УДК 004.021+004.023

**D. V. Maliovanyi**
**I. V. Bogach**
**L. V. Ibrahimova**

# SOLUTION OF NP-PROBLEMS VIA APPLICATION OF HEURISTIC ALGORITHMS

Vinnytsia National Technical University
Вінницький національний технічний університет

***Анотація***

*У даній статті розглянуто загальну проблематику і актуальність постановки і вирішення NP-складних задач, а також сучасну концепцію рішень даного класу задач.*

**Ключові слова:** обчислювальна складність, евристичний алгоритм, оптимізація, оптимальність рішення, гамільтонів шлях (цикл), евклідів шлях (цикл).

***Abstract***

*This paper deals with the general issues and relevance of setting and solving NP-complex problems as well as the modern concept of solutions of this class of problems.*

**Keywords:** computational complexity, heuristic algorithm, optimization, solution optimality, Hamiltonian path (cycle), Euclidean path (cycle).

## Introduction

The purpose of this paper is to describe and popularly explain both the definition of NP-complex problems and the importance of search of their solutions. There is also a review of modern approaches to such solutions to be described with some details. Some of the most famous problems are described as well as a few most common in practical use and the most efficient in P-complex optimization (heuristic) algorithms.

## NP-complexity

In general, 'NP' stands for 'Nondeterministic Polynomial' and characterizes an estimated time to be spent on searching the exact solution for the problem conditions' set on the start of the algorithm execution. In areas such as discrete mathematics, algorithmics and programming, the cyclic complexity should be understood. It is estimated by the biggest number of cycles embedded one into another that is present in the algorithm. For example, the program of printing "Hello, the world!" as a single string is considered to be O(n) = 1 – a constant time execution algorithm, while the program of printing as an array of letters 'H', 'e', 'l', 'l', 'o', … has an O(n) = n – linear execution time. Hence, the NP-complex algorithms consist of so many embedded cycles that their complexity is approaches either O(n) = n! or O(n) = $e^n$, i.e. factorial or exponential. This means the general population of cases of such algorithms' executions (except of specific cases) can't be precisely estimated. Furthermore, the real problems are encountered on application such algorithms for search for solutions of some real problems. There are two main issues:

1) Solution needs a lot of computational power in order to the huge computational complexity;
2) Solution cannot be found in efficient or limited time.

There are some cases, which would have taken at least hundreds of years even when used all the computational power available to human. For example, prediction of each possible variant of chess party is the most obvious NP-complex problem, exact solution algorithm for which can be implemented very easily. However, it has been proved there are so many combinations of turns and probable results [1, 2] that the solution can't be solved even in centuries and is complex enough to be solved on the 'perfect computer' which is, in turn, limited by three barriers (light, quantum and thermodynamical [2].

## Famous NP-complex problems

There is a class of the problems similar to the one described above. Also NP-complex problems subdivide into NP, NP-complete and NP-hard, ordered by complexity ascension. The most two famous and general kinds

of NP-hard problems are TSP (Travelling Salesman Problem) and CPP (Chinese Postman Problem). Almost all the problems of the NP class can be reduced to one.

The condition formulation of both problems looks very similar though it has a big difference. Conceptually, TSP can be described as below.

There is a set of cities placed in some coordinate space (map). The purpose is to make a path, which would contain every city on the map. Such a path is called Hamiltonian. A case where the path starts and end in the same node is called Hamiltonian cycle. This problem has two main aspects:

1) Does such a path (or cycle) exist;
2) The purpose is to find as short Hamiltonian path as possible;

Although it sometimes seems to be a trivial question on certain or enough small city sets, but it's not seems to be possible to take random path in Euclidean graph and find it good enough. It is even hasn't been found condition of Hamiltonian path and cycle existence where every Hamiltonian cycle belongs to Hamiltonian paths class but not the way reversed. The graph which has Hamiltonian path is called Hamiltonian itself. The only thing that has been proven is a sufficient condition of Hamiltonian path existence [3]. Hence, it's granted every Euclidean graph to be Hamiltonian.

The representation of CPP is given below.

There is a map with a set of cities and edges (links) between them. The purpose lays in finding the path going through every existing edge exactly once.

In cases with symmetric and/or undirected graph this problem even can be reduced so that it obtains deterministic polynomial complexity (P class can be solved in defined time). If the formed graph is fully directed (each edge has direction, and only one), such a problem is then called NYSSP (New York Street Sweeper Problem). Mixed graphs are the main problem since they are much harder to solve [4]. It is considered that in general, most complicated cases of TSP are harder to solve than most of complicated ones of CCP.

As a result, it comes to conclusion that these problems with exact solution found in NP-time aren't efficient, so a new approach was introduced to make profit on such cases and make result execution time and output relevant.

## Heuristic algorithms application to solving NP-hard problems

Even after a long and hard research, question of possibility of solving NP problems for exact result in polynomial time is not neither proved nor refuted. The decision is to benefit much faster solution execution by means of sacrifice the solution optimality. Algorithms giving good but not for granted the best result are called heuristic. They use some behavior patterns inspired either by nature or by some other processes. They, in turn, are divided into:

1) Tour construction algorithms;
2) Tour improvement algorithms;
3) Composite algorithms [5].

They differ by the way they generate the final path:

1) add a node to each iteration so the number of iterations is a number of nodes – 1;
2) every iteration generates a tour better than the previous one;
3) mix both these methods.

The most efficient and precise algorithms are the algorithms of local optimization, but they are complicated to implement and use. Moreover, they require the initial configuration from a user or some other agent and output depends strongly on the output of this algorithm.

One of the most efficient algorithms is a particular modification of n-opt algorithm – a Lin-Kernighan algorithm [5]. However, it is hard in terms of Mathematics and Programming while implementing. Another one, which proved its efficiency, is an elastic net algorithm. In simple words, this algorithm is a modification of lasso and annealing algorithms. Lasso algorithm may be represented as a nodes' ring 'thrown' around the population of cities. Every iteration this ring tightens around the cities and ring nodes 'sticks' to cities when it is close enough. When the ring is maximally tightened, the solution is found and it is a shape which an initial 'lasso' became. For the iteration and tightening optimization annealing algorithm is used. It slows lasso on start and accelerates the target function on the end, so better solutions are found.

One of the key features is a modification of lasso, so that it is 'thrown not around the entire city population but around mass center with a radius that can be adjusted manually. One more feature has been added, so that lasso can now not only be tightened but also extended to some cities distant from the main population place. This feature highly improved performance and results on widely and unregularly spread population.

It also should be noted that among the composite generation algorithms genetic ones worth attention.

## Conclusion

To summarize above, the NP-complex problems represent class of problems, which are hard to be solved with the best result, but they are often encountered in real life. The heuristic, or optimizational, algorithms are those which can give a good enough solution in limited time, but the solution may be not the best. Heuristic algorithms are subdivided into more certain classes of algorithms. Thus, the tendency shows the better heuristic algorithm is, the harder it to be understood and implemented.

## REFERENCES


1. Hector A.R.P. Predicting the Outcome of a Chess Game by Statistical and Machine Learning techniques / A.R.P. Hector — Universitat Polit`ecnica de Catalunya, 2016.
2. Bremermann H.J. Quantum Noise and Information / H.J. Bremermann, — Proc. 5th Berkeley Symp. Math. Statistics and Probability, 1965.
3. Li S. An efficient condition for a graph to be Hamiltonian / S. Li, R. Li, J. Feng — Discrete Applied Mathematics, vol. 155, no. 14, pp. 1842 – 1845, 2007.
4. Postman Problem [*Електронний ресурс*]. – Режим доступу: https://www-m9.ma.tum.de/graph-algorithms/directed-chinese-postman/index_en.html – назва з екрану.
5. Helsgaun Keld. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic / Keld Helsgaun — Department of Computer Science Roskilde University DK-4000 Roskilde, Denmark.
6. D.V.Maliovanyi. Elastic Net Algorithm Application to Travelling Salesman Problem. [*Електронний ресурс*]. – Режим доступу : https://github.com/str1k6rJP/Java-TSP-Elastic-Net-Solving-Algorithm/blob/precise-optimized/PaperTSPElasticNetUkrainian.docx – назва з екрану.



***Мальований Дмитро Вадимович*** — студент групи 1ІСТ-18б, кафедра автоматизації та інтелектуальних інформаційних технологій, Факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м.Вінниця, e-mail: fkca.1ict18.mdv@gmail.com

***Богач Ілона Віталіївна*** — кандидат технічних наук, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м.Вінниця.

***Ібрагімова Людмила Володимирівна*** — старший викладач кафедри іноземних мов, Вінницький національний технічний університет, м. Вінниця, e-mail: milatvin@ukr.net

***Maliovanyi Dmytro V***. — Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: fkca.1ict18.mdv@gmail.com

***Bogach Ilona V***. — PhD, Associate Professor of the Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia.

***Ibrahimova Liudmyla V.*** — Senior Lecture, Chair of Foreign Languages, Vinnytsia National Technical University, Vinnytsia, e-mail: milatvin@ukr.net