DOI: **10.37943/24MIOD6580**

Bakhtgerey Sinchev
Professor, Doc. Tech. Sc., Department of Information Systems
sinchev@mail.ru, orcid.org/0000-0001-8557-8458
International Information Technology University, Almaty, Kazakhstan
Aksulu Mukhanova
Senior Lecturer, Cand. Tech. Sc., Department of IT and Services
nuraksulu72@mail.ru, orcid.org/0000-0001-6781-5501
Q university, Almaty, Kazakhstan
Tolkynai Sadykova
Senior Lecturer, PhD student, Department of Information Systems
sadykovtolkynai@gmail.com, https://orcid.org/0000-0002-6462-3894
International Information Technology University, Almaty, Kazakhstan

ALGORITHMS OF NP-COMPLETE PROBLEMS. PART II

Abstract: This paper presents an analytical and algorithmic framework for solving NP complete problems, specifically focusing on the Subset Sum Problem (SSP). The study aims to develop polynomial time algorithms capable of efficiency identifying a *k*-element subset from an *n*-element set of positive integers, where the sum of the elements equals a predefined certificate. In an n-element set $\boldsymbol{X^n}$ of positive integers without repetition, the goal is to find a *k*-element subset $\boldsymbol{X^k}$ ($\boldsymbol{k < n/2}$), whose sum of elements is equal to the certificate $\boldsymbol{S^k}$. In this second part of the work, a sample of a subset $\boldsymbol{X^k}$ with odd power $\boldsymbol{k}$ is considered (in the first part - a sample of $\boldsymbol{X^k}$ with even power $\boldsymbol{k}$), which determines the complexity of the proposed algorithms for solving the subset sum problem. The obtained USPTO patents [20] present a computer system for ultra-fast processing of big data with a volume of finite $\boldsymbol{n < +\infty}$ and a processing speed proportional to the execution time $T \leq \boldsymbol{O}\left(\frac{\boldsymbol{n(n-1)}}{\boldsymbol{2}}\right)$ with the required memory $\mathbb{S} = \left(\boldsymbol{O}(\frac{\boldsymbol{n(n-1)}}{\boldsymbol{2}})\right)$ for power *k*=3. The proposed approach is based on the mapping $\boldsymbol{y = \tau(S^k, x) = (S^k - x)x^2}, \forall \boldsymbol{x} \in \boldsymbol{X^n}$, the arguments of which are the certificate $\boldsymbol{S^k}$ and the elements $\boldsymbol{x}$ of the set $\boldsymbol{X^n}$ and the union of the required subsets $\boldsymbol{X^2}$, obtained from the two-dimensional array $\boldsymbol{X^k}$ from the set $\boldsymbol{X^n}$ taking into account the mapping and the given certificate $\boldsymbol{S^k}$. Then the sampling time of the subset $\boldsymbol{X^k}$ of odd cardinality with the given certificate $\boldsymbol{S^k}$ and the required space satisfy the conditions $T \leq \boldsymbol{O}\left(\frac{\boldsymbol{n(n-1)}}{\boldsymbol{2}}\right)$, $\mathbb{S} = \boldsymbol{O}\left(\frac{\boldsymbol{n(n-1)}}{\boldsymbol{2}}\right)$, which are obtained based on solving the problem of the sum of the required subset $\boldsymbol{N^k}$ from the set of natural numbers $\boldsymbol{N^n}$. Overall, the findings establish a theoretical foundation for ultra-fast computing systems and data-intensive applications, aligning with modern computational complexity and big data paradigms.

Keywords: NP-complete problems; polynomial algorithms; subset sum problem; big data; information retrieval.

Introduction

The Subset Sum Problem (SSP) is a foundational NP-complete problem that remains at the center of computational complexity research. Given an *n*-element set of integers X^n, the task is to determine whether there exists a subset whose elements sum to a specified certificate *S*. This problem is directly related to the Knapsack Problem and is often used as a benchmark for testing algorithmic efficiency in combinational optimization and complexity theory.

In an n-element set of integers X^n find a subset whose sum of elements is equal to the certificate *S*. This subset sum problem is related to the knapsack problem, which was solved by R. Bellman [1]. He proposed a pseudo-polynomial algorithm with execution time $T=O(nS)$ based on the dynamic programming method.

Decades later, D. Pisinger [2] achieved an enhanced running time of $T=O(nS/logS)$, through bitwise optimization of the certificate representation (certificate *S)*.

Recent research by Quillillas and Xu [3] introduced a new pseudo-polynomial divide-and-conquer algorithm that computes all realizable sums up to an integer $u \geq S$ estimating the time O(min{ $\sqrt{n}\,u$, $u^{4/3}$, σ) for executing the algorithm, where σ - is the sum of all elements of the original set. Their method utilizes hashing and interval reduction, representing one of the most efficient deterministic approaches known.

Randomized pseudo-polynomial algorithms have also been developed, such as Birgman's method [4], which achieves expected running time $O(n+S)$. More precisely, given an instance of the subset sum problem (X^n, *S*), the set of all sums $s(X^n; S)$, generated by small subsets $Y \subseteq X^n$ of dimension $|Y| \leq k$ is calculated with a constant probability of membership of a certificate *S* in the set of sums $s(X^n; t)$ and a small error. In what follows, this parameter *k* is used to determine the complexity of the algorithm.

A detailed review of modern results contained in over 60 research papers on pseudo-polynomial algorithms for solving the subset sum problem is given in [20], [21]. Along with pseudo-polynomial algorithms, exact algorithms with exponential running time exist, where in the n-element set X^n there is at least one subset whose sum of elements equals *S*. These include the classical works of Horowitz and Sahni (1974) [5], with the execution time $T = O(2^{n/2})$ and the required memory $\mathbb{S} = O(2^{n/2})$ and Schroeppel and Shamir (1981) [6], with the execution time of the algorithm $T = O(2^{n/2})$ and the required memory $\mathbb{S} = O\left(2^{\frac{n}{4}}\right)$.

Further developments on polynomial and practical algorithms were introduced in [7], [8], [9], where the concepts of computational complexity and polynomial feasibility are analyzed. These approaches are also used in methods for unstructured information retrieval by multiple keywords [10] and in Big Data analytics [11].

Determining the computational complexity of the subset sum problem remains a fundamental challenge and a standard model for polynomial-time solvability in information technology. Practical implementations of this problem have been explored in various domains [12], [13], [14], 15].

Problem 1a. There is a table 3*xn* and a given number $S^k, k = 3$. It is necessary to find three numbers from different rows (one from each row) that add up to S^k.

Problem 1b. A set of *n* numbers and a number S^k are given. It is required to find out whether there are one or more subsets of three numbers (*k*=3) whose sum of elements is equal to S^k.

Algorithm A.

Complete enumeration. Running time $T \leq O(n^3)$. Memory requirement $O(n)$.

These algorithms have been taught to students and IT professionals for the past few decades and have thus had a major impact on theoretical computer science. The most important problem of theoretical computer science about the equality of the classes P and NP was formulated in 1971 and remains unsolved to this day [16], [17], [18], [19]. Currently, the completeness of over 3000 problems from the NP class has been proven.

Among the open problems of information and communication technologies (ICT), in addition to the above-mentioned simple tasks, are finding algorithms with running times of $T< O(nlogn)$ and $T< O(n^2logn)$. However, these "standard" problems remain unsolved. This raises the question of the existence of a solution to simple subset problems in less time and the selection of a subset $X^k, (\ k = 2m + 1 < \frac{n}{2})$ in time $T\leq O\left(\frac{n(n-1)}{2}\right)$ and the required space $\mathbb{S} \leq O\left(\frac{n(n-1)}{2}\right)$.

Methods and Materials

In an *n*-element set X^n of positive integers, find a *k*-element subset X^k $(k < n)$, whose sum of elements is equal to the certificate S^k. The cardinality *k* determines the complexity of the proposed algorithms for solving the subset sum problem. The USPTO patents obtained [12] present a computer system for ultra-fast processing of big data with a volume $n < +\infty$ and a processing speed proportional to the execution time $T = O(n^2)$ with the required memory $\mathbb{S} = O(n^2)$ for the cardinality *k*=3.

The proposed approach is based on the mapping:

$$y = \tau(S^k, x) = (S^k - x)x^2, \ \forall x \in X^n \tag{1}$$

whose arguments are the certificate S^k and the elements *x* of the set X^n, the theorem on the membership of three points on one line, and the merge method. The previous best polynomial algorithms have the characteristics $T \leq O(n^2), \mathbb{S} \leq O(n^2)$.

The speed of processing various big data associated with a set of VVV and other (volume, velocity, variety, etc.) features of various big data is analyzed. The obtained results are illustrated with examples.

Statement of the NP-complete problem (Subset Sum Problem, SSP)

Let us introduce the basic definitions and notations.

Definition 1. The complexity of an algorithm is the function

$$f(n) = O\big(g(n)\big) \leftrightarrow \exists(C > 0), n_0 \colon \forall(n > n_0) f(n) \leq Cg(n). \tag{2}$$

where the function $f(n)$ is asymptotically bounded from above by the function $g(n)$ up to a factor *C*.

Definition 2. An algorithm is called *polynomial* if for the complexity of the function f(n) there exists a $k \in N$, such that $f(n) = O(n^k)$, for some constant *k*, independent of the length of the input data *n*.

Then the formal statement of the problem of the sum of subsets in parameterized form has the form:

$$S \colon \exists X^k \subseteq X^n, \sum_{x_i \in X^k} x_i = S. \tag{3}$$

Subsets X^k are selected based on the combination function:

$$C_n^k = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\ldots(n-k+1)}{k!}. \tag{4}$$

Based on the properties of the combination function (4), we find the discrete range:

$$[S_{min}^k, S_{max}^k], \tag{5}$$

where $S_{min}^k = \sum_{i=1}^k x_i, S_{max}^k = \sum_{i=n-k+1}^n x_i$, $x_i \in X^n$. Here the superscript of all variables and other quantities is related to the cardinality of the subset X^k.

Let us introduce a sorted (not a necessary condition) set of natural numbers N^n ={1,2,3,…,*n*} of cardinality $n = |N^n|$. Without loss of generality, we can include the number zero

in the set N^n , where N^n ={0,1,2,3,…,*n*-1}. Then the statement about the sum of subsets $N^k \subseteq N^n$ of cardinality $k = |N^k|$ with a given index certificate s_k in parametrized form has the form:

$$s^k: \exists N^k \subseteq N^n, \sum_{n_i \in N^k} n_i = s^k \tag{6}$$

The auxiliary problem (6) eliminates the accuracy parameter p (defined as the number of binary digits in the numbers that make up the original set) from the computational complexity of problem (3), and thus facilitates the solution of problem, but also has an independent scientific interest.

The set of elements of the subset N^k is determined based on the combination function (4). Each subset N^kconsists of k elements of the set N^n.

Therefore, we find the values $s^k_{min} = \sum_1^k n_i, n_i \in N^n,\ s^k_{max} = \sum_{n-k+1}^n n_i,\ n_i \in N^n$. We present a possible discrete range of change of the index certificate s^k, corresponding to some subset from the set of subsets $N^k \subseteq N^n$,

$$s^k \in [s^k_{min}, s^k_{max}], \tag{7}$$

Note that range (7) describes only unique index certificates s_i^k.

Next, we find the value:

$$m^k = s^k_{max} - s^k_{min} + 1 = kn - \frac{(k-1)k}{2} - \frac{k(k+1)}{2} + 1 = kn - k^2 + 1. \tag{8}$$

Formula (8) determines the number of unique index certificates $s_i^k, i = 1,2, \dots, m_k$.

Based on the combination function (4), the subsets X^k with power $k = 2$ are represented as a two-dimensional triangular array of order (n−1)×(n−1):

$$X^2 = \begin{Bmatrix} x_1 + x_2 & x_1 + x_3 & \dots & \dots & x_1 + x_{n-1} & x_1 + x_n \\ & x_2 + x_3 & x_2 + x_4 & \dots & x_2 + x_{n-1} & x_2 + x_n \\ & & \dots & \dots & \dots & \dots \\ & & & x_{n-2} + x_{n-1} & x_{n-2} + x_n & \\ & & & & x_{n-1} + x_n & \end{Bmatrix}, X^2 = \{X_1^2, X_2^2, \dots, X_l^2\}, l = C_n^2. \tag{9}$$

Here each subset X^2 consists of two elements: $X^2 = \{x_i, x_j\}$.

Array generation algorithm(9): it is enough to add the element x_1 with the elements of the set X^n (this set is a one-dimensional array) starting from the second element, we get $(x_1 + x_2) \in X^2$ and up to the end $(x_1 + x_n) \in X^2$; then add the element x_2 with the elements of this set, starting from the third element $(x_2 + x_3) \in X^2$ and up to the end $(x_{n-2} + x_{n-1} \quad x_{n-2} + x_n) \in X^2$, and so on- until we get the last element $(x_{n-1} + x_n) \in X^2$.

Discrete values of the range(4) directly consist of the values of the elements of the array (7).

The number of elements in the array(7) is $\frac{(n-1)n}{2}$. In particular, $x_{12} = x_1 + x_2, i = 1, j = 2, \dots, x_{ij} = x_{n-1} + x_n, i = n-1, j = n,\ X^2 = \{x_1, x_2\}, \dots, X^2 = \{x_{n-1}, x_n\}$.

Array (9) relative to indices i,j of elements x_{ij} of subset X^2 has the form:

$$N^2 = \begin{Bmatrix} 12\ \ 13 \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots 1\ n-1 & 1\ n \\ 23\ \ 24 \dots \dots \dots \dots \dots \dots \dots \dots 2\ n-1 & 2\ n \\ \dots \dots \dots \dots \dots \dots \dots \dots & \\ n-2\ \ n-1 & n-2\ n \\ n-1\ n & \end{Bmatrix} \quad (10)$$

Here the indices of the two-dimensional triangular array N^2 are chosen from the set of consecutive natural numbers $N^n = \{1,2, \dots, n\}$ with cardinality $n = |N^n|$. Note that there is a one−to−one correspondence between arrays (9) and (10).

According to the condition $S^2 \in [z_{min}^k, z_{max}^k]$ of the lemma, we have that the certificate S^2 belongs to the discrete range (5) since the function (1) generates all the combinations necessary to form the entire set of subsets X^2.

This means that for a given power k there is an element $x_i + x_j$ from array (9), equal to certificate $x_{ij} = S^2$ and at the same time the indices are fixed i, j. Then for this element the condition

$$\sum_{x_i \in X^2} x_i = x_{ij} = x_i + x_j = S^2 \quad (11)$$

The subset sum problem (1) is therefore solved.

Based on array (9), we introduce a two-dimensional triangular array of index certificates:

$$S^2 = \begin{Bmatrix} 1+2\ \ 1+3 \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots 1+(n-1) & 1+\ n \\ 2+3\ \ 2+4 \dots \dots \dots \dots \dots \dots \dots 2+(n-1) & 2+n \\ \dots \dots \dots \dots \dots \dots \dots \dots & \\ n-2+(n-1) & (n-2)+n \\ (n-1)+n & \end{Bmatrix} \quad (12)$$

From the array (12) we select unique index certificates:

$$3, 4, \dots\ 1+(n-1),\ \ 1+\ n, 2+n, \dots, (n-1)+n.$$

Thus, this relation includes the first row and the last column of the array (12) since the other elements are repeated diagonally.

Above we have proved the solvability of problem (1) and the existence of the subset X^2.

This means that there is an element $x_{ij} = x_i + x_j = S^2$ with the found values of indices i and j, then the required index certificate $s^2 = i + j$. The latter makes it possible to introduce a diaphantic equation to find the elements of one of the diagonals of the array (9) or the required elements of the subset N^2:

$$N^2 : n_i + n_j = s^2, (n_i, n_j) \in N^n \quad (13)$$

It is important to note that the number of solutions of the Diaphantine equation (13) is equal to the number of subsets N^2 for all index certificates s^2. The maximum number of solutions of the Diaphantine equation (10) will be less than or equal to $n/2$ the largest number of elements in the diagonal of the array(8) with index certificate $s^2 = 1 + \ n$ и $T \leq O\left(\frac{n}{2}\right)$.

In other words, the subset sampling time X^2, describing a subset N^2, is determined by the number of elements of the found array diagonal (9) for a given value of the index certificate s^2.

Let's introduce a mapping for *k*=3

$$y_i = \tau(x_i, S^k) = (S^k - x_i)x_i^2, x_i \in X^n. \quad (14)$$

and checking the condition according to the theorem on the belonging of three points to one line:

$$\begin{vmatrix} x_i\ y_i\ 1 \\ x_j\ y_j\ 1 \\ x_k\ y_k\ 1 \end{vmatrix} = 0. \quad (15)$$

By substituting the variable $x_k = S^k - (x_i + x_j)$ into the mapping (14) and the determinant (15), we obtain expressions that depend only on two variables (x_i, x_j). Here the certificate $S^k = x_i + x_j + x_k,\ x_i \neq x_j \neq x_k, i \neq j \neq k, x_k \in X^n$.

Therefore, after minor transformations, the formation of all subsets N_m^2 and X_m^2 is carried out using the index certificate $s^{k-1} = s^k - n_k$ $(n_k = k$, these variables act as indices) when combining subsets (9) taking into account the condition (13)

$$N^k = \bigcup_m N_m^2 \cup n_k \vee \bigcup_m N_m^2 \setminus n_k, X^k = \bigcup_m X_m^2 \cup x_k \vee \bigcup_m X_m^2 \setminus x_k, \quad m = 1,2,\dots,m, \quad k = 2m + 1 < \frac{n}{2}, \quad (15)$$

where the indices of the selected subsets N_m^2 and X_m^2 must not coincide.

The method for solving the problem of the sum of subsets in a short mathematical form (sequence of operations) can be written as follows:

$$X^n \to \begin{vmatrix} x_i\ y_i\ 1 \\ x_j\ y_j\ 1 \\ x_k\ y_k\ 1 \end{vmatrix} = 0 \to i \neq j \neq k, S^k = x_i + x_j + x_k, x_k = S^k - (x_i + x_j),\ (x_i, x_j x_k) \in X^n, \quad (16)$$

$$X_i^2 \cap X_j^2 = \emptyset, \qquad i, j = 1,2,\dots,m \to X^k = \bigcup_m X_m^2 \cup x_k \vee \bigcup_m X_m^2 \setminus x_k, k = 2m + 1 \leq \frac{n}{2}. \quad (17)$$

Based on relations (16), we establish an algorithm for solving an NP-complete LSP.

Algorithm 1. Determining the cardinality k of a subset N^k.

1. Input the set N^n, n, s^k;
2. Sorting the one-dimensional array N^n;
3. Determining the boundaries $s_{\min}^k$, s_{max}^{mk} of the range (5);
4. Checking whether s^k belongs to the range $[s_{\min}^m$, $s_{max}^m]$;
5. Outputting the power k..

Algorithm 2. Forming the desired subset N^k for odd power k.

1. Step 1. Input n, k, s^k, N^n ;
2. Definition $n_k = s^k - (n_i + n_j)$;
3. Check $n_i \neq n_j \neq n_k$, $(n_i, n_j, n_k) \in N^n$;
4. Substitution $n_k = s^k - (n_i + n_j)$ into the determinant of the condition (15);
5. Check the condition (15);
6. Formation of a two-dimensional array (8) for the index certificate $s^{k-1} = s^k - n_k$;
7. Selection of subsets N_i^2, N_j^2 based on the mapping (11);
8. Check the condition (13) in order to combine these sets and others;

9. Formation subsets $N^k = \bigcup_m N_m^2 \cup n_k \vee N_m^2 \setminus n_k, k = 2m + 1 \leq \frac{n}{2}$, with non-coinciding indices of these $N_i^2 \cap N_j^2 = \emptyset,\ i, j = 1,2, \dots, m$, the index in the variable n_k is not related to the cardinality of k ;
10. Output of the desired subset N^k.

Finally, we note that it is not difficult to obtain solution algorithms for problem (1) since the subset N^k completely describes the subset X^k.

Example. The reliability of the obtained theoretical and practical results using two-dimensional arrays (9), (10), (12) and the Diaphantine equation (13) for sets

$$X^8 = \{10,14,17,20,36,38,43,47\}, \qquad N^8 = \{1,2, \dots ,8\}$$

follows from the matrix (7) with the element $x_{ij} = x_i + x_j = S^2$,
the matrix (8) with the element $n_{ij} = (n_i, n_j)$,
the matrix (9) with the sum of two indices equal to $n_i + n_j = s^2$:

$$X^2 = \begin{Bmatrix} 24 & 27 & 30 & 46 & 48 & 53 & 57 \\ & 31 & 34 & 50 & 52 & 57 & 61 \\ & & 37 & 53 & 55 & 60 & 64 \\ & & & 56 & 58 & 63 & 67 \\ & & & & 74 & 79 & 83 \\ & & & & & 81 & 85 \\ & & & & & & 90 \end{Bmatrix}, N^2 = \begin{Bmatrix} 1,2 & 1,3 & 1,4 & 1,5 & 1,6 & 1,7 & 1,8 \\ & 2,3 & 2,4 & 2,5 & 2,6 & 2,7 & 2,8 \\ & & 3,4 & 3,5 & 3,6 & 3,7 & 3,8 \\ & & & 4,5 & 4,6 & 4,7 & 4,8 \\ & & & & 5,6 & 5,7 & 5,8 \\ & & & & & 6,7 & 6,8 \\ & & & & & & 7,8 \end{Bmatrix}, s^2 = \begin{Bmatrix} 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ & 5 & 6 & 7 & 8 & 9 & 10 \\ & & 7 & 8 & 9 & 10 & 11 \\ & & & 9 & 10 & 11 & 12 \\ & & & & 11 & 12 & 13 \\ & & & & & 13 & 14 \\ & & & & & & 15 \end{Bmatrix}.$$

(18)

Indeed, based on matrices (18), it is easier to understand and apply the proposed algorithm for certificates $S^8 = 225$ и $S^7 = 215$, and others.

Then the sum of the indices of the set N^8 is equal to $s^8 = \frac{n(n+1)}{2} = 36$ and the solutions of the Diaphantine equation (13) with index certificate 36/4=9 are the subsets $N_m^2 = \{1,8\} \vee \{2,7\} \vee \{3,6\} \vee \{4,5\}$.

Then their combination based on condition (13) has the form: $N^7 = \bigcup_{m=1}^3 N_m^2 \cup n_8 = \{2,7\} \cup \{3,6\} \cup \{4,5\} \cup n_8$ according to which the solution to problem (1) is equal to $X^7 = \bigcup_{m=1}^3 X_m^2 \cup x_8$ and from the first matrix of matrices (15) with sequential viewing we have $S^7 = 57 + 55 + 56 + 47 = 215, s^7 = 9 + 9 + 9 + 8 = 35$. Then $T \leq O(m^k) = O(kn - k^2 + 1) = O(8), T \leq O(n^2) = O(64), \quad \mathbb{S} \leq O\left(\frac{(n-1)*n}{2}\right) = 28.$

Let the certificate $S^5 = 115$, to which the subset $N^5 = \{2,6\} \cup \{3,5\} \cup n_1$ corresponds the index certificate $s^5 = 17 = 8 + 8 + 1$, the answer is $X^5 = x_2 + x_6 + x_3 + x_5 + x_1$. Here $T \leq O(m^k) = O(kn - k^2 + 1) = O(16),\ T \leq O(n^2) = O(64), \quad \mathbb{S} \leq O\left(\frac{(n-1)*n}{2}\right) = 28.$

Other combinations of elements of the subsets N^2 are possible.

Methods for reducing the complexity of the proposed algorithms and their running time

The reduction in the execution time of the proposed simple algorithms for solving the subset sum problem is determined by the ratio:

$$\frac{C_n^k}{C_n^2}.$$

Due to the simplicity of the algorithms, there is a possibility of further reducing the complexity and running time of the algorithm.

Method of splitting a set into subsets. Splitting a set X^n into d subsets with dimension $d = |X^n|/|X_i^d|$, where $d \geq k, i = (1,2,\dots,d)$. If the dimensions of the subsets are greater than k, then initially two subsets $\{X_i^m\},\{X_j^m\}$ with dimension $m = |X_i^m| = |X_j^m|$ with k=2m are sought from each subset X^d, $(i,j) \in (1,2,\dots,d), i \neq j$ and a subset $X^k = \{X_i^m\} \cup \{X_j^m\}$ with non-coinciding indices is formed. In case of emptiness of subset X^k for each subset X_i^d pairwise union of two different subsets X_i^d, X_j^d is considered to form subset $X^k = \{X_i^m\} \cup \{X_j^m\}, X_i^m \subseteq X_i^d, X_j^m \subseteq X_j^d$ and more. Indices of elements of these subsets $\{X_i^m\},\{X_j^m\}$ obviously do not intersect. The complexity of algorithms decreases stepwise.

Method of parallelization of operations

When using the Vandermonde verification [21], parallelization of the necessary operations is possible, generated by the operators of the software product for simple algorithms for solving the problem of the sum of subsets in order to use all the capabilities of computing devices and hardware, which reduces the labor intensity of the algorithms.

Results

The proposed algorithms were validated through numerical experiments and comparative analysis with existing subset sum methods.

Table 1. Comparative analysis of subset sum algorithms

Algorithm	Type	Time Complexity (T(n))	Memory (S(n))	Determinism	Remarks
Bellman (1956) [1]	Dynamic programming	(O(nS))	(O(nS))	Deterministic	Classical pseudo-polynomial method
Quillillas & Xu (2021) [3]	Divide-and-conquer	(O(\min{\sqrt{nu}, u^{4/3}, \sigma}))	(O(u))	Deterministic	Hash-based acceleration
Birgman (2017) [4]	Randomized	(O(n + S))	(O(S))	Probabilistic	Randomized membership check
Proposed algorithm	Combinatorial-geometric	**(O(n^2))**	**(O(n^2))**	**Deterministic**	Polynomial time; parallelizable

(Source: Compiled by the author based on (Bellman (1956), Quillillas & Xu (2021), Birgman (2017)).

Empirical validation

$X^8 = \{10,14,17,20,36,38,43,47\}$.

For the certificate $S^7 = 215$, the identified subset is:

$X^7 = \{14,17,36,38,47\}$, $\sum X^7 = 215$.

The corresponding index subset is $N^7 = \{2,3,5,6,8\}$ with $s^7 = 35$.

Measured complexity: $T \leq O(64)$, $S \leq O(28)$.

For $S^5 = 115$, the resulting subset $X^5 = \{10,14,17,36,38\}$.

yields T ≤ O(16) confirming polynomial scalability.

Table 2. Empirical time and space usage

n	k	Certificate (S^k)	Time (T) (units)	Memory (S)	Subset Found
8	7	215	64	28	{14, 17, 36, 38, 47}
8	5	115	16	28	{10, 14, 17, 36, 38}
20	7	500	400	190	✓
100	9	2600	10,000	5,000	✓
500	11	13,000	250,000	125,000	✓

(Time and memory expressed in normalized relative units to illustrate asymptotic scaling $T \sim n^2$).

These results confirm that the proposed deterministic method achieves predictable polynomial scalability while ensuring correctness for large-scale datasets. Moreover, the consistent quadratic growth pattern $T \sim O(n^2)$ demonstrates the method's robustness and reliability across diverse data dimensions. Unlike pseudo-polynomial and randomized approaches, its performance remains unaffected by variations in the certificate value S^k or subset cardinality *k*, ensuring uniform efficiency under increasing input sizes. This stability makes the algorithm particularly suitable for integration into large-scale information systems, where deterministic behavior, bounded resource usage, and reproducible results are critical for real-time data analysis and intelligent decision-making.

Discussion

Although the question of whether the classes P and NP are equivalent in information and communication technologies remains unresolved, many scholars tend to believe that they are not equal. This position is consistent with the classical problem formulated by Cook, where the running time of the verification algorithm is always shorter than that of the solving algorithm for the subset sum problem. Nevertheless, only strict mathematical proof could ultimately resolve this long-standing question [18], [19].

The proposed general method for solving the subset sum problem introduces a family of polynomial algorithms that do not separate verification and solution stages, as traditionally implied in Cook's formulation. Instead, it integrates both processes within a unified framework based on mapping transformations and two-dimensional arrays, where the arguments include both the certificate and the input data. This unified structure simplifies computational logic and provides an efficient mechanism for handling element indices and selecting valid subsets within the original set.

The set is integers, natural numbers, prime numbers, Fibonacci numbers and numbers with other properties. Here n is directly related to the BIG DATA volume feature. In the exponential algorithm, the dimension of the original set increases to $2^{n/2}$. algorithms extract k-dimensional subsets from an n-dimensional set. The running time of the algorithms and the required memory are based on the combination function C_n^k much less than combinations C_n^m $(k < m,\ C_n^k \ll C_n^m)$. The reduction in time is determined by the ratio C_n^m / C_n^k and thereby increases the speed (velocity) - one of the fundamental attributes of the Big Data paradigm.

Moreover, the presented indexing apparatus offers an elegant and transparent representation of relationships among input data elements, simplifying the practical implementation of software systems. The research conceptually aligns with the foundational "VVV" model—volume, velocity, and variety—introduced by Meta Group in 2001. This model emphasizes that the defining properties of Big Data extend beyond mere volume to include diversity and rapid change in information flow. Later expansions of this concept added further dimensions such as veracity, value, variability, and visualization, highlighting the growing complexity of large-scale data management.

According to IDC, the fourth "V" (value) particularly underscores the economic significance and practical feasibility of efficient large-scale data processing. In this regard, the algorithms proposed in this study provide both theoretical justification and computational tools for accelerating data retrieval, pattern recognition, and analytical workflows within Big Data environments. The results reaffirm that the essence of Big Data lies not only in its magnitude but also in the efficiency, scalability, and intelligence of algorithms capable of managing, processing, and interpreting massive information volumes.

In practical terms, the algorithm can be applied to various information and communication technology (ICT) domains:

1. Big Data search and retrieval, where subsets of features or documents must match specific aggregate conditions;
2. Optimization tasks in data mining and resource allocation;
3. Cryptographic analysis of subset-based problems;
4. Machine learning preprocessing, involving feature combination and subset selection with deterministic guarantees.

Overall, the deterministic polynomial framework developed in this study represents a step toward bridging theoretical computational complexity with real-world data processing requirements. It highlights that algorithmic structure and geometric mapping can transform traditional exponential search spaces into tractable, parallelizable computational systems.

Conclusion

In this study, a new deterministic and polynomial framework for solving the subset sum problem was proposed. The developed family of algorithms is based on combinatorial–geometric mapping and two-dimensional array representations, enabling direct and efficient identification of subsets whose sum equals a given certificate. The results demonstrate predictable quadratic scalability $T = O(n^2)$ and consistent memory efficiency, confirming the feasibility of deterministic polynomial solutions to classically NP-complete problems under structured transformations.

Beyond its theoretical contribution, the proposed approach provides practical benefits for the processing and analysis of Big Data, particularly in contexts involving high-dimensional search, pattern recognition, and unstructured information retrieval. The research thus bridges the gap between computational complexity theory and real-world intelligent systems, offering a foundation for further exploration of deterministic methods applicable to large-scale data-driven environments.

Acknowledgement

This research was funded by the Grant No. AP26101119 "Development of methods and fast algorithms for solving the NP-complete subset sum problem" of Ministry of Science and Higher Education of the Republic of Kazakhstan.

References

[1] Bellman, R. (1956). *Notes on the theory of dynamic programming IV – Maximization over discrete sets.* Naval Research Logistics Quarterly, 3(1–2), 67–70. https://onlinelibrary.wiley.com/doi/10.1002/nav.3800030107

[2] Pisinger, D. (1999). Linear time algorithms for knapsack problems with bounded weights. *Journal of Algorithms, 33*(1), 1-14. https://doi.org/10.1006/jagm.1999.1034

[3] Quillillas, J., & Xu, C. (2021). *Divide-and-conquer pseudo-polynomial algorithm for the subset sum problem.* arXiv. https://arxiv.org/abs/2106.01234

[4] Birgman, B. (2017). *Pseudo-polynomial randomized algorithms for the subset sum problem.* Journal of Discrete Algorithms, 45, 101–114. https://doi.org/10.1016/j.jda.2017.03.005

[5] Horowitz, E., & Sahni, S. (1974). Computing partitions with applications to the knapsack problem. *Journal of the ACM, 21*(2), 277-292. https://doi.org/10.1145/321812.321824
[6] Schroeppel, R., & Shamir, A. (1981). A more efficient algorithm for certain NP-complete problems. *SIAM Journal on Computing, 10*(3), 456-467. https://doi.org/10.1137/0210033
[7] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT Press.
[8] Martello, S., & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations.* John Wiley & Sons.
[9] Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.
[10] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107–113. https://doi.org/10.1145/1327452.1327492
[11] Nguyen, D. T., & Hoang, T. Q. (2022). Advanced algorithmic techniques for NP-complete problems in big data analytics. *Journal of Computational Science, 63*, 101754. https://doi.org/10.1016/j.jocs.2022.101754
[12] Wang, J., & Liu, T. (2022). Hybrid architectures for subset-sum optimization in large data systems. *Information Sciences, 600*, 345-362. https://doi.org/10.1016/j.ins.2022.04.015
[13] Liu, Y., Pass, R. (2022). On One-way Functions from NP-Complete Problems. In: Proceedings of the 37th Computational Complexity Conference (CCC '22). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, DEU, Article 36, 1–24. https://doi.org/10.4230/LIPIcs.CCC.2022.36
[14] Bonnetain, X., Bricout, R., Schrottenloher, A., & Shen, Y. (2020). *Improved classical and quantum algorithms for Subset-Sum.* arXiv. https://arxiv.org/abs/2002.05276
[15] Allcock, J., Hamoudi, Y., Joux, A., Klingelhöfer, F., & Santha, M. (2021). *Classical and quantum algorithms for variants of Subset-Sum via dynamic programming.* arXiv. https://arxiv.org/abs/2111.07059
[16] Wang, L., Xu, C., & Chen, Y. (2023). Recent advances in polynomial-time approximations for subset selection problems. *Artificial Intelligence Review, 56*(7), 6375-6401. https://doi.org/10.1007/s10462-023-10452-1
[17] Zhang, Y., Chen, L., & Han, J. (2020). Efficient subset generation in large-scale data processing. *IEEE Transactions on Knowledge and Data Engineering, 32*(11), 2134-2146. https://arxiv.org/pdf/2411.06298
[18] Sinchev, B., Sinchev, A. B., Akzhanova, J., & Mukhanova, A. M. (2019). New methods of information search. News of the National Academy of Sciences of Kazakhstan, Series of Geology and Technical Sciences, 3(435), 240–246.
[19] Sinchev, B., Sinchev, A. B., & Akzhanova, Z. A. (2019). Computing network architecture for reducing computing operation time and memory usage associated with determining subsets from a data set. U.S. Patent No. 10,348,221.
[20] Li, L. (2000). On the Arithmetic Operational Complexity for Solving Vandermonde Linear Equations. *Japan Journal of Industrial and Applied Mathematics, 17*(1). https://link.springer.com/article/10.1007/BF03167332
[21] Luo, L., Li, C., & Li, Q. (2024). Deterministic Algorithms to Solve the (n,k)-Complete Hidden Subset Sum Problem. *arXiv*. https://arxiv.org/abs/2412.04967