Experimentation System for Evaluation of Heuristic Algorithms to Solving Transportation Problem

Kacper Rychard, Wojciech Kmiecik, Leszek Koszalka, and Andrzej Kasprzak
Department of Systems and Computer Networks
Wroclaw University of Technology
Wroclaw, Poland
e-mail: 170866@student.pwr.wroc.pl, {wojciech.kmiecik, leszek.koszalka, andrzej.kasprzak}@pwr.wroc.pl

***Abstract*—In this paper, we focus on transportation problem and different approaches to solving it. The main goal of the research was to determine accuracy and efficiency of the most popular algorithm solving the transportation problems and to test two heuristic algorithms. The additional objective was to test the optimization algorithm depending on the solution given as an input – comparison of optimizing the simple solution and the heuristic outputs. Our studies show that the processing time needed by the optimization algorithm depends on the input solution and its accuracy mostly. The experiments resulted with a complete comparison of the algorithms and a possibility to evaluate the advantages of using each one of them.**

Keywords-transportation problem; heuristic algorithms; cost reduction; experimentation system

I. INTRODUCTION

The transportation problem is a well-known issue that almost every company faces. Basically the main problem is how to move goods from group of *m* locations (for the purposes of this paper called ‘factories’) to *n* places (‘warehouses’) in a way that minimizes the total cost of transportation [1]. Main assumptions of the problem are that the cost of transportation between given factory and warehouse depends on the quantity of goods transported (all the unit costs are known) and the acceptable solution is the one that satisfies supplies of all factories and demands of all warehouses without the negative values of allocations [2].

An example graph illustrating the problem is shown in Fig. 1.

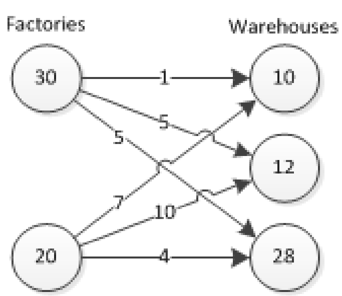

Figure 1. Example graph illustrating the transportation problem.

The factories and warehouses are represented by circles and the numbers they contain respectively stand for the factories’ supplies and warehouses’ demands. The arrows are meant to represent the shipping links and the numbers placed on them are the unit costs [3].Usually the matrix of costs is used to completely describe the problem. It comes with two vectors representing the supplies of the factories and the demands of the warehouses.

The problem can be defined [4] by a set of formulas:

- $C(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min$, (1)
- $\sum_{j=1}^{n} x_{ij} = s_i$, (2)
- $\sum_{i=1}^{m} x_{ij} = d_j$, (3)
- $x_{ij} \geq 0 \quad i = 1,2,...,m \quad j = 1,2,...,n$. (4)

The above expressions state as following:

(1) The total cost of the problem should be minimal, where *C(X)* is the total cost, c_{ij} are the unit costs and x_{ij} represent allocations,
(2) The total amount of goods sent form each factory should be equal to its supply where s_i are the factories’ supplies,
(3) The total amount of goods sent to each warehouse should be equal to its demand where d_j is the warehouses’ demands,
(4) All allocations should be non-negative.

The current papers treating the problem focus, inter alia, on the advanced modifications of the problem such as:

- Bi-criterion Transportation Problem [5],
- Fuzzy Transportation Problem [6],
- New methods of solving transportation problems [7].

Our work concerns the methods of solving the original problem and profitability of the heuristic approach used.

The rest of paper is organized as follows. Section II presents the most popular algorithms for solving the transportation problem and a short description of how they work with the most important pros and cons of using them. One authorial algorithm is described as well. Section III is a short brief about the environment of the experiments and a close-up look at the created testing tool. Section IV consists of the design of the experiments and their results with comments. This section presents the way of how the tool can be used and what information it can be used to gather. The Section V concludes the work. It also contains the plans for the future work and the development process of the application.

II. Algorithms for Soliving Transportation Problem

The algorithms tested in the paper as well as the additional objective are shown in the Table 1.

A. *North-West Corner rule (NWC)*

The most basic way of finding the solution of the problem is setting the maximum possible amount transport in each shipment link given. The maximum is calculated as the smaller number from the supply of the factory and demand of the warehouse linked. The links are considered in a sequence as they appear in the matrix of a problem. This method is supposed to provide a fast way of achieving a possible but not necessarily efficient solution [8].

B. *Cheap Means More (CMM)*

The idea of the first heuristic algorithm is to sort the connections by the unit cost and use the cheapest ones first, setting the amount of transportation as a maximum possible [9]. However, it is not likely to use this method to provide the final solution. The idea of using *CMM* as a heuristic algorithm is novel. Main advantages of this approach are the simplicity, quickness and way better solution than the *NWC*. Whilst *NWC* is not deterministic, *CMM* algorithm is (assuming that all unit costs of transportations are different, which means there is only one possible output of the sorting). The solution returned by this algorithm meets all the main requirements (satisfying all supplies and demands without negative amount of transportation) and is supposed to be close the optimal one. The flowchart of the *CMM* algorithm is shown in Fig. 2.

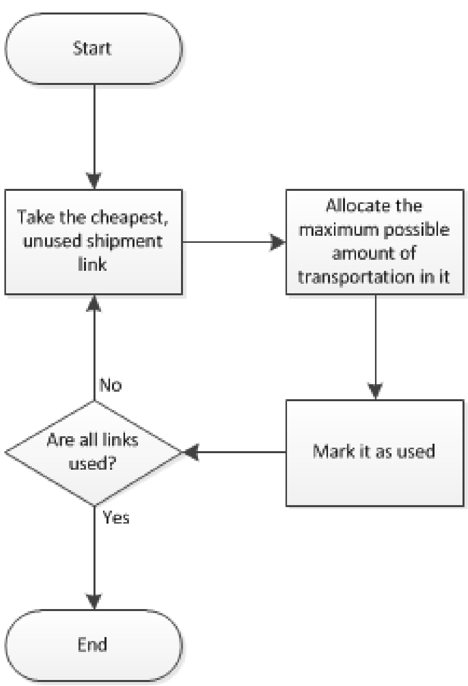

Figure 2. Flowchart of *CMM* algorithm.

TABLE I. ALGORITHMS TO TEST

Algorithm	NW corner rule	Cheap means more	Expensive means less
Optimization	✓	?	?

C. *Expensive means less (EML)*

The second heuristic algorithm is in theory similar to the previous one. The essential difference is that this one considers the most expensive shipment links first and sets the amount of transportation in them as little as possible. This minimum is calculated based on the rule best described as: ‘How small amount of good can I send / receive here to still have enough other warehouses / factories to satisfy my supply / demand? The main problem in this approach is that it postpones achieving the solution. The *CMM* algorithm was going to satisfy all of supplies and demands as quickly as possible. On each step the current situation was known and while making the decision about each allocation no assumptions were made. In this approach each decision about transporting X units of goods has additional information: ‘...assuming that I can still send / receive Y units of goods elsewhere’. After a few steps of the algorithm it may be that the assumptions made in calculating the minimum are not valid anymore and given factory / warehouse cannot satisfy its supply / demand.

It was the serious issue making the *EML* almost completely different algorithm than *CMM*. After consideration and tests of a few possible ways to solve this problem it was decided to use recursive approach to this algorithm with increasing the allocations instead of setting them. This was possible thanks to the observation that increasing any allocation by a number other than 0 means that given factory / warehouse makes an assumption of sending / receiving goods through all other links available for them at their maximums. This causes satisfying the whole supply / demands of the given factory / warehouse once the amount of transportation other than 0 is added to any allocation. Then the algorithm repeats with some values already calculated (some amounts of transportation set and some supplies / demands accordingly decreased).

The algorithm ends when no allocation was increased in a single cycle (which means there was no recursive call). Just like in the case of *CMM* the solution returned by *EML* meets all the main requirements and is supposed to be close the optimal one. The flowchart of the *EML* is shown in Fig. 3.

D. *Optimization*

Optimization algorithm takes any valid solution of the problem as an input and gives the best possible solution as an output. It checks the optimality of the solution, finds the non-used connection that should be used to reduce the total cost of transportation (if the solution was not optimal) and then adds it to the solution increasing and decreasing other allocations and repeats the described steps. It stops when the solution is optimal. The number of circles done varies and depends on the input solution – mostly on its accuracy but at some level also on other factors associated with its structure [1].

 ISBN: 978-1-61208-246-2

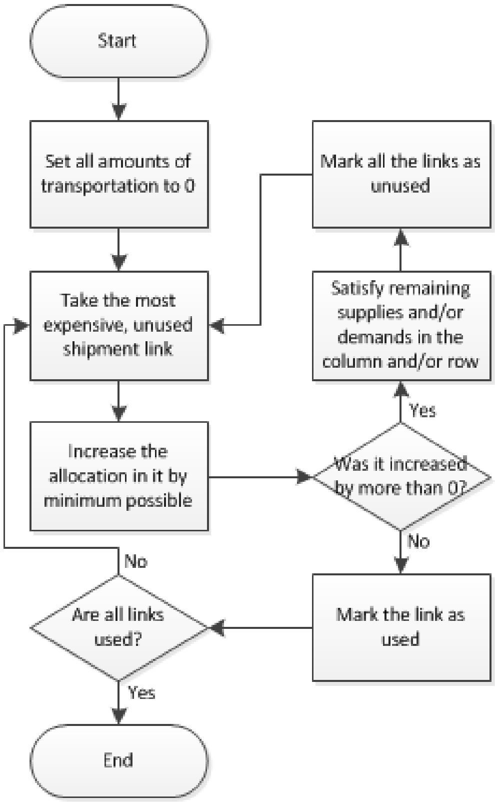

Figure 3. Flowchart of *EML* algorithm.

E. Heuristic algorithms vs. optimization

One of the main assumptions about the best solution for the transportation problems is that it uses no more than $(m + n - 1)$ connections where m is the number of factories and n is the number of warehouses.

Outputs of the described heuristic algorithms don't use more than the specified number of links. Unfortunately optimization algorithm can't handle the input solutions with less connections used either (degenerated solutions) [1]. This makes a necessity of marking some of the unused links as used with amount of transport equal 0. They should be chosen in a way that they don't create a closed cycle in a matrix corresponding to a problem. It is necessary for the algorithm of optimization to work properly. Creating methods of adding unused connections to degenerated solutions made the additional objectives of the project possible to complete.

III. Experimentation System

The testing tool was created entirely from the scratch for the purposes of the paper. I was an application implemented in C# language using Microsoft Visual Studio 2010. Class library ZedGraph was used to draw charts and present the effect of the tests in a graphical form.

The application contains two tabs which allow user to get solution for a single problem or run automatic tests of efficiency and accuracy of the algorithms.

The screenshots of the testing tool are shown in Fig. 4 and Fig. 5.

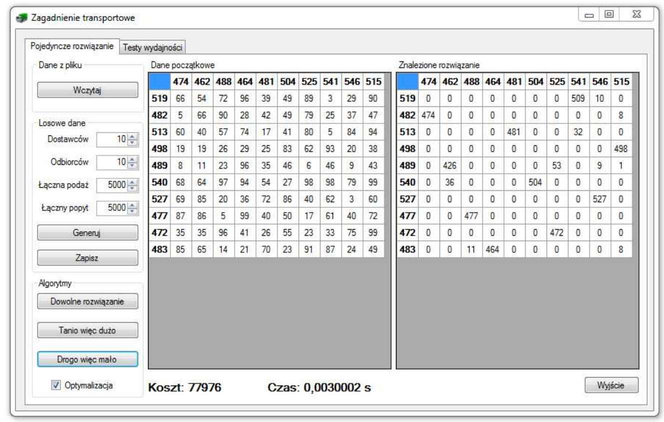

Figure 4. Single solution part of the application.

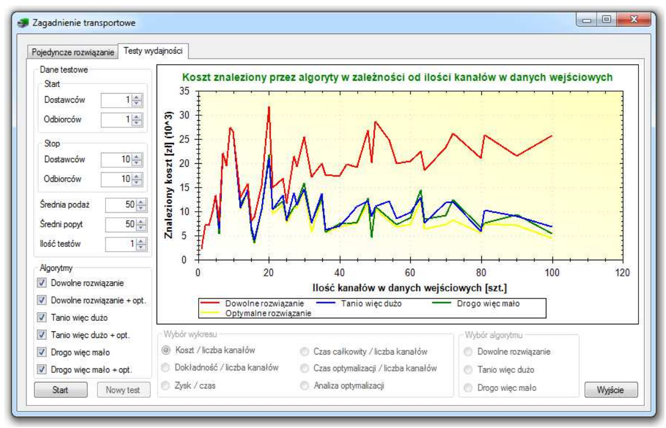

Figure 5. Automatic tests part of the application.

The application provided a complete and solid tool for testing the implemented algorithms.

IV. Research

The main part of the research was a series of experiments which were supposed to provide the data needed to determine accuracy and efficiency of the algorithms.

A. Experiment design

All the experiments were made using the presented tool. The main goal was to test the efficiency and accuracy of the implemented algorithms. The application allows user to select the range of the input data and the number of test from which the final answer is averaged. As for the amount of goods transported parameter, the user is allowed to input the average supply and demand. As the number of factories and warehouses varies during the test, each time the total supply and total demand is calculated and then the smaller value is rounded up to balance the other one.

The test were designed to deliver the information about the main characteristics of the implemented algorithm which are processing time and cost found. To allow the more valuable analysis it is possible to get information about processing time and cost reduction with optimization algorithm enabled. Before the main part of the experiments the preliminary experiment was made to determine how the results depend on the characteristics of the input data and how to choose the input data to make the tests more reliable.

Overall, several experiments were conducted in order to investigate:

- Processing time of the algorithms depending on the size of the input.
- Cost of the solution found by algorithms without optimization in comparison to the optimal one depending on the size of the input.
- Relative error of solutions found by algorithms without optimization depending on the size of the input.
- Time needed to improve the result depending on the relative error of the input solution.
- Processing time of each algorithm and optimization of its output depending on the size of the input.

All the experiments were made on a single machine within the one instance of the application with the number of tests set to 10. As the calculations have been proceeding no other actions on the machine were made.

B. *Preliminary experiment*

The preliminary experiment was based on testing the processing time of the algorithms depending on the size of the input data. It started from the 1×1 matrix and ended at the 50×50 set of data. Measure point with the same size (for example 4×6, 6×4, 2×12 etc.) were averaged. The results showed that despite the repetitive tests and averaged results the measured values spread as showed in Fig. 6. Identifying the variance points showed that the oddest results are returned when the number of the factories and warehouses in the input differ significantly (the matrix of costs is not close to a square shape).

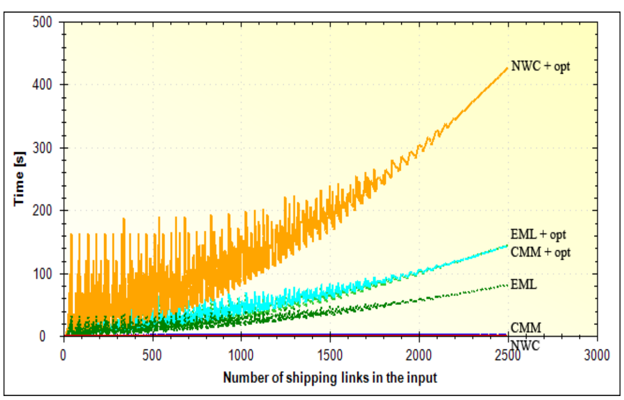

Figure 6. Preliminary experiment results.

In the further experiments it was decided to test the data with the ratio of the factories to warehouses between 0.9 and 1.1 in a range of 1×1 to 100×100.

C. *Results of Experiments*

The main experiment was made for finding the relationship between the processing time needed to return the solution and the size of the input measured by the number of shipment links. The result of the performance of the considered algorithms is shown in Fig. 7.

The result of this test is consistent with expectations. Without optimization, the first two algorithms (*NWC* and *CMM*) returned answer almost immediately. *EML* needs time to process which is probably caused by many recursive calls.

The most important observation from this part is that combining optimization with *NWC* takes much more time than calculating the best solution based on the heuristic methods. The optimization algorithm works the best with the *EML* solution as an input, but the long processing time of this algorithm causes that the quickest way of getting the optimal solution is the *CMM* plus optimization.

The second significant experiment was made for comparison of not optimized solutions found by all three algorithms with the optimum solution. The result is shown in Fig. 8. It can be easily observed that both heuristic algorithms give result that is very close to optimal. What is more, the *EML* is more accurate than the others in all cases.

The best way of evaluation of accuracy is the relative comparison of the cost of found solution and minimum cost. The relative error of the solution found by three algorithms without optimization is shown in Fig. 9. This graph shows that although the heuristic methods may seem accurate, they return the solution with cost about two times bigger than the calculated minimum. The percentage disproportion between them varies from 0 to approximately 50%.

The most important fact that this graph shows is that the bare *NWC* is unacceptable as a way of solving the transportation problem. It returns a valid solution, but it completely misses the main reason of solving the problem with a help of computer – minimizing the total cost of transportation.

The application also allows user to compare the time needed to improve the result given by the different algorithms. The result of this test is shown in Fig. 10.

This graph provides more accurate illustration of the relative error range for all three algorithms. Just as before, it is clearly visible that the heuristic algorithms return more accurate solution than the *NWC*.

Furthermore it proves that time of optimization depends not only on the relative error, but also on some other factors.

The last functionality of the created application is the analysis of two components of all algorithms joined with the optimization. The results are shown in Fig. 11, Fig. 12, and Fig. 13. These three graphs show that in case of the *NWC* and *CMM* the size of an input does not matter when it comes to the non-optimized solution. The optimizing process is the main cause of the time needed of an algorithm as a whole.

In the case of *EML* both calculating basic solution and optimizing it needs about the same amount of time. The optimization here is much faster than in the *CMM*, but the previous part takes more time so total time needed by the algorithm is bigger.

 ISBN: 978-1-61208-246-2

Based only on the three last graphs it can be said that because of the shortest optimizing time, the *EML* is supposed to return better solution than two others algorithms which is reflected in the previous graphs.

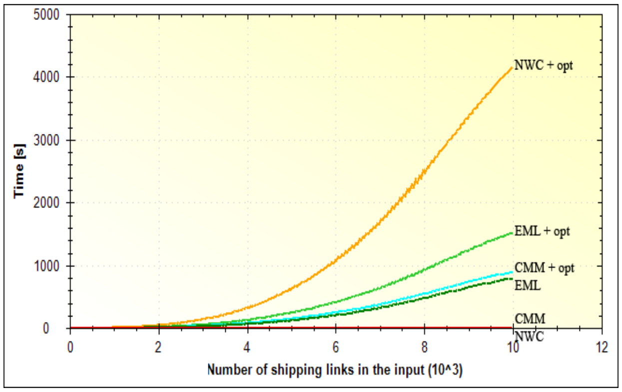

Figure 7. Processing time of the algorithms depending on the size of the input.

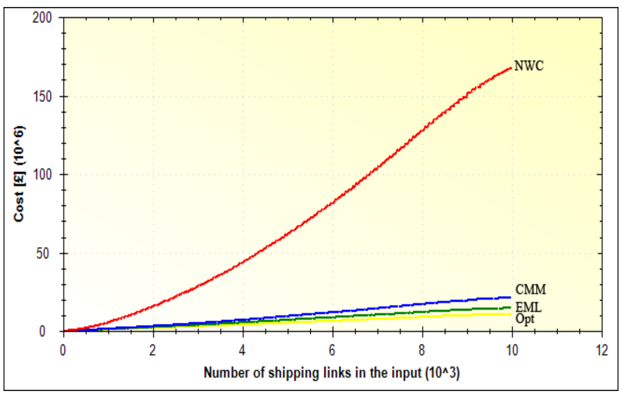

Figure 8. Cost of the solution found by algorithms without optimization depending on the size of the input.

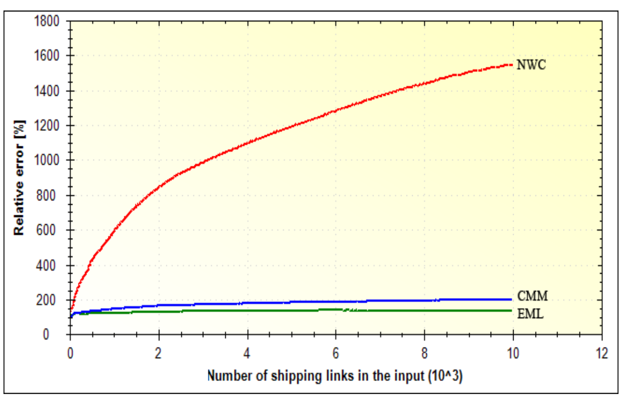

Figure 9. Relative error of solutions found by algorithms without optimization depending on the size of the input.

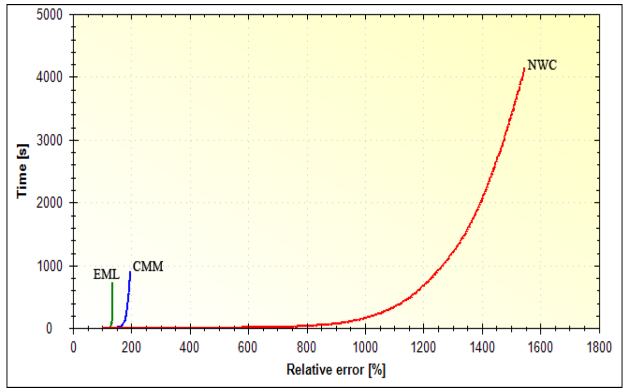

Figure 10. Time needed to improve the result depending on the relative error of the input solution.

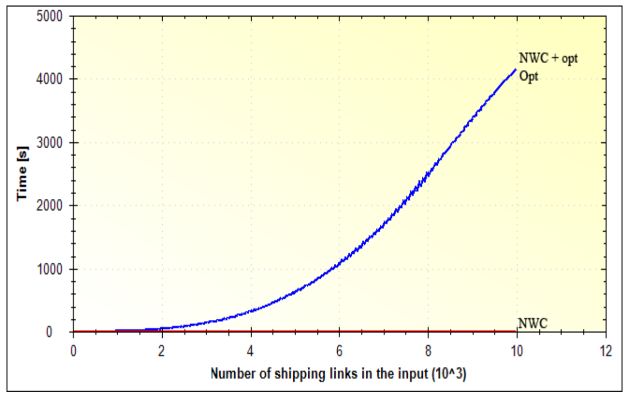

Figure 11. Time needed by the components of *NWC* with optimization algorithm depending on the size of the input.

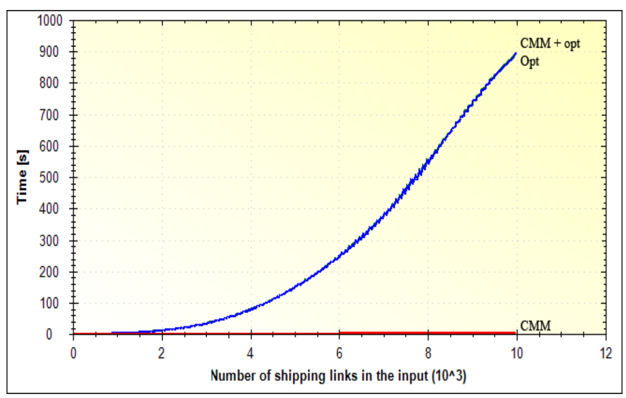

Figure 12. Time needed by the components of *CMM* with optimization algorithm depending on the size of the input.

V. CONCLUSION AND FUTURE WORK

All of the objectives of the research have been completed. The results are gathered as the graphs providing a simple, fast and clear way of evaluation main qualities of the transportation problem algorithms.

 ISBN: 978-1-61208-246-2

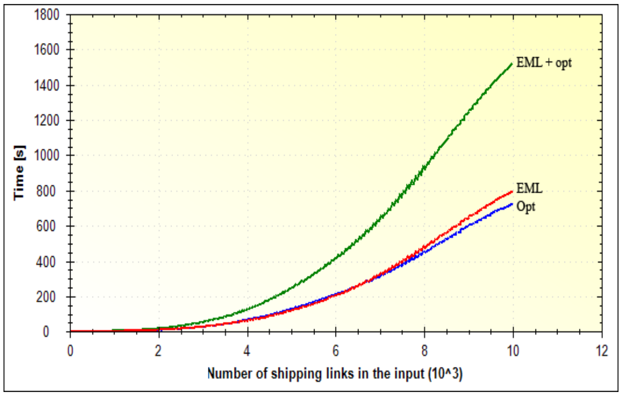

Figure 13. Time needed by the components of *EML* with optimization algorithm depending on the size of the input.

The accuracy of tested algorithms without optimization is different and should be considered while attempting to solve any given problem. On the other hand the optimized solutions take much time to be calculated.

When it comes to getting the optimal solution the *CMM* is the best algorithm to get the initial solution and optimize it. This method combined with the optimizing algorithm provides the quickest way of getting the optimal solution.

When calculating the minimum cost is remarkable time-consuming then the better solution is to choose one of two heuristic algorithms without optimization. They provide a quick way of obtaining a result much more accurate than any random solution such as the one returned by the *NWC*.

The quickest one is the *CMM* which is quite common and well known. The *EML* was an authorial idea and proved to be a great way of balancing between short time of calculations and small total cost returned. It works faster than optimized *CMM* algorithm but ends with a result nearly two times bigger.

In the future work we would like to test the influence of the ‘shape’ of the input data on the results and implement some additional functionality in the testing application allowing user to get solutions for more complicated versions of the problem. The most important are:

- Non-balanced problem [10],
- Costs of storage / shortage,
- Costs of production,
- Blockage of a shipment link [11],
- Partial blockage of a shipment link [12],
- Transshipment points.

All of the above can be easily transformed to a typical transportation problem. All actions needed to be taken are actions on the input matrix of costs and supplies / demands vectors. They do not make any of the algorithms work in other way and do not require any special treatment other than described preparation. This is very important from the point of view of future functionalities of the application that are planned to be implemented.

The application can be used in the real system providing a way of getting the solution to the problems in the field of transport, network traffic etc. The most profitable method of calculations can be chosen thanks to the tests results. It is significant in the cases where a lot of problems must be solved in a limited time (e.g. in computer networks flow control).

References

[1] G. B. Dantzig, and M. N. Thapa, Linear Programming 2: Theory and Extensions, Springer, New York, 2003.

[2] A. Calczynski, J. Sochanska, and W. Szczepankiewicz, Methods of shipment rationalization in trade /in Polish/, WAE, Cracow, 1988.

[3] A. Calczynski, Optimization methods in transport services market /in Polish/, PWE, Warszawa, 1992.

[4] G. Sharma, S. H. Abbas, and V. K. Gupta, “Solving transportation problem with the various methods of linear programming problem,” Asian Journal of Current Engineering and Maths, vol.1, no. 3, 2012, pp. 81-83.

[5] S. K. Kumar, I. B. Lal, and S. B. Lal, “Fixed – charge Bi-criterion Transportation Problem,” International Journal of Computer Application, vol. 1, 2012.

[6] D. Dutta and A. S. Murthy, “Fuzzy transportation problem with additional restrictions,” ARPN Journal of Engineering and Applied Sciences, vol. 5, no. 2, 2010.

[7] P. Pandian and G. Natarajan, “A new method for solving bottleneck-cost transportation problems,” International Mathematical Forum, vol. 6, no. 10, 2011, pp.451-460.

[8] F. S. Hillier and G. J. Lieberman, Introduction to operations research, McGraw Hill, New York, 2001.

[9] G. B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, 1963.

[10] A. Marczuk and W. Misztal, “Agricultural produce transport optimisation in the conditions of market non-balance,” /in Polish/, Journal Inzynieria Rolnicza, no 129, Cracow, 2011, pp. 221-226.

[11] K. Pienkosz, “Optimization models and methods of resource allocation” /in Polish/. Scientific Reports Series Elektronika, Warsaw University of Technology, no. 3-132. Warsaw, 2010.

[12] A. Marczuk, “A computer system for optimisation of soft fruit transportation in diffused purchasing networks,” Journal Eksploatacja i Niezawodnosc, vol. 44, no. 4, Warsaw, 2009.

 ISBN: 978-1-61208-246-2