

*ВНЗ Укоопспілки «Полтавський університет економіки і торгівлі»*

*Кафедра комп'ютерних наук та інформаційних технологій*



# **КОМП'ЮТЕРНІ НАУКИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ (КНІТ-2022)**



## **МАТЕРІАЛИ НАУКОВО-ПРАКТИЧНОГО СЕМІНАРУ Випуск 1**

*Вересень 2021 р. - червень 2022 р.*

Полтава 2022

**КОМП'ЮТЕРНІ НАУКИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ (КНІТ-2022):** матеріали науково-практичного семінару. Випуск 1 / за ред. Ольховської О.В. – Полтава: Кафедра КНІТ ПУЕТ, 2022. – 40 с.

Збірник матеріалів науково-практичного семінару містить добірку праць присвячених актуальній проблематиці, що висвітлює питання галузі сучасних інформаційних технологій, кібернетики, інформатики, математичного моделювання, системного аналізу, програмного забезпечення інформаційних систем та теорії прийняття оптимальних рішень.

У збірці представлено матеріали, що відображають проблематику підготовки фахівців з комп'ютерних наук та інформаційних технологій.

Ум. друк. арк. 2,5  
©Кафедра КНІТ ПУЕТ, 2022

**UDC 004.42**

**DEVELOPMENT OF SOFTWARE SIMULATOR FOR THE  
CUT-OFF METHOD OF THE DISTANCE LEARNING  
COURSE ELEMENTS OF COMBINATORIAL  
OPTIMAZATION**

*Rohit Bhakat, Master's, Student majoring in 122 «Computer Sciences»Poltava University of Economics and Trade.  
rohitbhakat98@gmail.com*

***Keywords: Cut-off method, Algorithm, Starter sets, JavaScript, Simulator***

Our ability to solve large, important combinatorial optimization problems has improved dramatically in the past decade. The availability of reliable software, extremely fast and inexpensive hardware and high-level languages that make the modeling of complex problems much faster have led to a much greater demand for optimization tools.

**The Algorithm Development**

JavaScript is a dynamic programming language for computers. It is a lightweight component of web pages whose implementations enable client-side script to interact with users and create dynamic sites. It is an object-oriented programming language that is interpreted.

The software implementation put into practice that, it gives students a platform to enhance their learning skills and capabilities through the simulator.

The testing revealed that, the simulator is easy to use and has a good readability for students to work with.

The permutation generation process is essentially a “traverse” from small to large, so there would be no permutations omitted

**Algorithm**

According to algorithm design and implementation, the main algorithm process is as follows:

- (8) Set an integer variable equal to zero;

- (9) Solve APLP (this method can be a simplex method, an artificial basis method, a double simplex method, etc. - depending on the type of PLP);
- (10) On the basis of  $y^*$  - the solution of APLP Check whether condition (5) is met for  $x^*$ .
- (11) Increase  $q$  by one;
- (12) Construct an inequality-cutting off of the point  $y^*$  ;
- (13) PLP can be solved directly by the method that gives the vertex of the feasible area (simplex method or artificial basis method), and you can in the case  $n=k$  (completely combinatorial problem) to use the following known facts. First, it is common knowledge that the solution of PLP is achieved at the vertex of the feasible polyhedron.

Students who live far away from any learning institutions may find it difficult to study unless accommodations and transportation are easily available.

Remote learning eliminates distance-based educational barriers, allowing anybody, anywhere to continue their education with the use of a computer and an internet connection.

### **Conclusion**

This study focuses only on the permutation development of the algorithm. Since the number of permutations grows rapidly as the number of elements of the set increase, computer source codes should be written to implement the algorithm CUT-OFF METHOD. In addition to that, the comparison between the new algorithm with the existing ones in terms of computational time and complexity needs also to be studied.

### **References**

1. MS .NET Framework Developer Resource, URL: <http://msdn.microsoft.com/en-us/netframework>
2. R. Sedgewick. Permutation Generation Methods. Computing Surveys[J], 1977, 9( 2):137~164
3. Banks J., (Ed). (1998): *Handbook of Simulation*. New York: John Wiley and Sons. [State-of-the-art handbook for discrete modeling and simulation]