Evaluation of Heuristic Algorithms for Solving a Transportation Problem

Kacper Rychard, Iwona Pozniak-Koszalka, Leszek Koszalka, and Andrzej Kasprzak
Dept. of Systems and Computer Networks
Wroclaw University of Technology
Wroclaw, Poland
e-mail: krychard@gmail.com, {iwona.pozniak-koszalka, leszek.koszalka, andrzej.kasprzak}@pwr.edu.pl

Dawid Zydek
Dept. of Electrical and Computer Engineering
University of Nevada Las Vegas
Las Vegas, U.S.A
e-mail: dawid.zydek@gmail.com

Abstract -- **This paper concerns different approaches to solve a transportation problem. A new idea for solving the formulated problem is developed. Three algorithms, named Highest Cost Method (HCM), Reverse Vogel's Approximation Method (RVAM), and Reverse Russel's Approximation Method (RRAM), have been created. The properties of these algorithms, including the accuracy and the efficiency, are evaluated on the basis of the simulations made using the designed and implemented experimentation system. Moreover, the paper contains the results of the comparison between known algorithms and the proposed algorithms. The comprehensive studies show that the proposed algorithms are more accurate; however, they require more processing time to find the solution.**

Keywords- transportation problem; algorithm; heuristic; cost reduction; experimentation system.

I. INTRODUCTION

The transportation problem is a well-known issue faced by majority of companies. Transportation is usually the main component of the company's logistics budget [1]. Ineffective transport generates unnecessary costs that can lead to wasting large amounts of money in a scale of a whole company. Even the largest companies take sometimes peculiar actions, e.g., avoiding the left turns in routes of their delivery trucks to reduce the total costs of transportation [2]. Essentially, the main problem is how to move goods from group of m sources to n destinations in a way that minimizes the total transportation cost [3]. As the pace of both industrial and economic development was increasing, more and more goods started to be transported. These changes include an increase in the need for transportation, new types of transported goods, and new ways of transporting them. At some point, the task of cost control in such a system has become too difficult to be performed without specialized tools.

Increasing attention in Internet and network services has created new categories of transportation systems in order to determine traffic of many different applications, e.g., P2P multicast or Content Delivery Networks [4]. These new adoptions of the problem define new requirements for approaches used to solve the problem. First of all, the solution often has to be obtained quickly as the time is the key factor affecting the quality of service and user experience. Moreover, for more complex problems with great number of sources and destinations, obtaining the optimal solution is often not possible in a reasonable amount of time. Because of that, heuristic algorithms combining time efficiency and capability of obtaining a close to the optimal solution need to be used. However, the accuracy of these algorithms is usually ensured at the expense of additional calculations. This creates the challenge of balancing between the short processing time and precise calculations when finding a way to solve the transportation problem. This paper is an extension of our work in [5]. We propose three new algorithms:

- Highest Cost Method (HCM),
- Reverse Vogel's Approximation Method (RVAM),
- Reverse Russel's Approximation Method (RRAM).

The rest of the paper is organized as follows. Section II presents the related work. Section III describes the transportation problem's mathematical model and its most popular representations. Section IV includes a description of the main algorithms and the most important pros and cons of their use. A new approach with the proposed algorithms is described in Section V. Section VI and Section VII contain the design of the experiments, next, their results, followed by comments. The developed tool and its use are also presented. The paper is concluded in Section VIII followed by the plans for the future work.

II. RELATED WORK

In [6], a fuzzy version of the transportation problem with additional restrictions is examined. The fuzzy transportation problem is characterized by fuzzy intervals as the unit costs of the shipment links. In this paper, the problem was transformed into the classical linear fractional programming problem presented in [7]. A time minimization in a fuzzy version of the problem is a subject of the research - in [8], the authors present a procedure to obtain an optimal solution which provides the longest time on active transportation routes as well. A numerical example is included to validate the presented approach. The problem with uncertain cost, supplies, and demands rather than fuzzy variables is studied in [9], where the authors discuss the possibility of transformation of the problem into the deterministic form. Problems

 ISBN: 978-1-61208-432-9

that take into account both cost and time are presented in [10]. However, this work is focused on finding new ways of solving cost transportation problems - two new methods (blocking method and blocking zero point method) are proposed. While most of the methods solving the transportation problem focus on minimizing only one factor, the work in [11] solves the problem in such a way that both cost and time are minimized. In [12], the novel Artificial Immune Algorithm is presented to solve the Fixed-Charge Transportation Problem. In this modification of the original transportation problem, the total cost of transportation depends on the unit costs and on additional cost associated with the link use. The authors compare their work to most recent methods [13] sowing that the proposed procedure is superior to them. An ant colony optimization algorithm is presented in [14] as well as an approach which uses both genetic algorithm and local search in solving multi-objective transportation problem. The paper considers the problem with a cross-docking network. The proposed algorithms reduce the total cost in some type of transportation networks and perform better than Branch-and-Bound method [15]. The authors emphasized the importance of heuristic algorithms, in particular hybrid evolutionary algorithms in optimization problems.

III. TRANSPORTATION PROBLEM STATEMENT

The main assumptions of the problem are that the cost of transportation between a given source and destination depends on the quantity of goods transported (all the unit costs are known) and the acceptable solution is the one that exhausts supplies of all sources and fulfills demands of all destinations without the negative values of allocations [16]. The considered problem is formulated as a set of formulas:

$$C(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = s_i \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \tag{3}$$

$$x_{ij} \geq 0 \qquad i = 1,2,...,m \qquad j = 1,2,...,n \tag{4}$$

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j \tag{5}$$

The above expressions are described in the following way: 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 (1). The total amount of goods sent from each source should be equal to its supply, where s_i are the sources' supplies (2). The total amount of goods sent to each destination should be equal to its demand - d_i are the destinations' demands (3). All allocations should be non-negative (4). In the balanced problems, the equation (5) states that the sum of all supplies s_i equals the sum of all demands d_j, which means that there is a solution exhausting all sources' supplies and fulfilling all destinations' demands.

A graph representation of the problem is shown in Fig. 1. The sources and destinations are represented by circles; they are denoted by s_i and d_i stand for the sources' supplies and destinations' demands, respectively. The arrows represent the shipping links and the numbers placed on them are the unit costs [16].

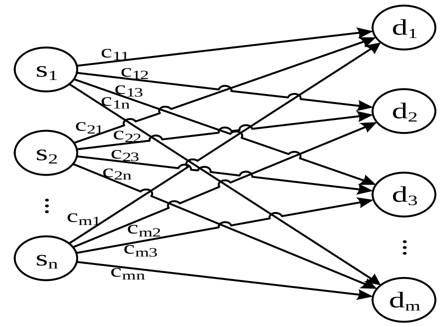

Figure 1. Graph representation of the transportation problem.

The same problem may be illustrated by Table I.

TABLE I. MATRIX REPRESENTATION OF THE PROBLEM

		d_1	d_2	d_3	$\cdots$	d_n
s_1		c_{11}	c_{12}	c_{13}	$\cdots$	c_{1n}
s_2		c_{21}	c_{22}	c_{23}	$\cdots$	c_{2n}
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
s_m		c_{m1}	c_{m2}	c_{m3}	$\cdots$	c_{mn}

The table shows the matrix of costs (c_{ij}) and two vectors representing the supplies of the sources (s_i) and the demands of the destinations *(*d_j).

IV. KNOWN ALGORITHMS

The most common algorithms for solving transportation problems are based on the triangularity rule [3] saying that the feasible solution is obtained after the operations:

Step 1. If the problem consists of only one source or destination, set all the amounts of transportation to highest possible. Go to STOP.

Step 2. For the next link *xij* to be considered, set the amount of transportation to highest possible. Reduce the problem using *i*-th source or *j*-th destination.

Step 3. Go to step 1.

The maximum possible amount of transportation in each shipment link is calculated as the smaller number from the two numbers of: supply of the source and demand of the destination linked. It is noteworthy that the main idea of the algorithm remains all the time the same; however, the technique sometimes is described differently. The changes concern a number of steps and, what is more important, the possibility of deleting the source and destination in one step. This action results in obtaining the solution that is degenerated (uses less than $m+n-1$ shipment links). Deleting only one source/destination at the time leads to allocations with the value of 0 in the solution, which does not increase the cost of the solution found, but allows using the solution as an input to optimization algorithm.

North-West Corner Rule (NCR). In this algorithm, the links are considered in a sequence as they appear in the problem matrix. This method is supposed to provide a fast way of achieving a feasible, but not necessarily efficient solution [17]. The step list of the NCR algorithm is:

Step 1. If the problem consists of only one source or destination, set all the amounts of transportation to highest possible. Go to STOP.

Step 2. For the North-West link (top left element in the matrix of costs) x_{ij}:

a. If $s_i > d_j$ (supply higher than demand), then allocate *dj* to this link. Decrease the *si (i*-th source's supply) by d_j. Delete the *j*-th destination.

b. If $s_i < d_j$ (demand higher than supply), then allocate s_i to this link. Decrease the d_j (*j*-th destination's demand) by s_i. Delete *i*-th source.

c. If $s_i = d_j$ (supply equals demand), then choose randomly action of 2a or 2b.

Step 3. Go to step 1.

The main advantage of this approach is that not all of the shipment links are considered by the algorithm. Once the source/destination is deleted, all other links leading from/to node are omitted.

Lowest Cost Method (LCM). The main idea behind LCM is to sort the connections by the unit cost *cij* and use the cheapest ones first [3]. However, it is unlikely to use this method to provide the final solution. The idea of using the LCM as a heuristic algorithm is novel. The main advantages of this approach are the simplicity, quickness, and way better solution than the NCR. The solution returned by this algorithm meets all the main requirements (exhausting all sources' supplies *si* and fulfilling all destinations' demands *dj* without negative amount of transportation) and is supposed to be closer the optimal solution than the output returned by the NCR. The step list of the LCM algorithm is almost identical to the NCR. The only difference is in the order of the links considered in step 3.

Vogel's Approximation Method (VAM). The VAM is in some sense an extension of the LCM. According to this algorithm, the unit cost of the link is not the only determinant of its position in a sequence. More important is the difference between the lowest unit cost *cij* in a row/column in the matrix of costs and the second smallest one. As the next link to be considered, the one with the lowest unit cost *cij* in the chosen row/column is selected [3]. The process of the link selection in the VAM is as follows: (i) For each row and column calculate the difference between two lowest values of the unit costs in this row/column; (ii) Select the row/column with the highest value of calculated difference; (iii) Consider the link with the lowest unit cost in the selected row/column as a next.

Russel's Approximation Method (RAM). The RAM, like the VAM, depends on the calculations made on the matrix of costs while choosing the next link to consider. In each step the maximum costs in each i-th row and j-th column are found. The assist value (γ_{ij}) is calculated to determine the next link. The link with the lowest γ_{ij} value is selected.

Optimization algorithm. This 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 and finds the non-used connection that should be used to reduce the total cost of transportation (if the solution was not optimal). Adding the connection to the solution may increase or decrease other allocations. Next, the described steps are repeated. The algorithm stops when the solution is optimal. The number of iterations done varies and depends on the input solution – mostly on its accuracy. The detailed description of how the algorithm works can be found in [3].

V. THE PROPOSED ALGORITHMS: HCM, RVAM, RRAM.

The main idea to design the proposed algorithms was based on the approach that is opposite to the LCM. If it is possible to use the cheapest links to transport goods, avoiding the use of links with highest unit cost should result in a similar solution.

The main problem of this approach is that the calculations of the minimum allocations assume that the values of the supplies and demands will not change and it will be possible to exhaust current supply/fulfill demand elsewhere in further steps of the algorithm. When some allocations are made, the previous assumptions cease to be valid. This is why the algorithm calls itself with some amounts of transportation pre-allocated and the corresponding supplies and demands decreased [5]. In every step, the minimum allocation is calculated as:

$$x_{ij\min} = \max\{s_i - d'_j, d_j - s'_i, 0\}, \quad s'_i = \sum_{\substack{k \neq i \\ e_{kj}=0}} s_k, \quad d'_j = \sum_{\substack{l \neq j \\ e_{il}=0}} d_l \tag{6}$$

In (6), the s'_i stands for the sum of all the unconsidered supplies in this run of the algorithm except *i*-th; the d'_j stands for all unconsidered demands in this run of the algorithm except *j*-th; the e_{ij} is a variable responsible for determining whether the link between *i*-th source and *j*-th destination was considered in this run of the algorithm (1 if it is true, 0 otherwise). When a given source has enough other destinations to send goods to, and a given destination has enough other sources to receive goods from, both $s_i - d'_j$ and $d_j - s'_i$ are less than 0 and x_{ij} min gets the value 0. To improve algorithm's accuracy, the case when $s_i - d'_j = 0$ or $d_j - s'_i = 0$ is distinguished. When an allocation due to the source is made, the amounts of transportation in all the other unconsidered links of this source are set to their maximum (calculating the minimum in the first link was based on the assumption of allocating the maximum in all the other links). The source and the destinations for which the maximum allocations were made are deleted and the algorithm repeats. If an allocation is caused both by the source and destination, only one of the above action chains is taken. The step list is as follows:

Step 1. If the problem consists of only one source or destination, set all the amounts of transportation to highest possible. Go to STOP.

Step 2. For the next link x_{ij} is to be considered:

(a) If $s_i - d'_j < 0$ and $d_j - s'_i < 0$, then mark link as considered. Go to step 3.

(b) If $s_i - d'_j \geq 0$ or $d_j - s'_i \geq 0$, then choose the case:

(*i*) If $s_i - d'_j > d_j - s'_i$ (allocation due to the source), then allocate $s_i - d'_j$ in this link and maximum in all of the other unconsidered links of this source. Delete *i*-th source and the destinations for which the maximum allocations were made.

(*ii*) If $s_i - d'_j < d_j - s'_i$ (allocation because of the destination), then allocate $dj - s'i$ in this link and maximum in all of the other unconsidered links of this destination. Erase the *j*-th destination and the sources for which the maximum allocations were made.

(*iii*) If $s_i - d'_j = d_j - s'_i$, then choose randomly action from step 2b(*i*) or 2b(*ii*).

Step 3. Go to step 1.

The presented idea may be also used for creating reverse versions of VAM and RAM - every single allocation results with deleting one source or destination before the algorithm continues until the only source or destination is preserved. Then, the remaining allocations are made. The total number of links used in the returned solution equals $m + n - 1$. It is the exact number of the used links in the solutions obtained by the methods based on the triangularity rule.

Highest Cost Method (HCM). This algorithm is a developed version of Expensive Means Less (EML), which was presented by the authors in [5]. The HCM is based on the main proposed idea of avoiding allocations on links with high unit cost. In this algorithm, the most expensive links are considered first. It is supposed to return solutions with cost similar to those of the LCM, but recursive calls may cause increase of its overall runtime. The only factor determining the order of the considered links is their unit cost. This algorithm, as well as the LCM, is characterized by kind of 'shortsightedness' - it does not take into account the consequences of made decisions, e.g., avoiding one expensive link may cause the necessity of the use a few others later one, leading to an increase in the overall cost of the solution.

Reverse Vogel's Approximation Method (RVAM). The RVAM is based on the VAM algorithm, but it uses different priorities during determining the next link to be considered. While the VAM selects the minimum element of the row/column of the cost matrix with the biggest difference between two smallest elements, the RVAM chooses the maximum element of the row/column with the biggest difference between two most expensive links. This can be interpreted as seeking the link which, if avoided, would potentially prevent the cost to increase the most. The step list of the link selection process in the RVAM is as follows: (i) For each row and column calculate the difference between two highest values of the unit costs in this row/column; (ii) Select the row/column with the highest value of the calculated difference; (iii) Select the link with the highest unit cost in the selected row/column; (iv) Consider this link as next.

Reverse Russel's Approximation Method (RRAM). As in the case of the VAM and the RVAM, the RRAM combines the original RAM approach with the proposed idea. To select the next link to be considered, the assist value γij is calculated. The link with minimum γij is to be chosen.

VI. EXPERIMENTS

The objective was to test the efficiency and accuracy of the implemented algorithms. The testing tool is 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.

Experimentation system. The implemented testing tool allows the user to select the range of the input data. The final solution is an average of the tests' results. As for the number of goods transported parameter, the user is allowed to input the average supply and the demand is calculated to balance the problem. As the number of sources (*m*) and destinations (*n*) varies during the tests, the problem of maximum size is generated and on each step of the test it is reduced to proper size. Then, the problem is balanced by increasing the supply of the last source (*sm*) or demand of last destination (*dn*) accordingly.

The tests were designed to deliver the information about the main characteristics of the implemented algorithm, which are processing time and cost found. To allow a 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. The experimentation system may be regarded as input-output system (the block-diagram is shown in Fig. 2).

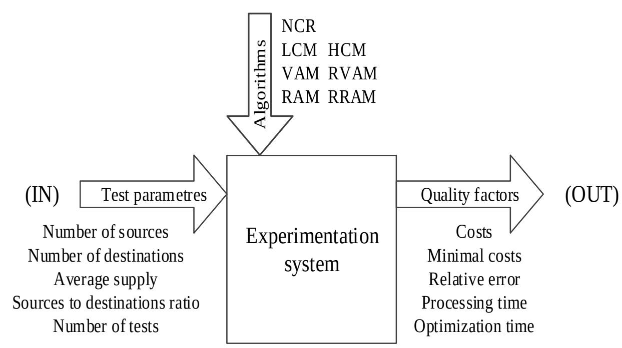

Figure 2. Experimentation system as input-output plant.

Experiments were conducted in order to investigate:

(*i*) Cost of the solution found by algorithms without optimization in comparison to the optimal one (depending on the size of the input);

(*ii*) Relative error of solutions found by algorithms without optimization (depending on the size of the input); (*iii*) Processing time of the algorithms (depending on the size of the input).

All the experiments were made with the number of tests (single experiments) set to 10.

Preliminary experiment. The experiment consisted in testing the total cost expressed by (1) obtained by the algorithms depending on the size of the problem (m x n) defined by the data matrix (see Table 1). For all algorithms, the

 ISBN: 978-1-61208-432-9

same matrices (the same size of the problem) were tested, e.g., 4 × 6, 6 × 4, 2 × 12, etc.

For any matrix, single experiments were repeated and the averaged values were treated as the results of the experiments. In Fig. 3, NCR and LCM solutions as well as the optimal cost (marked as the OPT) are presented.

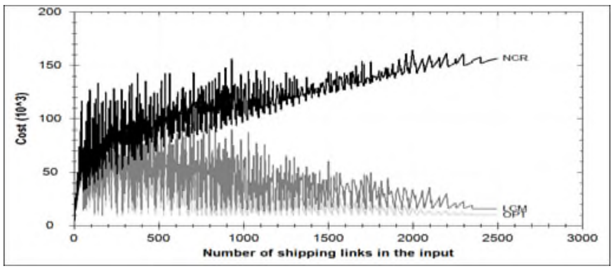

Figure 3. Preliminary experiment results.

It may be observed that NCR algorithm is about eight times less accurate than LCM. Therefore, NCR was excluded from further experiments. In the next experiments, the data in a range from 1 × 1 to 50 × 50 with the same number of sources and destinations *(m = n)* were taken into consideration.

VII. ANALYSIS OF THE RESULTS OF EXPERIMENTS

A. Cost of the Solution

The experiment was designed for finding the relationship between the cost expressed by (1) and produced by the known and proposed algorithms and the size of the input measured by the number of shipping links. The results are shown in Fig. 4.

(a) (b)

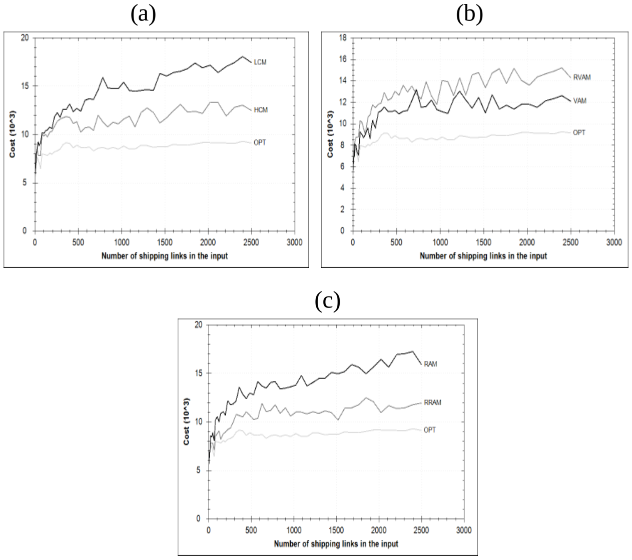

Figure 4. Cost comparison: (a) LCM with HCM, (b) VAM with RVAM, (c) RAM with RRAM.

Two of the proposed algorithms (the HCM and the RRAM) returned the solutions better than their known equivalents. The HCM outperforms the LCM by approximately 29%. The RRAM returns solutions that are cheaper than the ones returned by the RAM by approximately 25%. In case of the VAM and the RVAM, the proposed algorithm is not as good as the original one. The solutions of the VAM are about 15% cheaper than those found by the RVAM.

B. Relative Error

To determine the accuracy of the algorithms, the relative error of the returned solutions was examined as well. It was calculated as the ratio of the cost of the solution found to the minimal possible one (i.e., the cost of the optimal solution). The results are shown in Fig. 5.

(a) (b)

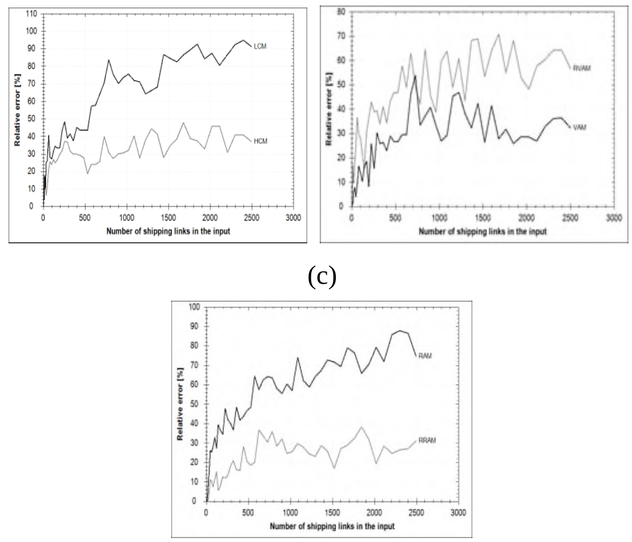

Figure 5. Relative error comparison: (a) *LCM* with *HCM, (b) VAM* with *RVAM, (c) RAM* with *RRAM.*

It may be observed that HCM and RRAM are found to be remarkably more accurate than LCM and RAM by 43% and 34%, respectively. In opposite, RVAM performed worse than VAM by 24 %. It is worth to notice that the relative error of all tested algorithms was correlated to the size of the problem.

C. Processing Time

Two cases were considered: (i) The time of finding solution without optimization (called as algorithm alone), (ii) The time spent on performing optimization procedure. The results are shown in Fig. 6.

(a) (b)

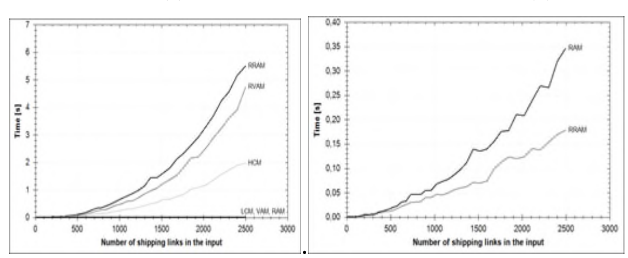

Figure 6. Processing time comparison: (a) algorithms alone, (b) algorithms with optimization.

The comparison of the total processing times needed by algorithms for finding the solution with optimization is given in Fig. 7.

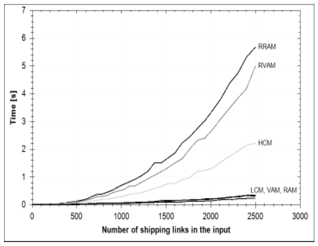

Figure 7. Processing time comparison – the total processing

The created algorithms take more time to calculate the solution. It can be explained by the recursive calls present in the proposed approach. The optimization based on the HCM and the RRAM solutions runs faster than the cases of the LCM and the RAM. However, what was expected, the total processing time is shorter for the classical algorithms.

VIII. CONCLUSION AND FUTURE WORK

In the paper, three created algorithms for solving the transportation problem were presented, evaluated, and compared with the existing ones. The HCM and RRAM found better solutions than their literature equivalents. The solutions closest to the optimal one were obtained by RRAM. However, from the processing time point of view, the proposed approach is more time consuming. Reasons of this fact should be sought in a more complex way of calculating, i.e., in recursive calls. The fastest is LCM, which sorts the shipment links considering their unit cost only.

In general, we may conclude that, if the processing time of the HCM and the RRAM satisfies the requirements of the logistic systems, they can be used to provide a solution more accurate than those of known algorithms.

As the future work, the influence of the 'shape' of the problem (number of sources *m* to number of destinations *n* ratio) on the results is to be tested, and extensions of the proposed algorithms to some modifications of the problem, such as fixed charge transportation, will be considered. The most important are: (i) non-balanced problem [18], (ii) costs of storage/shortage, (iii) blockage of a shipment link [19]. Moreover, it is planned to give opportunities for making tests in automatic way [20], and to apply the idea of the multistage experimentation [21].

ACKNOWLEDGMENT

This work was supported by the statutory funds of the Dept. of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Technology, Poland.

REFERENCES

[1] M. Morrison, "Optimize your transportation program for greater efficiency", [online] Logitics.com/digital/issues/digital.pdf [retrieved: March, 2015].

[2] J. Lovell, "Left-Hand-Turn Elimination," article in: The New York Times Magazine, Dec 9, 2007.

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

[4] C. Huang, N. Holt, Y. A. Wang, A. Greenberg, J. Li, and K.W. Ross, "A DNS reflection method for global traffic management", Proc. USENIX Annual Techn. Conference, Boston, 2010, pp. 265-270.

[5] K. Rychard, W. Kmiecik, L. Koszalka, and A. Kasprzak, "Experimentation system for heuristic algorithms to solving transportation problem", Proc. of 8th Intern. Conf. on Systems (ICN'13), IARIA, Seville, 2013, pp. 27-32.

[6] D.Dutta and A. S. Murthy, " Fuzzy transportation problem with additional restrictions", ARPN Journal of Engineering and Applied Sciences, vol. 5, 2010, pp. 36-40.

[7] 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, 2012, pp. 81-83.

[8] J. M. Vinotha, W. Ritha, and A. Abraham, "Total time minimization of fuzzy transportation problem", Journal of Intelligent & Fuzzy Systems, vol. 23, 2012, pp. 93-99.

[9] Y. Sheng and K. Yao, "A transportation model with uncertain costs and demands", International Interdisciplinary Journal, vol. 15, 2012, pp. 3179-3186.

[10] P. Pandian and G. Natarajan, "A new method for solving bottleneck-cost transportation problems," International Math. Forum, vol. 6, 2011, pp. 451-460.

[11] S. K. Kumar, I. B. Lai, and S. B. Lai, "Fixed – charge Bi-criterion transportation problem", International Journal on Computer Applications, vol. 2, 2012, pp. 39-42.

[12] K. M. Altassan, M. M. El-Sherbiny, Y. M. Ibrahim, and A. D. Abid, "A novel artificial immune algorithm for solving fixed charge transportation problems", Proc. of Conf. on Computer Science and Inform. Technology, vol. 36, 2013, pp. 114-121.

[13] M. Hajiaghaei-Keshtelia, S. Molla-Alizadeh-Zavardehib, and R. Tavakkoli-Moghaddama, "Addressing a nonlinear fixed-charge transportation problem using a spanning tree-based genetic algorithm", Computers & Industrial Engineering, vol. 55, 2010, pp. 259-271.

[14] R. Musa, J. P. Arnaout, and H. Jung, "Ant colony optimization algorithm to solve the transportation problem of cross-docking network", Computers & Industrial Engineering, vol. 59, 2010, pp. 85-92.

[15] S. A. Zaki, A. A. Mousa, H. M. Geneedi, and A. Y. Elmekawy, "Efficient multiobjective genetic algorithm for solving transportation, and transshipment problems", Appl. Math.. vol. 3, 2012, pp. 92-99.

[16] A. Calczynski, J. Sochanska, and W. Szczepankiewicz, Methods of Shipment Rationalization in Trade, /in Polish/, WAE, Cracow, 1998.

[17] F. S. Hillier and G. J. Lieberman, Introduction to Operations Research, McGraw Hill, New York, 2012.

[18] A. Marczuk and W. Misztal, " Agricultural produce transport optimization in the conditions of market non-balance" /in Polish/, Journal Inz. Rolnicza, vol. 129, 2012, pp. 221-226.

[19] K. Pienkosz, " Optimization models and methods of resource allocation," /in Polish/, Scientific Reports Series Elektronics, Warsaw University of Technology, vol. 3, 2010, pp. 124-132.

[20] D. Ohia, L. Koszalka, and A. Kasprzak, "Evolutionary algorithm for solving congestion problem in computer network", Lecture Notes in Computer Science, Springer, vol. 5711, 2009, pp. 112-121.

[21] L. Koszalka, D. Lisowski, and I. Pozniak-Koszalka, "Comparison of allocation algorithms for mesh structured networks using multistage simulation", Lecture Notes in Computer Science, Springer, vol. 3984, 2006, pp. 58-67.