N E W S
OF THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
SERIES OF GEOLOGY AND TECHNICAL SCIENCES
ISSN 2224-5278
Volume 3, Number 441 (2020), 97 – 101 **https://doi.org/10.32014/2020.2518-170X.59**

UDC 615.035.4

B. Sinchev[1], A. B. Sinchev[2], Zh. Akzhanova[3], Y. Issekeshev[4], A. M. Mukhanova[5]

[1]International University of Information Technology, Almaty, Kazakhstan;
[2]National Information Technology JSC, Nur-Sultan, Kazakhstan;
[3]Foundation of the First President - Elbasy, Nur-Sultan, Kazakhstan;
[4]ISS Corporation, Almaty, Kazakhstan;
[5]Almaty University of Technology, Almaty, Kazakhstan.
E-mail: sinchev@mail.ru, askar.sinchev@gmail.com,
zyekudayeva@gmail.com, yissekeshev@gmail.com, nuraksulu72@mail.ru

POLYNOMIAL TIME ALGORITHMS FOR SOLVING NP-COMPLETE PROBLEMS

Abstract. The paper proposes algorithms with a polynomial time complexity and memory for solving the Subset Sum problem. The algorithm is obtained by using a combination function and mapping, the arguments of which are input data with a given length and a certificate. The work with indices of the input data is based on combination function and the combination generation algorithm. Mapping and combination function help to deal with an exponential time complexity of existing algorithms and solve combinatorial problems with constraints. In fact, the proposed algorithms solve the Millennium problem posed by S.A. Cook in a polynomial time. The proposed algorithms are applicable for input data with a given length together with the certificate with a given value.

Key words: polynomial algorithms, the Subset Sum problem, NP-complete problems.

Introduction. One of the most important problems in computer science is the equality of classes P and NP. This problem was formulated in 1971 and still remains unsolved. Currently, the completeness of more than 3000 problems from the NP class has been proved. In addition, P vs NP is one of the seven Millennium Prize Problems, which emphasizes the problem's enormous complexity and fundamental nature.

The main idea of NP-complete is that NP problems can be reduced to NP-complete in polynomial time. Examples of NP-complete problem are the Knapsack problem and the Subset Sum problem among others. The Knapsack problem is a combinatorial optimization problem with constraints. Moreover, the Subset Sum problem is a subproblem of the Knapsack problem. A pseudo-polynomial algorithm exists to solve the Knapsack problem, the algorithm using dynamic programming. Therefore, all NP-complete problems are important.

Let us cite the famous S.A. Cook's problem setup: Could the verification of the correctness of a solution to the problem take longer (in terms of time) than the time it takes to find that solution, regardless of the verification algorithm. In other words, Cook's problem states that the time for the verification of any solution is less than the time required to solve the problem.

In [1] algorithms of combinatorial problems were divided into verification and solving problems, with the possibility to study their complexity. In addition, it implies dependence on the input data X and certificate S in Boolean form, $A(X,S)=1$.

The complexity of the algorithm disclosed in [1,2] is determined based on the function $f(n) = O(g(n)) \leftrightarrow \exists(C > 0), n_0: \forall(n > n_0)\, f(n) \leq Cg(n)$. The function $f(n)$ is asymptotically upper bounded by $g(n)$ by up to a factor C. An algorithm is polynomial if the complexity of function $f(n)$ can be represented by $f(n) = O(n^k)$, where k is a constant regardless of the length of the input data n. This is equivalent to $f(n) = O(p(n))$, in which the degree of the polynomial $p(n)$ does not exceed k.

The aim of this paper is to solve any NP-complete problem in a reasonable time, but not the proof of Cook's problem.

First of all, we should find algorithms which can solve the problem in a reasonable (polynomial) time. Let us note that the verification algorithms have polynomial time complexity.

The solution to the problem. As a basic problem of the NP-complete class, let us consider the Subset Sum problem. The computational complexity of the Subset Sum problem depends on the size of the input data *n* and the accuracy *p* (defined as the number of binary bits in the numbers that make up the set and certificate *S*). This exponential dependency on the input size was established when solving the Knapsack problem in [3,4].

The aforementioned algorithms are considered the best among the known exponential time algorithms for solving the Subset Sum problem.

However, these algorithms do not handle the indices of the original sets for defining subsets, whose sum of elements is equal to the certificate *S*, and only later on the Gray code is supposed to be used. In addition to that, an exponential growth complicates their application in practice. In the process of solving the problem, additional information embedded in the certificate *S* was not used.

Questions always arise to address these limitations. In [5], a new approach was proposed for solving the Subset Sum problem.

The idea of the approach is as follows:

– determine a range to which the certificate *S* belongs;

– determine the dimension m of the subset X_m over that range;

– define mapping $\tau(x, S)$ of the original set X^n into another set Y^n: $y_i = \tau(x_i, S), x_i \epsilon X^n, i = 1,2, \dots, n$;

– check the condition $y_i = y_j, i \neq j$, that verifies the certificate $S = x_i + x_j$;

We propose new algorithms for the Subset Sum problem X_m with dimensions two, three and four (m=2, m=3, m=4) belonging to the n-dimensional set X^n, with the advantage of time and memory optimization. For larger dimensions including four or more, we consider the composition of the Subset Sum algorithms X_m with dimensions two, three and more.

In the next paper [6] we further develop the proposed approach, where variable m could take even numbers 4,6,8,10 or more.

The idea of developing the approach is as follows:

– construct the subset $Z^l = \{z_1, z_2, \dots, z_l\}$ consisting of k elements $x_i \in X^n$ with indices determined by the generation algorithm of the combination C_n^k from the set X^n, $l = C_n^k, k \leq m/2$;

– define the mapping $\tau(z, S)$ of the subset Z^l into the set Y^l: $y_i = \tau(z_i, S), z_i \epsilon Z^l, i = 1,2, \dots, n$;

– the condition $y_i = y_j$ ($i \neq j$) guarantees that a subset $X_m = \{z_i\} \cup \{z_j\}$ with mismatching indices and automatic implementation of certificate $S = \sum_k \{z_i\} + \sum_k \{z_j\}$.

In other words, this corresponds to forming subsets X_m with parameters m=4, m=6 and more. Here, the certificate calculations $S = \sum_k \{z_i\} + \sum_k \{z_j\}$ and the complexity of the algorithm is reduced at least twice. The inequality $i \neq j$ means that you can always choose the indices i, j to form a subset of X_m with mismatching indices (which follows from the combination generation algorithm). This is the underlying essence of the proposed algorithms for solving the Subset Sum problems. The formation of subsets X_m with odd values for the parameter m was fully described in [5].

The algorithm is summarized as follows:

$$X^n \rightarrow C_n^k \rightarrow Z^l \rightarrow \tau(z, S) \rightarrow Y^l \rightarrow y_i = y_j \rightarrow X_m = \{z_i\} \cup \{z_j\}, \; l = C_n^k, k \leq \frac{m}{2}.$$

Further formation of these subsets with parameter $m > \frac{n}{2} + 1$ is not necessary. This follows from the basic properties of the combination function. The form of the map τ (z, S) is given in [5].

Methods of reducing complexity of the proposed algorithms. *Method 1.* For the value $S_i^{''} = \sum_k \{z_i\}$, a subset $\{z_i\}$ is formed from Z^l. The value of $S_i^{'}$ is found by the equation $S_i^{'} = S - S_i^{''}$. The subset $\{z_j\}$ corresponding to $S_i^{'}$ is found by the algorithm using the set X^n or via the subset Z^l using a binary search. Moreover, these quantities allow us to form the subsets $X_m = \{z_i\} \cup \{z_j\}$ with mismatching indices from the original set X^n based on $S = S_i^{'} + S_i^{''}$. The complexity of the algorithms is stepwise reduced by two or more times.

Method 2. Dividing the set X^n into k subsets with lower dimension. Then, combinations of subsets are considered to form X_m. The complexity of the algorithms is stepwise reduced by two or more times.

Method 3. Parallelization of the necessary operations generated by a simple software implementation of algorithms for solving the Subset sum problem in order to use all the existing capabilities of computing devices and hardware. The complexity of the algorithms is stepwise reduced by two or more times.

Thus, a scientific direction for the study of problems from the NP-complete class has been developed.

Let us give an example to show how the algorithm works. Suppose there is a set X^8 ={17,43,38,14,20,10,36,47} with dimension n=8. The problem asks to determine if there exists a subset X_m ={x_i, x_j, x_k, x_h}, where the sum of the subset's elements fulfils S=120. Here the certificate S is $S \in [S^4_{min}, S^4_{max}] = [61,164]$. Therefore, the parameter m=4. Let's sort the original set X^8 in ascending order, then the sorted X^8={10,14,17,20,36,38,43,47}. Next, form a subset $Z^l = \{z_1, z_2, \dots, z_l\}$, consisting of two elements x_i with indices defined based on C^k_n, from the set X^n, $l = C^k_n, k = \frac{m}{2}, l = 28$. Now, now apply the algorithm from [5] or the algorithm given above to the subsets to form subsets Then, apply the algorithm from [5] or the algorithm given above to the subset Z^l to form subsets X_4 ={x_1, x_4, x_7, x_8} or X_4 ={x_3, x_4, x_5, x_8}, where the sum of elements of each of them is 120.

Now we show the possibility of applying these algorithms to any set X^n and any subset X_m, for example, X^{1024}, X_{60}, S. We divide the sorted set X^n into 32 subsets with dimension 32. Then we get a subset Z^l based on the combination C^{30}_{32}, $l = 486$. The number of subsets X_{30} will be $C^2_{32} = 486$ to form the subset X_{60}. Thus, to construct a subset X_{60} with an arbitrary certificate S, 234196 combinations are required. This means that we have shown the possibility of solving the Subset Sum problem in a reasonable time, more precisely, solving the problem of the sum of subsets with modern computers.

It is easy to determine the upper bound for the run time of the algorithm and the required memory. Then the running time of the algorithm from the second approach will be $T = O(C * l)$, for some constant C, the required memory is $M = O(l)$, $l = C^k_n, k \le \frac{m}{2}$.

These estimates allow us to state that the Subset Sum problem is solvable on modern computers, and the brute force approach for the Subset Sum problems even using modern computers is not reasonable.

Conclusion. We conclude that although the question of the equality of classes P and NP has not yet been solved, many scholars tend to believe that they are not equal. This statement is valid for the problem posed by Cook and it follows from the proposed algorithms. But the final point in the dispute will be set only by rigorous mathematical proof.

However, the effectiveness of the proposed polynomial-time algorithms found on the basis of the developed approach to solve the Subset Sum problems, including NP-complete problems, confirms the existence of a solution to the same problem in a reasonable amount of time. This conclusion is very important for practice and computer science. In fact, a method is proposed for solving the Millennium problem posed by Cook with a polynomial time algorithm. A patent [7] for a computer system with a high processing speed of big data was obtained.

Finally, we note possible applications

The application scope of the results in practice:	The application scope of the results in theory:
1. Big data;	1. Search problem;
2. Search engines;	2. Satisfiability problem;
3. Encryption systems;	3. Theory of algorithms;
4. Coding systems;	4. Decision problem;
5. Banking systems;	5. Encipherement problem
6. Payment systems;	6. Encryption problem;
7. Intelligent systems;	7. Knapsack problem;
8. Medical diagnostic systems;	8. Traveling salesman problem;
9. Many other systems.	9. Many other problems.

Acknowledgments. We would like to acknowledge the assistance of Nurlan Abdukadyrov (PhD student, university of Illinois at Chicago), Sreenivas (Vas) Vedantam (Juris Doctor, Patent Attorney,

Baker McKenzie), and Claudia Borer (Dr. phil.-nat. (Physics), Patent Technical Advisor, Baker McKenzie) in preparing and checking our work.

Б. Синчев[1], А. Б. Синчев[2], Ж. А. Акжанова[3], Е. О. Исекешев[4], А. М. Муханова[5]

[1]Халықаралық ақпараттық технологиялар университеті, Алматы, Қазақстан;
[2]«Ұлттық ақпараттық технологиялар» акционерлік қоғамы, Нұр-Сұлтан, Қазақстан;
[3]Қазақстан Республикасы Тұңғыш Президенті – Елбасы Қоры, Нұр-Сұлтан, Қазақстан;
[4]ISS Corporation, Алматы, Қазақстан;
[5]Алматы технологиялық университеті, Алматы, Қазақстан

NP-COMPLETE КЛАСЫ ПРОБЛЕМАЛАРЫНА АРНАЛГАН ПОЛИНОМИАЛДЫ АЛГОРИТМДЕР ТУРАЛЫ

Аннотация. Мақалада жиынтықтар есебін шешуге арналған көпмағыналы уақыт пен жадқа тәуелділігі ұсынылған алгоритмдер сипатталған. Бұл көпмүшелік тәуелділік біріктірілген және дисплей функциясын енгізу арқылы алынады, олардың дәлелі берілген ұзындықтағы және куәлікпен енгізілетін мәліметтер болып табылады. Кіріс деректерінің индекстерімен жұмыс комбинация функциясы мен құрама алгоритмге негізделген. Енгізілген карта мен үйлесімділік функциялары экспоненциалды алгоритмдердің күрделілігіндегі өсудің жарылғыш сипатын және шектеулермен комбинаторлық оптимизация мәселелерін шешеді. Шын мәнінде, Кук жасаған көпжылдық алгоритммен мыңжылдық проблемасын шешудің әдісі ұсынылған. Бұл әдіс берілген мәні бар сертификат болған кезде ақырлы ұзындықтағы мәліметтерді енгізу үшін қолданылады. Нәтижелер жиынтықтар есебін шешудің ұсынылған нақты алгоритмдері осы алгоритмдердің жұмыс істеу уақытын едәуір қысқартады, сонымен қатар компьютерлердің, серверлердің және басқа пайдаланылған есептеу құрылғыларының жабдықтарына қойылатын талаптарды азайтады. Әзірленген алгоритмдердің негізгі тұстарын суреттейтін мысалдар келтірілген. Жиындар есебін шешуге арналған ұсынылған математикалық теория көптеген теориялық және практикалық есептерді шешуге мүмкіндік береді. Бұл NP-complete класы проблемаларды зерттеуге арналған ғылыми бағыттың мәні.

Экспоненциалды алгоритмдердің басты кемшілігі - ақпаратты өңдеуге қажетті уақыт пен жад экспонент арқылы көрсетіледі, атап айтқанда $T*M = O(2^n)$ уақытының жады (өңделген мәліметтердің n саны). Соңғы ескерту аппараттық құралдарға және ақпаратты өңдеудің басқа құралдарына қатаң талаптар қояды. Біз қолданыстағы алгоритмдердің қолданылу шегін анықтаймыз. Қойыңыз $n = 128$. 2^{64} элементтің ішкі жинын сұрыптау үшін уақыты қажет $O(2^{70})$. Қазіргі компьютерлер мен ақпараттық технологиялар 2^{66} дерек көлемінде жұмыс істей алатыны белгілі. Осылайша, тіпті осындай ішкі жиынын сұрыптау қиын.

Түйін сөздер: полиномиальды алгоритм, жиынтықтар есебі, NP-complete класы.

Б. Синчев[1], А. Б. Синчев[2], Ж. А. Акжанова[3], Е. О. Исекешев[4], А. М. Муханова[5]

[1]Международный университет информационных технологий, Алматы, Казахстан;
[2]АО «Национальные информационные технологии», Нур-Султан, Казахстан;
[3]Фонд Первого Президента Республики Казахстан - Елбасы, Нур-Султан, Казахстан;
[4]ISS Corporation, Алматы, Казахстан;
[5]Алматинский технологический университет, Алматы, Казахстан

О ПОЛИНОМИАЛЬНЫХ АЛГОРИТМАХ ДЛЯ ЗАДАЧ ИЗ КЛАССА NP-COMPLETE

Аннотация. В работе описаны предложенные алгоритмы с полиномиальной по времени и памяти зависимостью по решению задачи о сумме подмножеств. Эта полиномиальная зависимость получена путем введения функции сочетания и отображения, аргументами которого являются входные данные с заданной длиной и сертификат. Работа с индексами входных данных осуществлена на основе функции сочетания и алгоритма генерации сочетаний. Введенные отображения и функции сочетания сглаживают взрывной характер роста трудоемкости экспоненциальных алгоритмов и решения задач комбинаторной оптимизации с ограничениями. Фактически предложен метод решения задачи тысячелетия, поставленной Куком, с полиномиальным временем работы алгоритма. Метод применим для входных данных конечной длины при

наличии сертификата с заданным значением. Полученные результаты показывают, что предложенные точные алгоритмы для решения задачи о сумме подмножеств существенно сокращают время работы этих алгоритмов, а также уменьшают аппаратные требования к мощности компьютеров, серверов и других используемых вычислительных устройств. Приведены примеры, иллюстрирующие основные положения разработанных алгоритмов. Предлагаемая математическая теория по решению задачи о сумме подмножеств позволит решить многие теоретические и практические задачи. Таков смысл разрабатываемого научного направления по исследованию проблем из класса NP-complete.

Основным недостатком экспоненциальных алгоритмов является то, что требуемое время и используемая память для обработки информации выражается через экспоненту, а именно время, умноженное на память T*M = $O(2^n)$ (n-количество обрабатываемых данных). Последнее замечание налагает очень жесткие требования на аппаратные и другие средства обработки информации. Определим границу применимости существующих алгоритмов. Положим n=128. Для сортировки подмножества, состоящего из 2^{64} элементов, необходимо время $O(2^{70})$. Известно, что современные компьютеры и информационные технологии могут работать с количеством 2^{66} данных. Таким образом, даже сортировка такого подмножества затруднительна.

Ключевые слова: полиномиальный алгоритм, задача о сумме подмножеств, класс NP-complete.

Information about authors:

Sinchev B., Doctor of Technical Sciences, Professor, International University of Information Technology, Almaty, Kazakhstan; sinchev@mail.ru; https://orcid.org/0000-0001-8557-8458

Sinchev A.B., National Information Technology JSC, Nur-Sultan, Kazakhstan; askar.sinchev@gmail.com; https://orcid.org/0000-0002-7333-2255

Akzhanova Zh., Foundation of the First President of the Republic of Kazakhstan - Elbasy, Nur-Sultan, Kazakhstan; zyekudayeva@gmail.com; https // orcid.org /0000-0003-1250-8744

Issekeshev Y., «ISS Corporation», Almaty, Kazakhstan; yissekeshev@gmail.com; https://orcid.org/0000-0003-1875-5316

Mukhanova A.M., Almaty University of Technology, Almaty, Kazakhstan; nuraksulu72@mail.ru; https://orcid.org/0000-0001-6781-5501

REFERENCES

[1] J. Edmonds Paths, trees and flowers // Canadian Journal of Mathematics. 1965. Vol. 17. P. 449-467.

[2] A.V. Nikolaev A geometric approach to the problem of a section. Yaroslavl, Yaroslavl State University, 2014. 38 p.

[3] Horowitz E., Sanni S. Computing Partitions with Application to the Knapsack Problem // Journal of the ACM(JACM), 1974. T 21. P. 277-292.

[4] R. Schroeppel, A. Shamir A $T=O(2^{n/2})$, $S=O(2^{n/4})$ Algorithm for Certain NP-Complete Problem // SIAM Journal on Computing, 1981. Vol. 10, N 3. P. 456-464.

[5] Sinchev B., Sinchev A.B., Akzhanova J., Mukhanova A.M. New methods of information search. I. // News of the National Academy of Sciences of Kazakhstan, Series of Geology and Technical Sciences. Vol. 3, N 435 (2019). P. 240-246.

[6] Paper "A novel and efficient algorithm to solve subset sum problem" is available on a permanent arXiv.org of the Cornell University http://arxiv.org/abs/2003.06571. The paper provides fundamental proof of the proposed approach with theorems, lemmas and examples; it is under proceedings of the SIAM Journal on Computing after the submission in March, 2020.

[7] Sinchev B., Sinchev A.B., Akzhanova Z.A. Computing network architecture for reducing a computing operation time and memory usage associated with determining, from a set of data elements, a subset of at least two data elements, associated with a target computing operation result // Patent USPTO, 2019. 38 p.