eds.) Discrete Analysis and Operations Research. Mathematics and Its Applications, vol. 355, pp. 143–152. Springer, Dordrecht, 1996. `https://doi.org/10.1007/978-94-009-1606-7_11`

[6] J.C. Lagarias, A.M. Odlyzko, *Solving low-density subset sum problems*, Journal of the Association for Computing Machinery **32**:1 (1985), 229–246. `https://doi.org/10.1145/2455.2461`

[7] M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.P. Schnorr, J. Stern, *Improved low-density subset sum algorithms*, Computational Complexity **2**:2 (1992), 111–128. `https://doi.org/10.1007/BF01201999`

[8] Y. Pan, F. Zhang, *Solving low-density multiple subset sum problems with SVP oracle*, Journal of Systems Science and Complexity **29** (2016), 228–242. `https://doi.org/10.1007/s11424-015-3324-9`

[9] A.V. Seliverstov, *Binary solutions to large systems of linear equations* (in Russian), Prikladnaya Diskretnaya Matematika **52** (2021), 5–15. `https://doi.org/10.17223/20710410/52/1`

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