УДК 004.021, 004.023

I.V. Bogach
D.V. Maliovanyi

SOLUTION OF NP-PROBLEMS VIA APPLICATION OF HEURISTIC ALGORITHMS

Вінницький національний технічний університет

Анотація:
У даній доповіді 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 report examines 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. Also there is a review of modern approach of such solutions to be described with some details. Some of the most famous problems to be described as well as a few among the most common on practical use and the most efficient while P-complex optimization (heuristic) algorithms.

NP-complexity

Generally speaking, ‘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, usually the cyclic complexity should be understood. It is estimated by the biggest number of cycles embedded one in another present in the algorithm e.g. the program of printing “Hello world!” as a single string is considered to be O(n)=1 – a constant time execution algorithm while printing exactly same 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 can’t 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 (e.g. 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, but it has been proved there is so many combinations of turns and probable result[1] that the solution can’t be solved even in centuries and even complex enough to be too complex 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 about the chess 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 of them.

The condition formulation of both problems looks very similar though it has a big difference. Conceptually speaking, 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, but exactly once. 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) If such a path (or cycle) exists;

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 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 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 long and hard research, question of possibility of solving NP problems for exact result in polynomial time isn't 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, divide into[5]:

1) Tour construction algorithms;

2) Tour improvement algorithms;

3) Composite algorithms.

They differ the way they generate final path: 1) add a node each iteration so the number of iterations is a number of nodes – 1; 2) Every iteration generate tour better than previous one; and 3) mix both these methods.

The most efficient and precise algorithm are algorithms of local optimization (kind of 2)), but they are complicated to implement and in use. More than that, they require initial configuration from user or some other agent and output depends strongly on this algorithm's output.

One of the most efficient algorithms is a particular modification of n-opt algorithm – a Lin-Kernighan algorithm[5]. But it has hard mathematical and programming issues while implementation. Another one, which have proven 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 cities' population. Every iteration this ring tightens around the cities and ring nodes 'stick' to cities when 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[6].

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

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 subdivide on more certain classes of algorithm. Though not a strict rule, the tendency shows the better heuristic algorithm is, the harder it to understand and implement.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hector A. R. Predicting the Outcome of a Chess Game by Statistical and Machine Learning techniques [Electronic resource] – Electronic data. – Mode of access: https://pdfs.semanticscholar.org/6115/260e66a8e4e683a43532d8ec9017b4ab6f0f.pdf?_ga=2.257547144.261932811.1587835596-1568095603.1587835596 - Title from the screen.

2. Bremermann H.J. Quantum Noise and Information, Proc. 5th Berkeley Symp. Math. Statistics and Probability [Electronic resource] – Electronic data. – Mode of access: https://projecteuclid.org/download/pdf_1/euclid.bsmsp/1200513783 - Title from the screen.

3.Li S., Li R., and Feng J. An efficient condition for a graph to be Hamiltonian. - Discrete Applied Mathematics, 2007, vol. 155, no. 14. - 1842–1845 PP.,.

4. Postman Problem [Electronic resource] – Electronic data. – Mode of access: https://www-m9.ma.tum.de/graph-algorithms/directed-chinese-postman/index_en.html |- Title from the screen.

5. Helsgaun K. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, Department of Computer Science Roskilde University DK-4000 Roskilde, Denmark [Electronic resource] – Electronic data. – Mode of access: http://akira.ruc.dk/~keld/research/LKH/LKH-2.0/DOC/LKH_REPORT.pdf - Title from the screen.

6. Maliovanyi D.V. Elastic Net Algorithm Application to Travelling Salesman Problem [Electronic resource] – Electronic data. – Mode of access: https://github.com/str1k6rJP/Java-TSP-Elastic-Net-Solving-Algorithm/blob/precise-optimized/PaperTSPElasticNetUkrainian.docx - Title from the screen.

Богач Ілона Віталіївна, кандидат технічних наук, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, ilona.bogach@gmail.com.

Мальований Дмитро Вадимович, студент групи 1ІСТ-18б, Факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, dmytro.maliovanyi@gmail.com.

Bogach Ilona Vitaliivna, PhD, Associate Professor of the department of automation and intelligent information technologies, Vinnytsia National Technical University, ilona.bogach@gmail.com.

Maliovanyi Dmytro Vadymovych, the student of group 1IST-18b, the faculty of computer systems and automation, Vinnytsia National Technical University, dmytro.maliovanyi@gmail.com.