Computer Science Journal of Moldova, vol.**30**, no.1(88), 2022

# Solving the non-linear multi-index transportation problems with genetic algorithms

Tatiana Pașa

**Abstract**

In this paper we study the non-linear multi-index transportation problem with concave cost functions. We solved the non-linear transportation problem on a network with 5 indices (NTPN5I) described by sources, destinations, intermediate nodes, types of products, and types of transport, that is formulated as a non-linear transportation problem on a network with 3 indices (NTPN3I) described by arcs, types of products, and types of transport. We propose a genetic algorithm for solving the large-scale problems in reasonable amount of time, which was proven by the various tests shown in this paper. The convergence theorem of the algorithm is formulated and proved. The algorithm was implemented in Wolfram Language and tested in Wolfram Mathematica.

**Keywords:** non-linear programming, concave function, transport problem, index.

**MSC 2010:** 68W25, 68W50, 90B06, 90C06, 90C08, 90C26, 90C30, 90C35, 90C59.

## 1 Introduction

Nowadays, the transportation problem is relevant, because of the globalization and the fact that people tend to buy products from different parts of the world, and the price depends on several factors that must be taken into account. The minimum cost of transporting products varies depending on the types of products (construction materials, food, petroleum products), the number and types of transport used [1]

(train, ship, truck, plane), the number of sources and destinations of the transportation network, and also on the transportation route (air, water, roads, pipes, cables) chosen by the carrier. To model these dependencies, transportation costs described by non-linear functions are associated with the paths between intermediate nodes in the transportation network.

The transportation problems on networks with concave cost functions with 5 indices, described by sources, destinations, intermediary nodes, types of products, and types of transport, are formulated and solved in this paper. In order to simplify the mathematical model of the problem, it is formulated as a non-linear transportation problem with 3 indices described by arcs, products, and transport types. Therefore, it is required to minimize the costs of transportation from sources to destinations of several types of products using several types of transport. Results that refer to the solution of the non-linear transportation problem with 4 indices with genetic algorithms are presented in the works [2]–[4]. In [5], the heuristic algorithms to the solution of the non-linear transportation problem with 4 and 5 indices are presented.

This problem is part of the group of transportation problems with $n$ indices. The general case of TPnI was formulated by Ph.-X. Ninh [6], who also proposed an exact method of solving, an extension of the method of potentials by coordinating the solving of the primary and dual problem. An important result is the formulation and proof of the theorem of the necessary and sufficient condition of the existence of the solution. Among the authors who studied the multi-index transportation problem are K. B. Hayley [7], who presented an algorithm, and W. Junginer [8], who conducted a study on the characteristics of the multi-index problem and proposed a set of logical problems for solving transportation problems with several indices.

Solving this problem is difficult, because of the existence of several extremas and the impossibility of differentiating a global minimum from a local one. Because this problem belongs to the class of NP-hard problems, there are no polynomial algorithms to determine the global solution. Heuristic algorithms can be used to solve this problem, in particular, genetic algorithms that allow the solution of the problem to

be generated in a reasonable amount of time, even for large-scale problems. The use of these algorithms in favor of other heuristic algorithms is recommended, because they do not need the gradient or Hessian information. They are also resistant to locks in a local minimum and can be used to solve large-scale non-linear optimization problems.

When a genetic algorithm is applied, it will take into account the fact that each chromosome obtained at the generation of the initial population and after the application of the crossover or mutation operator must correspond to a single admissible solution in the codomain. Another aspect that must be considered is the number of iterations after which the algorithm is stopped and the solution associated with the chromosome with the best characteristics is accepted as the optimal solution of the formulated problem. A large number of iterations means that the generated solution is very close to the global optimum, but in the case of large-scale problems, the large number of iterations also means longer execution time. For this reason, a balance must be found that would allow the solution of the problem to be obtained in a reasonable amount of time, even for large-scale problems.

## 2 Problem formulation. Main results

In the following, we consider the non-linear transportation problem described by: $n$ sources $A_1, A_2, ..., A_n$ with the respective product capacities $\alpha_1, \alpha_2, ..., \alpha_n$; $m$ destinations $B_1, B_2, ..., B_m$ with the respective product capacities $\beta_1, \beta_2, ..., \beta_m$; $p$ product types $P_1, P_2, ..., P_p$ of respective quantities $\gamma_1, \gamma_2, ..., \gamma_p$; $q$ transport types $T_1, T_2, ..., T_q$ with the respective capacities $\delta_1, \delta_2, ..., \delta_q$; $g$ intermediate nodes $v_r \in V_{int}$, where neither production nor consumption of the flow takes place. Each $e \in E$ arc of the transportation network is associated with a non-linear function that describes the transportation cost that depends on the volume of flow transported over that arc.

In this case, the transportation network is described by an acyclic oriented graph $G = (V, E)$, with the set of nodes $V = V_s \bigcup V_t \bigcup V_{int}$, where $V_s$ – the set of sources, $|V_s| = n$; $V_t$ – the set of destinations, $|V_t| = m$; $V_{int}$ – the set of intermediate nodes, $|V_{int}| = g$; $E$ – the

set of arcs, $|E| = u$. This problem is part of the group of non-linear transportation problems with 5 indices described by sources, destinations, intermediate nodes, product types, and types of transport that is formulated as a non-linear transportation problem on a network with 3 indices (NTPN3I) described by arcs, types of products, and types of transport.

In order to determine the minimum transportation cost, the restrictions on the conservation of the product flow in the network must be observed:

- The total quantity of product of the type $P_k$, $k = 1, 2, ..., p$ of quantity $\gamma_k$, $k = 1, 2, ..., p$ which comes out of all the network sources is described by the relation:

$$\sum_{e \in E^-(V_s)} \sum_{l=1}^{q} x_{ekl} = \gamma_k, k = 1, 2, ..., p, \tag{1}$$

  where $E^-(V_s)$ – the set of outgoing arcs of the network sources.

- The total quantity of transport of the type $T_l$, $l = 1, 2, ..., q$ with the capacity $\delta_l$, $l = 1, 2, ..., q$ coming out of all the network sources is described by the relation:

$$\sum_{e \in E^-(V_s)} \sum_{k=1}^{p} x_{ekl} = \delta_l, l = 1, 2, ..., q, \tag{2}$$

  where $E^-(V_s)$ – the set of outgoing arcs of the network sources.

- The total quantity of product coming out of each source $A_i$, $i = 1, 2, ..., n$ is $\alpha_i$, $i = 1, 2, ..., n$ and described by the relation:

$$\sum_{e \in E^-(s_i)} \sum_{k=1}^{p} \sum_{l=1}^{q} x_{ekl} = \alpha_i, i = 1, 2, ..., n, \tag{3}$$

  where $s_i \in V_s$ – source $i$ of the network, $E^-(s_i)$ – the set of outgoing arcs of the source $i$ of the network.

- The total quantity of product entering each destination $B_j$, $j = 1, 2, ..., m$ is $\beta_j$, $j = 1, 2, ..., m$ and described by the relation:

$$\sum_{e \in E^+(d_j)} \sum_{k=1}^{p} \sum_{l=1}^{q} x_{ekl} = \beta_j, j = 1, 2, ..., m, \tag{4}$$

  where $d_j \in V_t$ – the destination $j$ of the network, $E^+(d_j)$ – the set of incoming arcs of the destination $j$ of the network.

- The quantity of product $\gamma_k$, $k = 1, 2, ..., p$ that enters an intermediate node $v_r \in V_{int}$ is equal to the quantity of product that leaves that node and described by the relation:

$$\sum_{e \in E^+(v_r)} \sum_{l=1}^{q} x_{ekl} - \sum_{e \in E^-(v_r)} \sum_{l=1}^{q} x_{ekl} = 0, \begin{array}{l} r = 1, 2, ..., g, \\ k = 1, 2, ..., p, \end{array} \tag{5}$$

  where $v_r \in V_{int}$ – intermediate node of the network, $E^-(v_r)$ – the set of outgoing arcs of the node $v_r$, $E^+(v_r)$ – the set of incoming arcs of the node $v_r$.

- The transport quantity $\delta_l$, $l = 1, 2, ..., q$ entering an intermediate node $v_r \in V_{int}$ is equal to the transport quantity leaving that node and described by the relation:

$$\sum_{e \in E^+(v_r)} \sum_{k=1}^{p} x_{ekl} - \sum_{e \in E^-(v_r)} \sum_{k=1}^{p} x_{ekl} = 0, \begin{array}{l} r = 1, 2, ..., g, \\ l = 1, 2, ..., q, \end{array} \tag{6}$$

  where $v_r \in V_{int}$ – intermediate node of the network, $E^-(v_r)$ – the set of outgoing arcs of the node $v_r$, $E^+(v_r)$ – the set of incoming arcs of the node $v_r$.

- The values that describe the product flow on the network arcs are non-negative numbers:

$$x_{ekl} \geq 0, \forall (e, k, l), \tag{7}$$

NTPN3I consists in determining a flow $x^*$ that minimizes the cost function:

$$F(x) = \sum_{e\in E}\sum_{k=1}^{p}\sum_{l=1}^{q}\varphi_{ekl}(x_{ekl}), \tag{8}$$

where $\varphi_{ekl}(x_{ekl})$ are non-decreasing concave functions, so the solution of the following non-linear problem is required:

$$F(x) \longrightarrow \min \tag{9}$$

$$\begin{cases} \displaystyle\sum_{e\in E^-(V_s)}\sum_{l=1}^{q} x_{ekl} = \gamma_k, & k = 1,2,...,p, \\ \displaystyle\sum_{e\in E^-(V_s)}\sum_{k=1}^{p} x_{ekl} = \delta_l, & l = 1,2,...,q, \\ \displaystyle\sum_{e\in E^-(s_i)}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ekl} = \alpha_i, & i = 1,2,...,n, \\ \displaystyle\sum_{e\in E^+(d_j)}\sum_{k=1}^{p}\sum_{l=1}^{q} x_{ekl} = \beta_j, & j = 1,2,...,m, \\ \displaystyle\sum_{e\in E^+(v_r)}\sum_{l=1}^{q} x_{ekl} - \sum_{e\in E^-(v_r)}\sum_{l=1}^{q} x_{ekl} = 0, & r = 1,2,...,g, \; k = 1,2,...,p, \\ \displaystyle\sum_{e\in E^+(v_r)}\sum_{k=1}^{p} x_{ekl} - \sum_{e\in E^-(v_r)}\sum_{k=1}^{p} x_{ekl} = 0, & r = 1,2,...,g, \; l = 1,2,...,q, \\ x_{ekl} \geq 0, & \forall(e,k,l), \end{cases} \tag{10}$$

for which the non-negativity conditions are satisfied, so the quantities $\alpha_1, \alpha_2, ..., \alpha_n$, $\beta_1, \beta_2, ..., \beta_m$, $\gamma_1, \gamma_2, ..., \gamma_p$, and $\delta_1, \delta_2, ..., \delta_q$ are non-negative.

The general form of the optimal solution of NTPN3I described by (9)-(10) has the form $x^* = (x^*_{111}, x^*_{112}, ..., x^*_{upq})$ for a transportation network described by arcs. It is assumed that all data are real values and it is desired to obtain a feasible real value as the optimal solution.

## 2.1 Genetic algorithm MGA

In the following, the NTPN3I problem is solved, using a genetic algorithm, which involves coding the problem so that each chromosome in the population can be associated with only one admissible solution. Each crossover and mutation operator are described so that whenever they are applied to the chromosomes that describe the population, there is only one admissible solution that is associated with it after decoding. The notion of a spanning tree that correctly describes an admissible solution to the problem is used to code the admissible solutions of the problem. Each newly created population retains the chromosomes with the best characteristics from the previous population, so that during the selection the solutions associated with low costs are not lost.

Each chromosome is described by 5 compartments:

- the *first* contains spanning trees, each with the root in one of the sources of the transportation network;
- the *second* contains a permutation of the indices $i = 1, 2, ..., n$ of the sources and describes their service order;
- the *third* contains a permutation of the indices $j = 1, 2, ..., m$ of the destinations and describes their service order;
- the *fourth* contains a permutation of the indices $k = 1, 2, ..., p$ of the types of products and describes in what order they are transported;
- the *fifth* contains a permutation of the indices $l = 1, 2, ..., q$ of the types of transport and describes in what order they can be used.

The proposed ***MGA genetic algorithm*** for solving NTPN3I with concave cost functions is described by the following steps:

***Step 1.*** *Initialization.* The initial population is formed from $4|V|$ randomly generated chromosomes, each described by the set $P = \{\{T_r, r = 1, 2, ..., n\}, P_n, P_m, P_p, P_q\}$, where $T_r = \{(i, j) \,|i, j = 1, 2, ..., n + m + g\}$ – spanning trees with root at source $r = 1, 2, ..., n$ of the transportation network, $P_n = \{p(1), p(2), ..., p(n)\}$ – a permutation of

the indices $i = 1, 2, ..., n$ of the sources, $P_m = \{p(1), p(2), ..., p(m)\}$ – a permutation of the indices $j = 1, 2, ..., m$ of the destinations, $P_p = \{p(1), p(2), ..., p(p)\}$ – a permutation of the indices $k = 1, 2, ..., p$ of the types of products, $P_q = \{p(1), p(2), ..., p(q)\}$ – a permutation of the indices $l = 1, 2, ..., q$ of the types of transport.

***Step 2.*** *Decoding and Chromosome Evaluation.* Decoding assumes that each chromosome is associated with an admissible solution of the form $x = (x_{111}, x_{112}, ..., x_{epq})$, where the flow of $k$ products with $l$ types of transport from each source to destinations on the arcs corresponding to the spanning trees is transported. For this purpose:

1. One-dimensional arrays are created which contain respectively the quantities of products in sources, the product capacities in destinations, the types of products with their respective quantities and the types of transport with the respective capacities as follows: $\alpha' = [\alpha'_1, \alpha'_2, ..., \alpha'_n]$, $\alpha' = [\beta'_1, \beta'_2, ..., \beta'_m]$, $\gamma' = \left[\gamma'_1, \gamma'_2, ..., \gamma'_p\right]$, $\delta' = \left[\delta'_1, \delta'_2, ..., \delta'_q\right]$.

2. For each index $r \in P_{ij}$ the intermediate solution is determined as follows:

   - for each index $j \in P_m$, each index $k \in P_p$ and each index $l \in P_q$ is determined: $x_{ekl} = min\left\{\alpha'_i, \beta'_j, \gamma'_k, \delta'_l\right\}$, after which the update $\alpha'_i := \alpha'_i - x_{ekl}$, $\beta'_j := \beta'_j - x_{ekl}$, $\gamma'_k := \gamma'_k - x_{ekl}$, $\delta'_l := \delta'_l - x_{ekl}$ is applied, where $e$ – the index of the arc $(i, j) \in T_r$;
   - the flow associated to the arc $(i, j)$ with the index $e$, where $j \notin V_t$, for each spanning tree $T_r$:

$$x_{rekl} = \begin{cases} 0, & (i,j) \notin T_r, \\ \sum\limits_{(j,j') \in T_r} x_{re'kl}, & e'-\text{the index of the arc } (j,j') \end{cases}$$

   where $k = 1, 2, ..., p, l = 1, 2, ...q$.

As a result, intermediate solutions of the following form are obtained:

$$x_{rekl} = (x_{r111,}, x_{r112}, ..., x_{rupq}), r = 1, 2, ..., n$$

and the final admissible solution associated to the chromosome has the form:

$$x = \left( \sum_{i=1}^{n} x_{i111}, \sum_{i=1}^{n} x_{i112}, ..., \sum_{i=1}^{n} x_{iupq} \right).$$

Chromosome evaluation involves determining the value of the objective function for each of the admissible solutions associated with the chromosomes.

***Step 3.*** *Chromosome selection.* Chromosomes are sorted in order of increasing value of the objective function of the admissible solution associated with the chromosome. Chromosomes from the first half of the population are transferred to the new population.

***Step 4.*** *Crossover.* Crossover is applied to chromosomes transferred to the population $P(i)$ from the population $P(i-1)$ by selection. The same cut is randomly applied to compartment 1 of both parent-chromosomes. The offspring-chromosomes are obtained as follows:

- the first offspring-chromosome receives the spanning trees up to the cut of the mother-chromosome with the respective order of sources and types of products and the spanning trees after the cut of the father-chromosome with the respective order of destinations and types of transport;

- the second offspring-chromosome receives the spanning trees up to the cut of the father-chromosome with the respective order of sources and types of products and the spanning trees after the cut of mother-chromosome with the respective order of destinations and types of transport.

Each pair of parent-chromosomes generates two offspring-chromosomes and the size of the new population remains constant.

***Step 5.*** *Mutation.* A mutation with probability $\epsilon \in [0.1, 0.5]$ is applied to a gene on the offspring-chromosome and involves:

- for each compartment, *two* – sources, *three* – destinations, *four* – types of products, and *five* – types of transport, it is randomly decided whether the mutation is applied;
- in each of the selected compartments the values are changed between two randomly selected elements.

***Step 6.*** *Check the stop condition.* It involves stopping the algorithm after $k$ iterations. The final solution to the problem is the one that corresponds to the chromosome with the minimum objective function value in the last generated population. Go to *Step 2* if the stop condition is not met.

*In the case of large-scale problems, the execution time can serve as a condition for stopping the algorithm. Therefore, after a certain period of time, the generation of new populations is stopped, and the solution that corresponds to the chromosome with the best characteristics from the last population serves as a final solution to the problem.*

## 2.2 Theoretical results

The MGA algorithm is applied to transportation networks that meet the following conditions:

1. The graph which describes the transportation network is connected and acyclic;
2. There are at least 2 sources, 2 destinations, 2 types of products, 2 types of transport, and at least one intermediate node;
3. There is at least one path from each source to each destination;
4. Source nodes do not contain ascending arcs, and destination nodes do not contain descending arcs.

The following theorems are formulated and proved:

**Theorem 1.** *The MGA algorithm requires* $O(|V|(n|V|+n+m+p+q))$ *memory.*

*Proof.* The input data described by the restriction table is of size $O(upq)$. $O(upq)$ memory is required for the solution associated to the chromosome with the best characteristics of the population. A chromosome requires $O(n|V| + n + m + p + q)$ memory, and for the entire population – $O(4|V|(n|V| + n + m + p + q))$ memory is required. Therefore, the MGA algorithm requires $O(|V|(n|V| + n + m + p + q))$ memory. □

**Theorem 2.** *The complexity of an iteration of the MGA algorithm is* $O(npq|V|(m + |V| + u))$.

*Proof.* $O(upq)$ operations are required to initialize the input data described by the constraint table. Generating a chromosome requires $O(n|V| + n + m + p + q)$ operations, so generating the entire population requires $O(|V|(n|V| + n + m + p + q))$ operations. Evaluating the solution associated with a chromosome requires $O(n(mpq + |V|pq + upq))$ operations. Because there are $4|V|$ chromosomes in the population, the complexity of evaluating all solutions associated to the chromosomes is $O(npq|V|(m|V| + u))$. The crossover is of an $O(n|V| + n + m + p + q)$ complexity for one chromosome and $O(|V|(n|V| + n + m + p + q))$ for the entire population. Therefore, the time complexity of an iteration of the MGA algorithm is $O(npq|V|(m + |V| + u))$. □

*From Theorem 2, it results that the complexity of the MGA algorithm is* $O(U(n|V|+n+m+p+q))$*, where* $U$ *is the number of iterations necessary to obtain a final solution associated with the chromosome with good characteristics.*

**Theorem 3.** *The MGA algorithm converges to the local optimum.*

*Proof.* From the population $P(i-1)$ chromosomes are transferred to the population $P(i)$ for which the value of the objective function in the associated solution is minimal. The chromosomes of the populations generated at each iteration are associated with an admissible solution,

so it can be said that after the execution of a finite number of iterations the chromosome is detected to which is associated the solution that describes the local optimum. Therefore, the MGA algorithm converges to the local optimum. □

## 2.3 Practical results

The MGA algorithm was implemented in the Wolfram language and tested based on a series of examples, transportation problems of different sizes, that is, different number of sources, destinations, intermediate nodes, types of products, and types of transport that can be used.

Tables 1 and 2 present the results of solving a series of problems of different sizes, where cost functions are defined by concave non-decreasing functions. The size of the transportation problem is described by sources, destinations, types of products transported, types of transport used, and intermediate nodes. The total of the products available in the sources is equal to the total of the products needed in the destinations. The total of products of different types coincides with the total capacity of transport types used.

The tests were performed on an Intel i5-2500 machine with 4 Cores and 8 GB of DDR3 memory in Wolfram Mathematica 12. The time being given after the generation of 10 populations.

Table 1 presents the execution time of the MGA algorithm for problems of different sizes: number of sources, destinations, intermediate nodes, product types, and transport types, the solution of which uses from 48 unknowns in the case of the problem with 2 sources, 2 destinations, 2 types of products, 2 types of transport, and 2 intermediate nodes, up to 34500 unknowns in the case of the problem with 10 sources, 10 destinations, 10 types of products, 10 types of transport, and 10 intermediate nodes. As can be seen, the algorithm allows obtaining pseudo-optimal solutions in a reasonable amount of time even in the case of large-scale transportation problems.

Table 2 presents the results of solving a series of network transportation problems of different sizes, where for each of the 10 populations generated, there can be seen the changes in the value of the objective

Table 1. Execution time of the MGA algorithm

| *n/m/p/q* | *2/2/2/2* | *3/3/3/3* | *4/4/4/4* | *4/5/4/5* | *5/5/5/5* |
|---|---|---|---|---|---|
| 1 | 0.2500 | 1.3125 | 4.3750 | 6.6406 | 12.2656 |
| 2 | 0.2656 | 1.2500 | 4.3437 | 6.5781 | 12.4375 |
| 3 | 0.2812 | 1.2500 | 4.5468 | 6.6562 | 12.5781 |
| 4 | 0.3437 | 1.2968 | 4.4531 | 6.5937 | 12.4063 |
| 5 | 0.3281 | 1.2500 | 4.4531 | 6.5468 | 12.4844 |
| Unkn. | 48 | 261 | 864 | 1240 | 2125 |
| *n/m/p/q* | *6/6/6/6* | *7/7/7/7* | *8/8/8/8* | *9/9/9/9* | *10/10/10/10* |
| 1 | 29.6719 | 62.1563 | 118.938 | 213.656 | 353.234 |
| 2 | 29.5156 | 62.0000 | 117.906 | 211.906 | 359.266 |
| 3 | 29.4531 | 62.7344 | 118.922 | 216.281 | 361.453 |
| 4 | 29.6406 | 62.0625 | 117.375 | 211.656 | 356.750 |
| 5 | 20.9531 | 62.6406 | 117.031 | 216.406 | 355.703 |
| Unkn. | 4428 | 8232 | 14080 | 22599 | 34500 |

function calculated for the solution associated with the chromosome with the best characteristics in that population and the value of the total objective function for that population.

From the data presented in Table 2 it can be observed that $F_T(x)$ generally decreases from one iteration to another, with some exceptions – when crossover and mutation operations result in solutions that correspond to higher costs than those of the previous population. The insignificant increase of $F_T(x)$ is followed by decreases, which suggests that the majority of chromosomes in new populations have better characteristics than those in the population obtained in the previous iteration.

Although at several consecutive iterations the same value of $F(x)$ is repeated, which means that at several consecutive iterations a chromosome with better characteristics is not found, the algorithm is able to exit such a lock in a local optimum by determining a new chromosome with better characteristics and therefore an admissible solution for which the value of the objective function is lower.

Table 2. $F(x)$ and $F_T(x)$ of the MGA algorithm

| *n/m/p/q* Iteration | *4/4/4/4* | *5/5/5/5* | *6/6/6/6* | *7/7/7/7* | *10/10/10/10* |
|---|---|---|---|---|---|
| I | 30/2431 | 43/4078 | 60/6301 | 73/8897 | 94/17495 |
| II | 30/2263 | 43/3767 | 51/5837 | 73/8560 | 94/17176 |
| III | 29/2113 | 43/3584 | 51/5691 | 69/8059 | 94/16846 |
| IV | 29/2117 | 42/3343 | 51/5317 | 61/7774 | 94/16546 |
| V | 28/1885 | 41/3208 | 50/5145 | 61/7657 | 88/16157 |
| VI | 28/1890 | 40/3033 | 45/4878 | 55/7529 | 88/15799 |
| VII | 26/1851 | 37/2923 | 45/4744 | 55/7219 | 88/15839 |
| VIII | 17/1831 | 32/2801 | 45/4619 | 55/7286 | 85/15480 |
| IX | 17/1826 | 32/2775 | 44/4559 | 55/7126 | 85/15403 |
| X | 17/1819 | 32/2774 | 40/4551 | 55/6865 | 85/15299 |

The algorithm allows the determination of an optimal local solution for which the value of the objective function is lower the more iterations of the algorithm are executed. The number of iterations depends on the size of the problem being solved, the time available to provide the solution, and also the technological capabilities available.

## 3 Conclusions

For the formulated problem, the MGA genetic algorithm is proposed, which can obtain the solution in a reasonable amount of time for large-scale problems. In connection with other studies of non-linear transportation problems with several indices, the results have been obtained that complement the currently known results in this field. In particular:

1. the formulation of the non-linear transportation problem on a network with 5 indices described by sources, destinations, intermediate nodes, product types, and transport types as a non-linear transportation problem on a network with 3 indices described by

products, transport types, and the set of arcs;

2. building the MGA genetic algorithm for solving the non-linear transportation problem on a network with 5 indices formulated as a non-linear transportation problem on a network with 3 indices, described by concave cost functions, which is based on the application of the selection, crossover, and mutation operators of the genetic algorithm;

3. the description of a correct coding of the respective problem for the MGA genetic algorithm, so that after decoding only one admissible solution is obtained for each chromosome;

4. description of crossover and mutation operators, so that whenever they are applied within the MGA genetic algorithm, after decoding the newly created chromosome, an admissible solution of the problem is obtained.

## References

[1] O. Diaz-Para, A. Ruiz-Vanoye, B. B. Loranca, A. Fluentes-Penna, and R. A. Barrera-Camara, "A survey of transportation problem," *Journal of Applied Mathematics*, vol. 2014, Article ID: 848129, 17 p., Hindawi Publishing Corporation, DOI: http://dx.doi.org/10.1155/2014/848129. [Online] Available: http://downloads.hindawi.com/journals/jam/2014/848129.pdf.

[2] T. Pașa, "Multi-index transport problem with non-linear cost functions," *Romai J.*, vol. 14, no. 2, pp. 129–137, 2018. [Online] Available: https://rj.romai.ro/arhiva/2018/2/Pasa.pdf.

[3] T. Pașa and V. Ungureanu, "Solving the non-linear 4-index transportation problem," in *The Fifth Conference of Mathematical Society of the Republic of Moldova*, (Chișinău: Vladimir Andrunachievici Institute of Mathematics and Computer Science), 2019, pp. 221–224. [Online] Available: https://ibn.idsi.md/sites/default/files/imag_file/221-224_9.pdf.

[4] T. Pașa, "Solving non-linear multi-index transportation problems," *Romai J.*, vol. 15, no. 2, pp. 91–99, 2019. [Online] Available: https://rj.romai.ro/arhiva/2019/2/Pasa.pdf.

[5] T. Pașa, "Algorithms for solving nonlinear multi-index transportation problems," *Studia Universitatis Moldaviae, Seria Științe Exacte*, vol. 132, no. 2, pp. 36–44, 2020. ISSN: 1857-2073. Online ISSN: 2345-1033. (in Romanian)

[6] P.-X. Ninh, "N index transportation problem," *Mathematical Journal*, vol. 7, no. 1, pp. 18–25, 1979.

[7] K. B. Haley, "The solid transportation problem," *Operat. Research*, no. 10, pp. 448–463, 1962.

[8] W. Junginger, "On reprezentative of multi-index transportation problems," *European Journal of Operational Research*, no. 66, pp. 353–371, 1993. DOI:10.1016/0377-2217(93)90223-A.

Tatiana Pașa,

Moldova State Univerity
Faculty of Mathematics and Computer Science
60 A. Mateevici, MD-2009, Chișinău
Republic of Moldova
E–mail: pasa.tatiana@yahoo.com

Received August 16, 2021
Accepted January 30, 2022