

Article

Indexed Subset Construction: A Structured Algorithmic Framework

Bakhtgeray Sinchev¹, Askar Sinchev², Aksulu Mukhanova³, Tolkynai Sadykova^{1,*} , Anel Auezova¹ 
and Kuanysh Baimirov¹

¹ Department of Information Systems, International Information Technology University, Almaty 050000, Kazakhstan; b.sinchev@iitu.edu.kz (B.S.); a.auezova@iitu.edu.kz (A.A.); kuanysh23@gmail.com (K.B.)

² Executive Office of the President of the Republic of Kazakhstan, Astana 010000, Kazakhstan; a.sinchev@akorda.kz

³ Department of IT and Services, University “Q”, Almaty 050013, Kazakhstan; nuraksulu72@mail.ru

* Correspondence: t.sadykova@iitu.edu.kz

Abstract

This paper studies subset construction in NP-complete problems from the perspective of structured exploration of combinatorial search spaces. Classical approaches rely on exhaustive enumeration of subsets, which leads to exponential growth in time and memory requirements. To address this limitation, we introduce an indexed framework based on the correspondence between a finite set and its associated index set. Within this framework, subsets are represented as ordered index sequences, allowing subset construction to be reformulated as a constraint-guided search process over index space. Candidate subsets are characterized by numerical descriptors derived from their indices (referred to as index certificates), which guide and filter the construction process. Subset generation is further organized through admissible index intervals that restrict feasible transitions and reduce the effective search space. The framework is based on an index-based representation and structured traversal of pairwise index combinations. Computational experiments on representative instances illustrate the behavior of the indexed construction procedure and indicate its efficiency relative to classical enumeration-based methods for small and medium-sized instances. The proposed approach provides a structured perspective on combinatorial search and offers a basis for further development of algorithms based on constrained exploration of subset structures.

Keywords: combinatorial search; subset construction; indexed representation; constraint-guided search; subset sum problem; NP-complete problems



Academic Editor: Roberto Montemanni

Received: 26 March 2026

Revised: 8 May 2026

Accepted: 11 May 2026

Published: 15 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

Combinatorial problems arising in NP-complete classes are characterized by the need to construct or verify subsets of a finite set under given constraints. Since the foundational results of Cook [1] and Karp [2], such problems have been recognized as inherently challenging due to the combinatorial growth of the search space. A common feature of many NP-complete problems is that their solution can be formulated in terms of selecting subsets that satisfy structural or numerical conditions.

Classical approaches to subset construction typically rely on exhaustive or partially exhaustive enumeration of subsets, which leads to exponential growth in computational

cost as the size of the underlying set increases. This behavior is well illustrated in classical algorithms for the subset sum and related problems, including meet-in-the-middle strategies and time–space trade-offs [3,4], as well as in more recent developments aimed at improving practical and pseudopolynomial performance [5]. More generally, standard algorithmic techniques such as dynamic programming, backtracking, and branch-and-bound methods attempt to mitigate this combinatorial explosion, but still rely on exploring large portions of the power set [6].

To address these limitations, alternative representations of combinatorial structures can be introduced to organize and constrain the exploration of the search space. In this work, we consider an indexed representation of finite sets based on the one-to-one correspondence between a finite set and its associated index set. Within this representation, subsets are described through ordered index sequences, allowing subset construction to be reformulated as a structured search process over index space.

Within the proposed framework, candidate subsets are characterized by numerical descriptors derived from their indices (referred to as index certificates), which are used to guide and filter the construction process. The search is further constrained through admissible index intervals that restrict feasible transitions and organize subset generation without requiring explicit traversal of the full power set.

The framework is based on an index-based representation and structured traversal of pairwise index combinations. The contribution of this work is algorithmic. Specifically, we introduce:

- (i) an index-based representation of subsets;
- (ii) a certificate-driven characterization of candidate solutions;
- (iii) a structured subset construction procedure based on interval-restricted index combinations;
- (iv) an empirical evaluation against standard baseline methods.

From a theoretical standpoint, the proposed framework provides an additional structural perspective on subset construction in NP-complete problems. In particular, it enables comparison between different forms of subset generation, including exhaustive power-set enumeration and indexed construction procedures.

From a practical standpoint, the indexed representation enables a more organized exploration of the combinatorial search space and may reduce effective time and memory requirements in subset construction compared to direct enumeration-based approaches, especially for small and medium-sized instances. The proposed approach does not alter the worst-case exponential complexity of NP-complete problems, but provides a structured organization of the search process.

The remainder of the paper is organized as follows. Section 2 introduces the indexed representation framework and the necessary definitions. Section 3 develops the main structural properties of indexed construction and establishes the corresponding theoretical results. Section 4 presents the algorithmic construction procedure and practical examples, together with numerical experiments. Section 5 discusses the applicability and limitations of the proposed framework. Finally, Section 6 concludes the paper.

2. Indexed Representation Framework

2.1. Basic Definitions

Definition 1. *Two sets are called equicardinal if there exists a one-to-one correspondence between their elements.*

Definition 2. *Finite sets are equicardinal if and only if they contain an equal number of elements.*

Definition 3. Let X^n be a finite set of cardinality n . The set X^n is said to be equicardinal to the index set of natural numbers $N^n = \{1, 2, 3, \dots, n\}$ of cardinality $n = |N^n|$ (or $N^n = \{0, 1, 2, \dots, n-1\}$), if there exists a one-to-one correspondence between their elements. In this case the set X^n can be represented as $\{x_1, x_2, x_3, \dots\}$ or $\{x_0, x_1, x_2, \dots\}$, where $x_i \in X^n$ corresponds to the index $i \in N^n$, and there is one-to-one correspondence, so that all x_i are distinct. Here, the superscript denotes the cardinality $n = |X^n|$ of any set X^n .

Definition 4. The characteristic function of a set $X \subset U$ is the function χ_X , which equals 1 on the elements of X and 0 on the rest of U .

Definition 5. Let N be a finite set. The set of all subsets of N is called the power set and is denoted by $P(N)$. Each subset of N uniquely corresponds to its characteristic function introduced in Definition 4.

2.2. Power Set and Set Operations

The number of elements of a set can be expressed through its characteristic function as $|X| = \sum_u \chi_X(u)$. The union $A_1 \cup A_2 \cup \dots \cup A_n$ can be written as the complement of the intersection of complements of the sets A_i . In terms of characteristic functions, these yields:

$$\chi_{A_1 \cup A_2 \cup \dots \cup A_n} = 1 - (1 - \chi_{A_1}) \dots (1 - \chi_{A_n}). \quad (1)$$

Expanding the parentheses on the right side, we obtain $\sum_i \chi_{A_i} - \sum_{i < j} \chi_{A_i} \chi_{A_j} + \sum_{i < j < k} \chi_{A_i} \chi_{A_j} \chi_{A_k} - \dots$ summing the left and right sides over all elements of U we obtain the inclusion-exclusion formula:

$$|A_1 \cup A_2 \cup \dots \cup A_n| = \sum_i |A_i| - \sum_{i < j} |A_i \cap A_j| + \sum_{i < j < k} |A_i \cap A_j \cap A_k| - \dots \quad (2)$$

Thus, the power set $P(N)$ of a finite set N provides the combinatorial structure underlying subset construction. In this work, the analysis is restricted to finite sets and their associated power sets. If a finite set is equicardinal to a corresponding index set of natural numbers, its elements can be represented through their indices. Such an indexed representation allows subsets to be constructed through operations on indices rather than through direct enumeration of elements. This representation forms the basis for the indexed construction framework considered in this work.

2.3. Indexed Representation of Finite Sets

Let us introduce the combination function:

$$C_n^k = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\dots(n-k+1)}{k!} \equiv C(n, k) \quad (3)$$

and consider the family of two-element subsets $X_1, X_2, X_3, \dots, X_n$, each obtained by the combination function C_n^2 to the set X^n . Each element x_{ij} of a subset X_i consists of a pair $\{x_i, x_j\}$ so that $X_i = \{x_i\} \times X^n$, i.e., the elements of the subset X_i are pairs whose first component has a fixed index i , with $\{i\} \in N^n$.

The same construction applies to the index set N^n , since the sets X^n and N^n are equicardinal. Let $|X^n| = n$ and consider subsets $X^k \subseteq X^n$, where $1 < k \leq n/2$. This range follows from the symmetry property of the combination function, $C_n^k = C_n^{n-k}$.

This representation defines the admissible range of subset cardinalities and provides a structured way to describe subsets through their indices. In this form, subset construction can be viewed as a combinatorial process over index sets, where subsets are generated and analyzed through index combinations rather than direct enumeration of elements.

This perspective naturally leads to a reformulation of several classical combinatorial problems in terms of indexed subset construction, which is developed in the following section.

3. Structural Properties of Indexed Construction

Proposition 1. *Let X^n be a finite set of cardinality n . The union of a finite collection of its two-element subsets can be organized in a form where elements are represented through their indices.*

Proof. Consider arranging the elements of each subset X_i into a sequence $X_i = \{x_{i1}, x_{i2}, \dots\}$, where x_{ij} denotes the element of the subset X_i corresponding to the element $X_j \in X^n$. Placing these sequences one under another yields the table

$$\begin{array}{ccccccc} x_{11}, & x_{12}, & x_{13}, & x_{14}, & \dots & & \\ x_{21}, & x_{22}, & x_{23}, & x_{24}, & \dots & & \\ x_{31}, & x_{32}, & x_{33}, & x_{34}, & \dots & & \\ x_{41}, & x_{42}, & x_{43}, & x_{44}, & \dots & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \end{array} \quad (4)$$

Each row of the table corresponds to a sequence of elements obtained from a particular subset X_i .

Traversing the diagonals of this table produces the sequence:

$$x_{11}, x_{12}, x_{21}, x_{13}, x_{22}, x_{31}, x_{14}, x_{23}, x_{32}, x_{41} \dots \quad (5)$$

which enumerates elements across subsets. If subsets intersect within a diagonal, repeated elements are removed. The resulting sequence contains all distinct elements generated through the diagonal traversal.

This construction provides a systematic way to organize the generation of elements from two-element subsets and to represent them through index-based relationships. In particular, it enables the construction of subsets of X^n with cardinalities ranging from 2 to n .

This proposition establishes a connection between structured subset construction and the combinatorial organization of the power set. $\square$

Observation 1 (Classical). *Let set X^n be a finite set of cardinality n . Then the cardinality of its power set $P(X^n)$ equals 2^n , and therefore $|X^n| < |P(X^n)|$.*

Proof. The cardinality of the finite set X^n equals $|X^n| = n$. According to Definition 5, the cardinality of the power set $P(X^n)$ is given by $|P(X^n)| = \sum_{k=0}^n C_n^k = 2^n$. Since $n < 2^n$ for any natural n , it follows that $|X^n| < |P(X^n)|$.

The exponential growth of the power set $P(X^n)$ provides a structural explanation for the combinatorial complexity observed in algorithms based on exhaustive subset enumeration. Classical approaches to subset construction therefore exhibit exponential time and space behavior.

Let X^k be a k -element subset of the set X^n , and let X^{n-k} denote the complementary subset such $X^{n-k} = X^n \setminus X^k$, $X^k \cap X^{n-k} = \emptyset$. Without loss of generality, we consider $k \leq \frac{n}{2}$ since the symmetry of the combination function (3) implies $C_n^k = C_n^{n-k}$. Therefore, it is sufficient to analyze subsets whose cardinality does not exceed $\frac{n}{2}$. Within this framework, the subset selection process can be expressed in the following form:

$$S^k : \exists X^k \subseteq X^n, \sum_{x_i \in X^k} x_i = S^k \quad (6)$$

Remark 1 (Classical baseline). Let X^n be a finite set partitioned into two subsets X^k and X^{n-k} , where $k = \frac{n}{2}$. In this case, classical meet-in-the-middle approaches achieve time and space complexity on the order of $O\left((\sqrt{2})^n\right)$, as established in [3]. This result reflects the well-known time–space trade-off in subset construction problems and serves as a reference point for evaluating alternative structured approaches.

The result holds not only for $k = \frac{n}{2}$, but also for $1 \leq k < \frac{n}{2}$. since $|P(X^k)| = \sum_{i=0}^k C_k^i = 2^k$ and $|P(X^{n-k})| = \sum_{i=0}^{n-k} C_{n-k}^i = 2^{n-k}$.

The results above describe the combinatorial structure of subset construction based on power sets. To further develop a structured representation, we introduce an indexed formulation that enables the characterization of subsets through index-based parameters.

Let us define the admissible range of values of the index certificate s^k , corresponding to subsets $N^k \subseteq N^n$, of cardinality k :

$$s^k \in [s_{min}^k, s_{max}^k]. \quad (7)$$

The number of distinct index certificates within this range is given by

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

Thus, the index certificates take values in the discrete set $s_i^k \in \{s_{min}^k, s_{min}^k + 1, s_{min}^k + 2, \dots, s_{max}^k\}$.

According to Definition 3, the equicardinality of the sets X^n and N^n allows the subset selection problem to be expressed in indexed form. In particular, for a subset $N^k \subseteq N^n$, the indexed formulation takes the form:

$$s^k : \exists N^k \subseteq N^n, \sum_{n_i \in N^k} n_i = s^k, \quad (9)$$

where, in the simplest case $n_i = i$, for $i \in N^n$.

It is important to note that the index i simultaneously serves as the address of element x_i in the space X^n . This indexed formulation provides an alternative representation of the subset selection problem and removes the explicit dependence on the precision parameter present in the original formulation.

The collection of subsets $\{N^k\}$ can be generated based on the combination function C_n^k , where each subset N^k consists of k elements of the set N^n . According to Proposition 1, the structure of pairwise combinations can be represented through the Cartesian product $N^n \times N^n$, which can be arranged in the form of a symmetric matrix $n \times n$. By removing the main diagonal and the elements below it, the resulting upper-triangular structure represents all distinct two-element subsets of N^n , which can be written as:

$$N^2 = \left\{ \begin{array}{ccccccc} \{1, 2\} & \{1, 3\} & \dots & \{1, n-1\} & \{1, n\} \\ & \{2, 3\} & \{2, 4\} & \dots & \{2, n-1\} & \{2, n\} \\ & & & \dots & & \\ & & & & \{n-2, n-1\} & \{n-2, n\} \\ & & & & & \{n-1, n\} \end{array} \right\} \quad (10)$$

Based on this representation, subsets N^k with a given index certificate s^k can be constructed through combinations of pairwise disjoint N^2 . In particular, taking into

account the parity of the subset cardinality k , the construction can be expressed as $s^k = \sum_{m=1}^m s_m^2$, $N^k = \{\bigcup_m N_m^2\}$, or for odd cardinality,

$$s^k = \sum_{m=1}^m s_m^2 \pm i, N^k = \left\{ \bigcup_m N_m^2 \right\} \cup \{i\} \vee \left\{ \bigcup_m N_m^2 \right\} \setminus \{i\}, \quad i \in N^n \quad (11)$$

where the subsets N_m^2 are pairwise disjoint, and the index i does not coincide with the indices with the indices already used in the construction. The corresponding index certificates s^2 are determined by the structure of matrix (10).

The validity of the construction is understood in terms of both the cardinality of the resulting subset and the value of the index sum.

The choice of the parameter m depends on the value of the index certificate s^k and the target cardinality k . If the given index certificate s^k belongs to the admissible range (7), then candidate subsets (one or several) N^k can be constructed according to (11) that satisfies condition (9).

Proposition 2 (Indexed construction bound). *Let the given index certificate s^k belongs to the admissible range defined by (7) and let candidate index pairs be generated through diagonal traversal of matrix (10) and the subset formation rules (11).*

Then the generation of candidate index pairs and their organization into diagonal structures can be performed within $O(n^2)$ time and space with respect to the size of the index set.

These bounds apply specifically to the construction and traversal of pairwise index structures within the indexed representation. They do not extend to the full subset construction process, which remains exponential in the worst case for NP-complete problems. $\square$

Proof. The diagonal traversal of matrix (10) generates all two-element subsets N^2 in a structured order. Each diagonal corresponds to a fixed value of the index sum $s^2 = i + j$, and the number of elements in each diagonal is bounded by $O(n)$. Since the total number of such diagonals is also bounded by $O(n)$, the total number of candidate index pairs generated through diagonal traversal does not exceed $O(n^2)$.

Higher-cardinality candidate subsets N^k are constructed by combining pairwise disjoint N^2 according to the rules (11), within the admissible range defined by (7). This construction organizes candidate generation through index-based combinations and structured traversal of the pairwise index matrix.

Therefore, the generation and organization of pairwise index structures within the indexed representation can be performed within $O(n^2)$ time and space bounds. The structure of the diagonal traversal can be illustrated by grouping pairs according to their index sum s^2 , where each group corresponds to a diagonal of matrix (10):

$$\begin{aligned} s^2 = 3 &\rightarrow \{1, 2\}; \\ s^2 = 4 &\rightarrow \{1, 3\}; \\ s^2 = 5 &\rightarrow \{1, 4\}, \{2, 3\}; \\ s^2 = 6 &\rightarrow \{1, 5\}, \{2, 4\}; \\ &\dots\dots\dots \\ s^2 = n + 1 &\rightarrow \{1, n\}, \{2, n-1\}, \dots, \{m, m+1\}; \\ &\dots\dots\dots \\ s^2 = 2n - 4 &\rightarrow \{n-4, n\}, \{n-3, n-1\}; \\ s^2 = 2n - 3 &\rightarrow \{n-3, n\}, \{n-2, n-1\}; \\ s^2 = 2n - 2 &\rightarrow \{n-2, n\}; \\ s^2 = 2n - 1 &\rightarrow \{n-1, n\}. \end{aligned} \quad (12)$$

This representation shows that each diagonal corresponds to a fixed value of the index sum and contains all pairs of indices satisfying $i + j = s^2$.

The number of elements in each diagonal increases toward the central diagonals and then decreases symmetrically, forming a structured distribution of candidate pairs.

A similar structure is obtained for odd cardinality n with minor differences in the number of central diagonals:

$$\begin{aligned}
 s^2 = 3 &\rightarrow \{1, 2\}; \\
 s^2 = 4 &\rightarrow \{1, 3\}; \\
 s^2 = 5 &\rightarrow \{1, 4\}, \{2, 3\}; \\
 s^2 = 6 &\rightarrow \{1, 5\}, \{2, 4\}; \\
 &\dots\dots\dots \\
 s^2 = n &\rightarrow \{1, n-1\}, \{2, n-2\}, \dots, \{m, m+1\}; \\
 s^2 = n+1 &\rightarrow \{1, n\}, \{2, n-1\}, \dots, \{m, m+2\}; \\
 s^2 = n+2 &\rightarrow \{2, n\}, \{3, n-1\}, \dots, \{m+1, m+2\}; \\
 &\dots\dots\dots \\
 s^2 = 2n-4 &\rightarrow \{n-4, n\}, \{n-3, n-1\}; \\
 s^2 = 2n-3 &\rightarrow \{n-3, n\}, \{n-2, n-1\}; \\
 s^2 = 2n-2 &\rightarrow \{n-2, n\}; \\
 s^2 = 2n-1 &\rightarrow \{n-1, n\}.
 \end{aligned} \tag{13}$$

In both cases, the diagonal structure provides a systematic organization of all two-element subsets N^2 . Subsets N^k can be formed by combining pairwise disjoint subsets N^2 according to (11), within the admissible range defined by (8).

Remark 2. The indexed organization of candidate subsets provides a structured mechanism for exploring the search space. While it enables systematic generation of candidate index combinations, it does not reduce the worst-case combinatorial complexity of subset construction. $\square$

4. Algorithmic Construction of Target Subsets

The proposed approach defines a structured procedure for subset construction using indexed representations, building on earlier developments of index-based subset construction methods [7]. The complete procedure for indexed subset construction is presented in Algorithm 1.

Algorithm 1. Indexed subset construction

The algorithm is described in procedural form below.

Input:

- Finite set $X^n = \{x_1, \dots, x_n\}$;
- Target sum S ;

Output:

- Subset $X^k \subseteq X^n$, such that $\sum_{x_i \in X^k} x_i = S$

if such a subset exists.

Procedure:

Step 1. Construct the index set $N^n = \{1, 2, \dots, n\}$ corresponding to the elements X^n .

Step 2. Determine admissible subset cardinalities k and compute the corresponding bounds s_{min}^k, s_{max}^k according to (7).

Algorithm 1. *Cont.*

-
- Step 3.** Construct the set of all two-element subsets N^2 using the matrix representation (10).
- Step 4.** Group subsets N^2 by their index sums s^2 using diagonal traversal.
- Step 5.** Identify candidate index combinations whose sums are consistent with the target value S , and compute the corresponding index certificate s^k .
- Step 6.** Check whether the index certificate s^k belongs to the admissible range (7) and locate the corresponding diagonal structures.
- Step 7.** Construct candidate subsets N^k iteratively based on index combinations. Starting from the set of two-element subsets N^2 , higher-cardinality subsets are constructed by extending existing combinations. In particular, elements are added sequentially to previously constructed subsets while preserving index ordering and disjointness. This process can be interpreted as an index concatenation procedure that builds subsets of increasing cardinality k from lower-order components. The resulting subsets correspond to index certificates s^k formed as sums of lower-order certificates in accordance with (11).
- Step 8.** Map the constructed index subset N^k to the corresponding subset $X^k \subseteq X^n$.
- Step 9.** Verify the condition $\sum_{x_i \in X^k} x_i = S$.
- Step 10.** If the condition is satisfied, return the subset X^k ; otherwise, continue the construction within the admissible range. If no subset is found, report that no solution is identified within the indexed construction framework.
-

Remark 3. *The algorithm operates on indexed representations and avoids explicit enumeration of the full power set. The complexity of the construction procedure is determined by the number of candidate index pairs generated from matrix (10) and can be organized within bounds $T \leq O(n^2)$, $S \leq O(n^2)$.*

These bounds refer to the indexed construction procedure and do not constitute a general polynomial-time solution for NP-complete problems.

4.1. Illustrative Examples

For clarity of presentation, we briefly introduce two types of subset combinations used in the examples.

The basic union refers to the union of two-element subsets N^2 within the same diagonal of the indexed matrix representation, corresponding to combinations with equal index sums.

The special union refers to the union of two-element subsets N^2 with disjoint indices selected from different diagonals, allowing the construction of subsets with target index certificates.

These constructions follow directly from the indexed representation framework and the combination rules defined in (11).

Example 1. *Given the set $N^8 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ with a given even cardinality $n = 8$. Then, for visual clarity of matrix (10) with the given index certificates s^2 and applying diagonal traversal, let us write out table (12) row by row:*

$$\begin{aligned}
s^2 = 3 &\rightarrow \{1, 2\}; \\
s^2 = 4 &\rightarrow \{1, 3\}; \\
s^2 = 5 &\rightarrow \{1, 4\}, \{2, 3\}; \\
s^2 = 6 &\rightarrow \{1, 5\}, \{2, 4\}; \\
s^2 = 7 &\rightarrow \{1, 6\}, \{2, 5\}, \{3, 4\}; \\
s^2 = 8 &\rightarrow \{1, 7\}, \{2, 6\}, \{3, 5\}; \\
s^2 = 9 &\rightarrow \{1, 8\}, \{2, 7\}, \{3, 6\}, \{4, 5\}; \\
s^2 = 10 &\rightarrow \{2, 8\}, \{3, 7\}, \{4, 6\}; \\
s^2 = 11 &\rightarrow \{3, 8\}, \{4, 7\}, \{5, 6\}; \\
s^2 = 12 &\rightarrow \{4, 8\}, \{5, 7\}; \\
s^2 = 13 &\rightarrow \{5, 8\}, \{6, 7\}; \\
s^2 = 14 &\rightarrow \{6, 8\}; \\
s^2 = 15 &\rightarrow \{7, 8\};
\end{aligned} \tag{14}$$

analogous to table (13) for the given odd cardinality $n = 9$ of the set $N^9 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, which will have three “main” diagonals:

$$\begin{aligned}
s^2 = 3 &\rightarrow \{1, 2\}; \\
s^2 = 4 &\rightarrow \{1, 3\}; \\
s^2 = 5 &\rightarrow \{1, 4\}, \{2, 3\}; \\
s^2 = 6 &\rightarrow \{1, 5\}, \{2, 4\}; \\
s^2 = 7 &\rightarrow \{1, 6\}, \{2, 5\}, \{3, 4\}; \\
s^2 = 8 &\rightarrow \{1, 7\}, \{2, 6\}, \{3, 5\}; \\
s^2 = 9 &\rightarrow \{1, 8\}, \{2, 7\}, \{3, 6\}, \{4, 5\}; \\
s^2 = 10 &\rightarrow \{1, 9\}, \{2, 8\}, \{3, 7\}, \{4, 6\}; \\
s^2 = 11 &\rightarrow \{2, 9\}, \{3, 8\}, \{4, 7\}, \{5, 6\}; \\
s^2 = 12 &\rightarrow \{3, 9\}, \{4, 8\}, \{5, 7\}; \\
s^2 = 13 &\rightarrow \{4, 9\}, \{5, 8\}, \{6, 7\}; \\
s^2 = 14 &\rightarrow \{5, 9\}, \{6, 8\}; \\
s^2 = 15 &\rightarrow \{6, 9\}, \{7, 8\}; \\
s^2 = 16 &\rightarrow \{7, 9\}; \\
s^2 = 17 &\rightarrow \{8, 9\}.
\end{aligned} \tag{15}$$

The noted difference between tables (14) and (15) lies in the number of “main” diagonals for even (one) and odd (always three) cardinality n of the original set N^n . Thus, in the examples the computed cardinality k of the subset N^k can range from 2 to 9 with *basic* union of elements within a row in these tables according to diagonal traversal, with no overlapping indices. In particular, with the basic union of elements from the third to the eleventh rows of table (14) (union of two-element subsets in the direction of the “main diagonal”), the number of subsets N^4 of cardinality $k = 4$ will be 22. With the special union (union of two-element subsets in the direction of the main diagonal), the number of subsets N^4 of cardinality $k = 4$ equals 48. The total number of subsets N^4 will be $48 + 22 = 70$ for both unions. According to the combination function C_8^4 the exact number of subsets N^4 will be 70.

Example 2. Given the certificate $S^4 = 120$ and the sets $X^8 = \{10, 14, 17, 20, 36, 38, 43, 47\}$, $N^8 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ of cardinality $n = 8$, index certificate $s^4 = 20$. It is required to find subsets X^4 of cardinality $k = 4$ with certificate S^4 for solving the subset sum problem (6). Let us find all subsets N^2 , s^2 in the form of matrix (10):

$$N^2 = \left\{ \begin{array}{l} \{1,2\}\{1,3\}\{1,4\}\{1,5\}\{1,6\}\{1,7\}\{1,8\} \\ \{2,3\}\{2,4\}\{2,5\}\{2,6\}\{2,7\}\{2,8\} \\ \{3,4\}\{3,5\}\{3,6\}\{3,7\}\{3,8\} \\ \{4,5\}\{4,6\}\{4,7\}\{4,8\} \\ \{5,6\}\{5,7\}\{5,8\} \\ \{6,7\}\{6,8\} \\ \{7,8\} \end{array} \right\}, \quad (16)$$

$$s^2 = \left\{ \begin{array}{l} 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \\ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ 7 \ 8 \ 9 \ 10 \ 11 \\ 9 \ 10 \ 11 \ 12 \\ 11 \ 12 \ 13 \\ 13 \ 14 \\ 15 \end{array} \right\}$$

The basic union involves determining $\frac{s^4}{2} = s^2 = 10$ and the subsets $N^2 = \{2,8\} \vee \{3,7\} \vee \{4,6\}$, the number of N^4 equals $C_3^2 = 3$, that is, $N^4 = \{2,8\} \cup \{3,7\} \vee \{2,8\} \cup \{4,6\} \vee \{3,7\} \cup \{4,6\}$. However, these subsets do not yield certificates equal to $S^4 = 120$, namely, $S^4 = 121 = 61 + 60 \vee S^4 = 119 = 61 + 58 \vee S^4 = 118 = 60 + 58$. Therefore, the transition from the special union to the basic one is started with the minimum index certificate $20 - 3 = 17 \vee 20 - 4 = 16$. The values of these certificates are greater than the maximum index certificate $7 + 8 = 15$. Therefore, the next index certificates will be $s^2 = 1 + 4 = 5, s^2 = 20 - 5 = 15 = 7 + 8$. For the first certificate we have the subsets $N^2 = \{1,4\} \vee \{2,3\}$, and for the second— $N^2 = \{7,8\}$. The answer is the subset $N^4 = \{1,4\} \cup \{7,8\}$ and the equality holds $S^4 = 120 = 30 + 90$. It is important to note here that for the special union and the given index certificate s^4 the conditions hold $T \leq O(n^2) = O(64), S \leq O(n^2) = O(64)$.

Let us subtract the given certificate s^4 from the full certificate. Then for the found certificate $s - s^4 = 16$ we find s^2 , equal to $16/2 = 8$. Next, using the diagonal in the first matrix or the row in table (14) with index certificate $\frac{s^4}{2} = s^2 = 8$, according to formulas (11) $N^k = \{\bigcup_m N_m^2\}$ then we have $N^4 = \{\bigcup_2 N_2^2\}$, $N^4 = \{1,7\} \cup \{2,6\} \vee \{2,6\} \cup \{3,5\}$. Answer: $X^4 = X^8 \setminus (\{x_1, x_7\} \cup \{x_2, x_6\}) \vee X^4 = X^8 \setminus (\{x_2, x_6\} \cup \{x_3, x_5\})$, after performing the set subtraction operation we finally have $X^4 = \{x_4, x_5\} \cup \{x_3, x_8\} \vee X^4 = \{x_1, x_8\} \cup \{x_4, x_7\}$, $T \leq O(n^2) = O(64), S \leq O(n^2) = O(64)$.

With the basic union the computed index certificates: $s^2 = 9, N^2 = \{1,8\} \vee \{2,7\} \vee \{3,6\} \vee \{4,5\}; s^2 = 11, N^2 = \{3,8\} \vee \{4,7\} \vee \{5,6\}$. Here the given index certificate $s^4 = 9 + 11$. The answer is the target subset: $N^4 = \{1,8\} \cup \{4,7\} \vee N^4 = \{4,5\} \cup \{3,8\}$. Thus, this answer coincides with the previous answer, and the set subtraction operation is not needed. The inequalities also hold here $T \leq O(n^2) = O(64), S \leq O(n^2) = O(64)$.

Example 3. Given the certificate $S^3 = 100$ and the sets $X^8 = \{10, 14, 17, 20, 36, 38, 43, 47\}$, $N^8 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ of cardinality $n = 8$. It is required to find subsets X^3 of cardinality $k = 3$ with certificate S^3 for the subset sum problem (6). Then according to table (13) with corresponding index certificates and formulas (11), we have that $N^k = i \cup (\bigcup_m N_m^2) \vee (\bigcup_m N_m^2) \setminus i$. The description of these formulas and found answers is as follows:

$$N^3 = \bigcup_1 N_1^2 \cup i, N_1^2 \cup i = \{7,8\} \cup \{1\}, s^3 = s^2 + i = 15 + 1 = 16, \text{answer : } m = 1, k = 2 + 1 = 3, X^3 = \{x_7, x_8, x_1\}, (x_7 + x_8) + x_1 = 100, T \leq O(n^2) = O(64), S \leq O(n^2) = O(64),$$

or

$\cup_2 N_2^2 \setminus i = (\{1, 8\} \cup \{2, 7\}) \setminus \{2\}$, $s^3 = s^4 - i = 18 - 2 = 16$, answer: $m = 2$, $k = 4 - 1 = 3$,
 $X^3 = \{x_1, x_8, x_7\}$, $(x_1 + x_8) + (x_2 + x_7) - x_2 = 100$, $T \leq O(n^2) = O(64)$, $S \leq O(n^2) = O(64)$.

Example 4. Finally, let us give an example confirming the conclusions of item 5 of Corollary 3 for the original set $N^8 = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Indeed, from the “main” diagonal of the first matrix (16) we have four subsets N^2 with index certificate $s^2 = 9$, six subsets N^4 , $s^4 = 18$, four subsets N^6 , $s^6 = 27$, one subset (the full set itself) N^8 , $s^8 = 36$. If the Vandermonde convolution needs to be applied, all target subsets N^k can be obtained from the “main” diagonal of the first matrix (16) according to the combination function (3).

4.2. Numerical Experiments and Analysis of Results

This section presents a comparative experimental evaluation of the proposed indexed construction method against classical approaches to the subset sum problem. The study examines execution time, memory usage, and solution completeness across different subset cardinalities.

4.2.1. Experimental Setup

The experiments were conducted on a desktop computer with the following configuration: Intel(R) Core(TM) i5-8500 CPU @ 3.00 GHz, 16 GB RAM, running Windows 10 Pro (64-bit). The implementation was developed in Python 3.9 using PyCharm 2020.1.2 Community Edition.

A test instance was generated as a finite set of size $N = 50$ with integer values bounded by $M = 10,000$. For each subset cardinality $k \in [2, 12]$, a corresponding target sum S was defined. The same dataset was used across all methods to ensure comparability.

The performance of the algorithms was evaluated with respect to:

- execution time;
- memory usage.

The comparison included four approaches:

- brute-force enumeration;
- dynamic programming using bitset representation;
- meet-in-the-middle (MM);
- the proposed index-based certificate method.

All baseline methods were implemented using optimized standard techniques. The brute-force approach relies on `itertools.combinations`, which is implemented in C and provides efficient iterator-based generation without materializing all subsets. The dynamic programming method is implemented using bitset operations based on Python’s built-in integer type, enabling efficient low-level bitwise computation. The meet-in-the-middle algorithm follows the classical Horowitz–Sahni scheme with sorting and binary search operations implemented via optimized standard library functions. These implementation choices ensure that each baseline method is evaluated under near-optimal practical conditions within the Python environment.

4.2.2. Baseline Methods

The considered baseline methods represent classical approaches to the subset sum problem and are widely studied in the literature.

The brute-force method enumerates all possible subsets of a given cardinality and guarantees completeness, but exhibits exponential growth in computational cost due to the combinatorial nature of the search space. In practice, the number of candidate subsets

increases rapidly, making this approach impractical even for moderate problem sizes. This method is commonly used as a reference for exact enumeration and benchmarking [8].

Dynamic programming methods, particularly bitset-based implementations, exploit the structure of the target sum and demonstrate pseudo-polynomial behavior. Classical formulations rely on knapsack-style dynamic programming techniques [6,8], while more recent developments improve practical performance under specific conditions [5,9]. These methods are widely used due to their stability and efficiency in structured settings.

Meet-in-the-middle algorithms, originally proposed by Horowitz and Sahni [3] and further developed by Schroepel and Shamir [4], reduce the effective search space by splitting the problem into two parts and combining partial solutions. These methods significantly improve performance compared to brute-force enumeration but require substantial memory to store intermediate results. They represent a classical time–space trade-off in exact combinatorial search.

More broadly, subset sum and related problems have been extensively studied in the context of knapsack-type optimization and combinatorial enumeration [10–12], providing a rich theoretical and algorithmic foundation for comparison.

These approaches constitute widely used reference methods for exact subset construction and provide a benchmark for evaluating the proposed indexed framework.

4.2.3. Results and Scaling Behavior

The experimental results are presented in Tables 1–3.

Table 1. Execution time (seconds).

<i>k</i>	Brute Force	DP	MM	Proposed Algorithm
2	0.000832	0.004859	0.002704	0.000440
3	0.013477	0.056155	0.015735	0.000686
4	0.162232	0.175204	0.096054	0.001880
5	1.487983	0.347012	0.489582	0.009901
6	11.801322	0.582465	1.868361	0.053281
7	75.863641	0.855779	5.661248	0.276737
8	410.724292	1.226117	14.392234	1.408649
9	1930.081431	1.607236	32.073095	5.771941
10	8194.289910	2.185671	67.489330	21.926987
11	30,392.163831	2.629544	184.990949	75.611436
12	95,885.083828	3.567317	406.939238	244.409451

Table 2. Memory usage (MB).

<i>k</i>	Brute Force	DP	MM	Proposed Algorithm
2	0.00	0.09	0.10	0.09
3	0.00	1.23	0.74	0.16
4	0.00	3.21	4.68	0.26
5	0.09	5.46	22.61	0.39
6	0.56	9.72	85.17	0.63
7	3.28	13.62	262.94	1.23

Table 2. *Cont.*

<i>k</i>	Brute Force	DP	MM	Proposed Algorithm
8	16.68	17.70	678.20	3.73
9	73.30	21.96	1489.38	16.99
10	283.11	26.37	2856.66	62.79
11	973.54	31.00	4914.39	213.73
12	3032.79	40.58	7717.29	661.01

Table 3. Number of valid subsets identified by different methods.

<i>k</i>	Brute Force	DP (Feasibility)	MM	Proposed Algorithm
2	0	0	0	0
3	0	0	0	0
4	13	1	13	13
5	133	1	133	133
6	800	1	800	800
7	4664	1	4664	4664
8	23,754	1	23,754	23,754
9	104,346	1	104,346	104,346
10	403,260	1	403,260	403,260
11	1,385,718	1	1,385,718	1,385,718
12	4,320,584	1	4,320,584	4,320,584

Note: The dynamic programming method is implemented in a feasibility form and reports only the existence of a valid subset rather than enumerating all solutions. The brute-force method is used as a reference for complete enumeration, providing a ground truth for validating the correctness and completeness of the other methods. The agreement with brute-force results confirms that the proposed method preserves completeness while reducing the explored search space.

The results demonstrate distinct scaling behaviors across the considered methods as the subset cardinality k increases. The brute-force method exhibits rapid exponential growth in execution time. While it performs efficiently for small values of k , it becomes computationally impractical as k increases. For example, execution time grows from fractions of a second at low cardinalities to several hours for larger values of k .

The meet-in-the-middle method improves upon brute-force enumeration by reducing the effective search space. However, it still demonstrates substantial growth in both execution time and memory usage. In particular, memory consumption increases significantly due to the need to store intermediate subsets and partial sums.

The dynamic programming approach demonstrates stable performance due to its feasibility-based formulation. In this implementation, the method determines the existence of a valid subset rather than enumerating all solutions. As a result, its execution time remains relatively insensitive to the number of feasible subsets and is primarily influenced by the magnitude of the target sum.

The proposed index-based method exhibits a different scaling profile. While execution time increases with k , the growth remains more controlled compared to exhaustive enumeration methods. This behavior can be attributed to the structured nature of the indexed representation, which constrains candidate generation through admissible index intervals

and avoids explicit traversal of the full combinatorial space. In terms of memory usage, the proposed method demonstrates significantly lower requirements compared to the meet-in-the-middle approach. This is due to the absence of large intermediate data structures and the reliance on index-based representations rather than explicit subset storage. Importantly, the observed performance differences are not attributable to implementation artifacts, as all baseline methods were implemented using optimized standard library components. Overall, the results indicate that the indexed construction framework provides an alternative computational pathway for subset construction, particularly effective for moderate instance sizes where structural constraints can be exploited.

The experiments are intended to provide a comparative evaluation on moderate-size instances and to illustrate scaling behavior across methods, rather than to serve as a definitive large-scale benchmark. The chosen problem size ($N = 50$) is sufficient to reveal differences in computational behavior across methods, while broader large-scale validation remains a direction for future work.

5. Discussion

The proposed indexed construction framework can be applied to several classical combinatorial problems that admit formulations in terms of subset selection. In particular, the subset sum problem can be naturally expressed within the indexed framework, where the task reduces to identifying subsets $X^k \subseteq X^n$ whose elements satisfy a given target sum. The indexed representation allows this problem to be reformulated in terms of index certificates, enabling a structured exploration of candidate subsets.

The partition problem, which considers splitting a set of distinct positive integers into two subsets with equal sums can be interpreted as a special case of subset sum, where the target value corresponds to half of the total sum of the set. Within the indexed framework, this formulation enables the construction of subsets satisfying balance constraints through controlled index combinations.

Graph-based problems, such as the independent set and vertex cover problems, are classical NP-complete problems [1,2] and can also be expressed in terms of subset selection over index sets. In this context, the relationship between independent sets and vertex covers can be represented through complementary subsets N^k and $N^{(n-k)}$, allowing these problems to be analyzed within the same indexed structure.

The common feature across these problems is the need to construct subsets under combinatorial constraints. The indexed construction framework provides a unified representation in which subset generation is guided by index-based parameters rather than by direct enumeration of the full power set.

The novelty of the proposed approach lies in replacing explicit subset enumeration with structured navigation in index space governed by admissible constraints.

From a practical standpoint, this approach facilitates the structured organization of candidate subsets, reduces redundant combinatorial exploration, and enables efficient subset construction for small- and medium-sized instances. At the same time, the framework is intended as an alternative representation and construction mechanism. The obtained results should be interpreted within the indexed setting and do not imply a general polynomial-time solution for NP-complete problems.

6. Conclusions

This work examined subset construction in NP-complete problems through an index-based representation framework. By exploiting the equicardinality between a finite set and its corresponding index set, elements can be represented and manipulated through their indices, enabling an alternative formulation of subset selection problems.

Within this framework, subset construction is organized through index certificates and admissible index ranges, allowing candidate subsets to be generated and analyzed without explicit traversal of the full power set. The use of diagonal traversal provides a structured mechanism for organizing pairwise combinations and constructing higher-cardinality subsets.

The proposed framework offers a unified perspective for analyzing several classical NP-complete problems, including the subset sum problem, the partition problem, the independent set problem, and the vertex cover problem, all of which can be expressed in terms of subset selection within indexed representations.

From a computational perspective, the results demonstrate that the indexed construction procedure can be organized with polynomial bounds within the indexed representation of pairwise structures. At the same time, these bounds apply specifically to the structured construction process and do not imply a general polynomial-time solution for NP-complete problems.

Overall, the proposed approach provides a structured and analytically transparent framework for studying subset construction processes and may support further development of algorithms based on constrained exploration of combinatorial search spaces.

Author Contributions: B.S. contributed to the conceptual development of the indexed construction framework and the underlying theoretical formulation; A.S. contributed to methodological structuring, formalization, algorithmic interpretation, and manuscript preparation; T.S. contributed to validation, computational analysis, and manuscript review; K.B. contributed to software implementation, computational experiments, and experimental data generation; A.M. and A.A. contributed to literature analysis, verification of results, and editorial review of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP26101119).

Data Availability Statement: The data are available within the article. No additional datasets were generated or publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cook, S.A. The complexity of theorem-proving procedures. In Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights, OH, USA, 3–5 May 1971; pp. 151–158.
2. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; IBM Research Symposia Series; Springer: Boston, MA, USA, 1972; pp. 85–103.
3. Horowitz, E.; Sahni, S. Computing partitions with applications to the knapsack problem. *J. ACM* **1974**, *21*, 277–292. [\[CrossRef\]](#)
4. Schroeppe, R.; Shamir, A. A $T = O(2^{\lceil n/2 \rceil})$, $S = O(2^{\lceil n/4 \rceil})$ algorithm for certain NP-complete problems. *SIAM J. Comput.* **1981**, *10*, 456–464. [\[CrossRef\]](#)
5. Koiliaris, K.; Xu, C. A Faster Pseudopolynomial Time Algorithm for Subset Sum. In Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Barcelona, Spain, 16–19 January 2017; pp. 1062–1072.
6. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 3rd ed.; MIT Press: Cambridge, MA, USA, 2009.
7. Sinchev, B.; Sinchev, A.; Mukhanova, A.M. Algorithm based on the subset sum problem for high performance computing. In *Springer Link Proceedings of Ninth International Congress on Information and Communication Technology*; Springer: Singapore, 2024; pp. 627–637.
8. Kellerer, H.; Pferschy, U.; Pisinger, D. *Knapsack Problems*; Springer: Berlin/Heidelberg, Germany, 2004.
9. Bringmann, K. A Near-Linear Pseudopolynomial Time Algorithm for Subset Sum. In Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Barcelona, Spain, 16–19 January 2017; pp. 1073–1084.
10. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; Freeman: San Francisco, CA, USA, 1979.

11. Pisinger, D. Linear time algorithms for knapsack problems with bounded weights. *J. Algorithms* **1999**, *33*, 1–14. [[CrossRef](#)]
12. Nederlof, J. Fast polynomial-space algorithms using inclusion–exclusion. *Algorithmica* **2013**, *65*, 868–884. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.