A novel heuristic approach for solving the two-stage transportation problem with fixed-charges associated to the routes

Ovidiu Cosma
Department of Mathematics and Computer Science
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
ovidiu.cosma@cunbm.utcluj.ro

Petrica Pop
Department of Mathematics and Computer Science
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
petrica.pop@cunbm.utcluj.ro

Cosmin Sabo
Department of Mathematics and Computer Science
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
cosminsabo@gmail.com

***Abstract*—In this paper, we are addressing the two-stage transportation problem with fixed charges associated to the routes and propose an efficient heuristic algorithm for the total distribution costs minimization. Our heuristic approach builds several initial solutions by processing customers in a specific order and choosing the best available supply route for each customer. After each initial solution is built, a process of searching for better variants around that solution follows, restricting the way the transport routes are chosen. Computational experiments were performed on a set of 20 benchmark instances available in the literature. The achieved computational results show that our proposed solution approach is highly competitive in comparison with the existing approaches from the literature.**

Keywords—two-stage transportation problem, heuristic algorithms

I. INTRODUCTION

This paper focuses on a variant of the transportation problem, namely the two-stage transportation problem with fixed charges associated to the routes. The problem models a distribution network in a two-stage supply chain which involves: manufacturers, distribution centers and customers and its main characteristic is that a fixed charge is associated with each route that may be opened, in addition to the variable transportation cost which is proportional to the amount of goods shipped. The objective of the considered transportation problem is to identify and select the routes from manufacturers through the distribution centers to the customers satisfying the capacity constraints of the manufacturers in order to meet specific demands of the customers under minimal total distribution costs. In this form, the problem was introduced by Gen et al. [3]. This work deals with a variant of the transportation problem, namely the fixed-cost transportation problem in a two-stage supply chain network. In this extension, our aim is to identify and select the manufacturers and the distribution centers fulfilling the demands of the customers under minimal costs.

The two-stage transportation problem was first considered by Geoffrion and Graves [4]. Since then different variants of the problem have been proposed in the literature determined by the characteristics of the transportation system which models the real world application and several methods, based on relaxation techniques and on exact, heuristic and metaheuristics algorithms, have been developed for solving them.

Marin and Pelegrin [7] developed an algorithm based on Lagrangian decomposition and branch-and-bound techniques in the case when the manufacturers and the distribution centers have no capacity constraints and there are fixed costs associated to opening the distribution centers and the number of opened distribution centers is fixed and established in advance. Marin [8] proposed a mixed integer programming formulation and provided lower bounds of the optimal objective values based on different Lagrangian relaxations for an uncapacitated version of the problem when both manufacturers and distribution centers have associated fixed costs when they are used. Pirkul and Jayaraman [13] studied a multi-commodity, multi-plant, capacitated facility location version of the problem and proposed a mixed integer programming model and a solution approach based on Lagrangian relaxation of the problem. The same authors in [6] extended their model by taking into considerations the acquisition of raw material and described a heuristic algorithm that uses the solution generated by a Lagrangian relaxation of the problem. Amari [1] investigated a different version of the problem, allowing the use of several capacity levels of the manufacturers and distribution centers and developed an efficient solution approach based on Lagrangian relaxation for solving it.

Raj and Rajendran [18] proposed two scenarios for the two-stage transportation problem: the first scenario, called Scenario-1, takes into consideration fixed costs associated to the routes in addition to unit transportation costs and unlimited capacities of the distribution centers, while the second one, called Scenario 2, takes into consideration the opening costs of the distribution centers in addition to unit transportation costs.

They developed a genetic algorithm with a specific coding scheme suitable for two-stage transportation problems and as well they introduced a set of 20 benchmark instances. The same authors proposed in [17] a solution representation that allows a single-stage genetic algorithm to solve the considered problem. The major feature of these methods is a compact representation of a chromosome based on a permutation. A different genetic algorithm proposed for solving the two-stage transportation problem with fixed charge associated to the routes from manufacturers to customers was described by Jawahar and Balaji [5]. Recently, Pop et al. [16] presented, in the case of Scenario-1, a hybrid algorithm that combines a steady-state genetic algorithm with a powerful local search procedure. In the case of the two-stage transportation problem with fixed charges for opening the distribution centers, as introduced by Gen et al. [3] and called it Scenario-2 by Raj and Rajendran [18], the mentioned authors developed genetic algorithms based on sequentially getting first a transportation tree for the transportation problem from distribution centers to customers and secondly a transportation tree for the transportation problem from manufacturers to distribution centers. In both genetic algorithms, the chromosome contains two parts, each encoding one of the transportation trees. A different genetic algorithm was described by Calvete et al. [2], whose main characteristic is the use of a new chromosome encoding that provides information about the distribution centers that can be used within the distribution system.

A different variant was investigated by Molla et al. [9] in which it is considered only one manufacturer. They presented an integer programming mathematical model of the problem and proposed a spanning tree-based genetic algorithm with a Prüfer number representation and an artificial immune algorithm for solving the problem. Some remarks concerning the mathematical model of the problem were pointed out by El-Sherbiny [19]. For this variant of the two-stage transportation problem, Pintea et al. [10,12] developed some hybrid classical heuristic approaches and described an improved hybrid algorithm that combines the Nearest Neighbor search heuristic with a local search procedure for solving the problem. Recently, Pop et al. [15] proposed a novel hybrid heuristic approach which was obtained by combining a genetic algorithm based on a hash table coding of the individuals with a powerful local search procedure.

Another version of the two-stage transportation problem with one manufacturer takes into consideration the environmental impact by reducing the greenhouse gas emissions. This variant was introduced by Santibanez-Gonzalez et al. [19], dealing with a practical application occurring in the public sector. Considering this variant of the two-stage transportation problem, Pintea et al. [11] proposed a set of classical hybrid heuristic approaches and Pop et al. [14] developed an efficient reverse distribution system for solving the problem.

The variant addressed in this paper considers a two-stage transportation problem with fixed charge associated with each route that may be opened, as introduced by Gen et al. [3]. This transportation problem has been also studied by Raj and Rajendran [18], who called it Scenario-1, Jawahar and Balaji [5] and Pop et al. [16]. In all mentioned papers, the authors proposed genetic algorithms for solving the problem.

Our paper is organized as follows: in Section 2, we formally define the two-stage transportation problem with fixed-charge associated to the routes. The developed heuristic algorithm is presented in Section 3 and the computational experiments and the achieved results are presented, analyzed and discussed in Section 4. Finally, in the last section, we present the obtained results in this paper and propose some future research directions.

II. DEFINITION OF THE TWO-STAGE TRANSPORTATION PROBLEM WITH FIXED-CHARGES ASSOCIATED TO THE ROUTES

In order to provide a formal definition the considered two-stage transportation problem with fixed-charges associated to the routes, we begin by defining the sets, decision variables and parameters used in our paper:

p is the total number of manufacturers

q is the total number of distribution centers

r is the total number of customers

i is a manufacturer identifier, $i \in \{1,\dots,p\}$

j is a distribution center identifier, $j \in \{1,\dots, q\}$

k is a customer identifier, $k \in \{1,\dots, r\}$

$D[k]$ is the demand of the customer k

$I[k]$ is the number of units delivered to customer k

$S[i]$ is the capacity of manufacturer i

$O[i]$ is the number of units delivered by manufacturer i

$F1[i,j]$ is the fixed transportation charge for the link from manufacturer i to distribution center j

$F2[j,k]$ is the fixed transportation charge for the link from distribution center j to customer k

$C1[i,j]$ is the unit cost of transportation from manufacturer i to distribution center j

$C2[j,k]$ is the unit cost of transportation from distribution center j to customer k

$X1[i,j]$ is the number of units transported from manufacturer i to distribution center j

$X2[j,k]$ is the number of units transported from distribution center j to customer k

The structure of the distribution system is presented in the next figure.

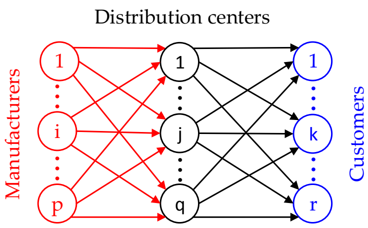

Fig. 1. The structure of the distribution system

Next we present the mathematical model of the two-stage transportation problem with fixed charges associated to the routes introduced by Jawahar and Balaji [5], based on integer programming.

Minimise

$$Z = \sum_{i=1}^{p}\sum_{j=1}^{q}(C1[i,j]X1[i,j] + F1[i,j]\delta 1[i,j]) + \\ + \sum_{j=1}^{q}\sum_{k=1}^{r}(C2[j,k]X2[j,k] + F2[j,k]\delta 2[j,k]) \quad (1)$$

Subject to

$$\sum_{j=1}^{q} X1[i,j] \leq S[i] \quad \forall i \in \{1,\dots,p\} \quad (2)$$

$$\sum_{j=1}^{q} X2[j,k] = D[k] \quad \forall k \in \{1,\dots,r\} \quad (3)$$

$$X1[i,j] \geq 0, \quad X2[j,k] \geq 0 \quad (4)$$

where

$$\delta 1[i,j] = \begin{cases} 0, if\ X1[i,j] = 0 \\ 1, if\ X1[i,j] > 0 \end{cases}, \quad \delta 2[j,k] = \begin{cases} 0, if\ X2[j,k] = 0 \\ 1, if\ X2[j,k] > 0 \end{cases}$$

The objective function (1) minimizes the total transportation cost: the fixed costs and transportation per-unit costs. Constraints (2) guarantee that the quantity shipped out from each plant does not exceed the available capacity and constraints (3) guarantee that the total shipment received from DCs by each customer is equal to its demand. Constraints 4 ensure the integrality and non-negativity of the decision variables.

III. THE HEURISTIC ALGORITHM FOR SOLVING THE TWO-STAGE TRANSPORTATION PROBLEM WITH FIXED CHARGES ASSOCIATED TO THE ROUTES

The operation of the proposed algorithm is shown in Fig. 2. It executes a fixed number of iterations that build several solution variants, of which the best is retained. The algorithm consists of the following two nested blocks:

A. Build variants,

B. Build distribution solution.

The *Build variants* block (A) calls the *Build distribution solution* block (B) to build a distribution solution, and then looks for better variants around it by applying a set of restrictions to the supply routes. The defined restrictions determine how the supply routes will be chosen within the new variants built by calling block B. The *Build distribution solution* block (B) builds a distribution solution, satisfying the demands of all customers, one by one. The resulting solution is saved only if it is better than all previous solutions.

The algorithm uses the following data structures:

- *Instance properties* (6) containing the fixed costs of opening the routes (*F1*, *F2*), the unit transport costs (*C1*, *C2*), the production capacities of the manufactures (*S*) and the demands of the customers (*D*).
- *Solution data (4)* containing the quantities transported on the routes from manufacturers to distribution centers (*X1*) and from distribution centers to customers (*X2*), the input quantities to customers (*I*) and the output quantities from manufacturers (*O*). This data structure will be updated during the construction of a solution. So at the end $I[k] = D[k], k \in \{1,\dots,r\}$ and $\sum_{i=1}^{n} O_i = \sum_{k=1}^{r} D_k$. Also this structure contains a series of restrictions for the routes from manufacturers to the distribution centers, which are applied in the route selection process (*R*).
- *Route* (5.1) that specifies a transport route for *a* units from manufacturer *i* through, distribution center *j*, to customer *k*.
- *Customers order* (2.2), which is an array containing the order in which customers will be processed by the algorithm.
- *Used permutations* (2.1), which is a hash set that contains all the permutations that were used in previous iterations of the algorithm.
- *Best solution (9)*, containing the quantities transported on the distribution routes within the optimum solution.

The algorithm works in the following way:
The *Shuffle customers* module (1) arranges the customers in random order, through the *Duplicate detector* module (2), which saves all permutations that were previously used in a hash set (*Used permutations* 2.1). Thus, any permutation that has been previously used is effectively detected and rejected. The operation ends when a permutation that was not previously used is generated.

Next, more iterations of the *Build variants* block (A) are processed. The total number of iterations is set at the initialization of the algorithm, based on the number of customers. This block builds a distribution solution by processing customers in the order given by the *Customers order* array (2.2), and then searches for more advantageous variants around this solution. For the construction of each solution, the *Reset solution* block (3) deletes the data corresponding to the previous solution, initializing the *X1*, *X2*, *I* and *O* arrays from the *Solution data* structure (4) with zeros. Thus, this structure is initialized for building a new solution.

The *Build distribution solution* block (B) builds a distribution solution, processing all customers, one by one, in the order given by the *Customers order array* (2.2). The *Route selection* module (5) seeks for each customer the most advantageous route of supply in the conditions created by meeting the demands of previous customers, resulting in the opening of some transport routes and consuming a part of the production capacity of manufacturers. The result returned by this module is a supply route (5.1). Each client's request can be satisfied in one or more steps, as the amount *a* of the route supports or not the customer's entire demand. The *Reserve route module* (7) reserves the route (5.1) by updating the *X1*, *X2*, *I* and *O* arrays from the *Solution data* structure (4). The processing of client k ends when $D[k] = I[k]$.

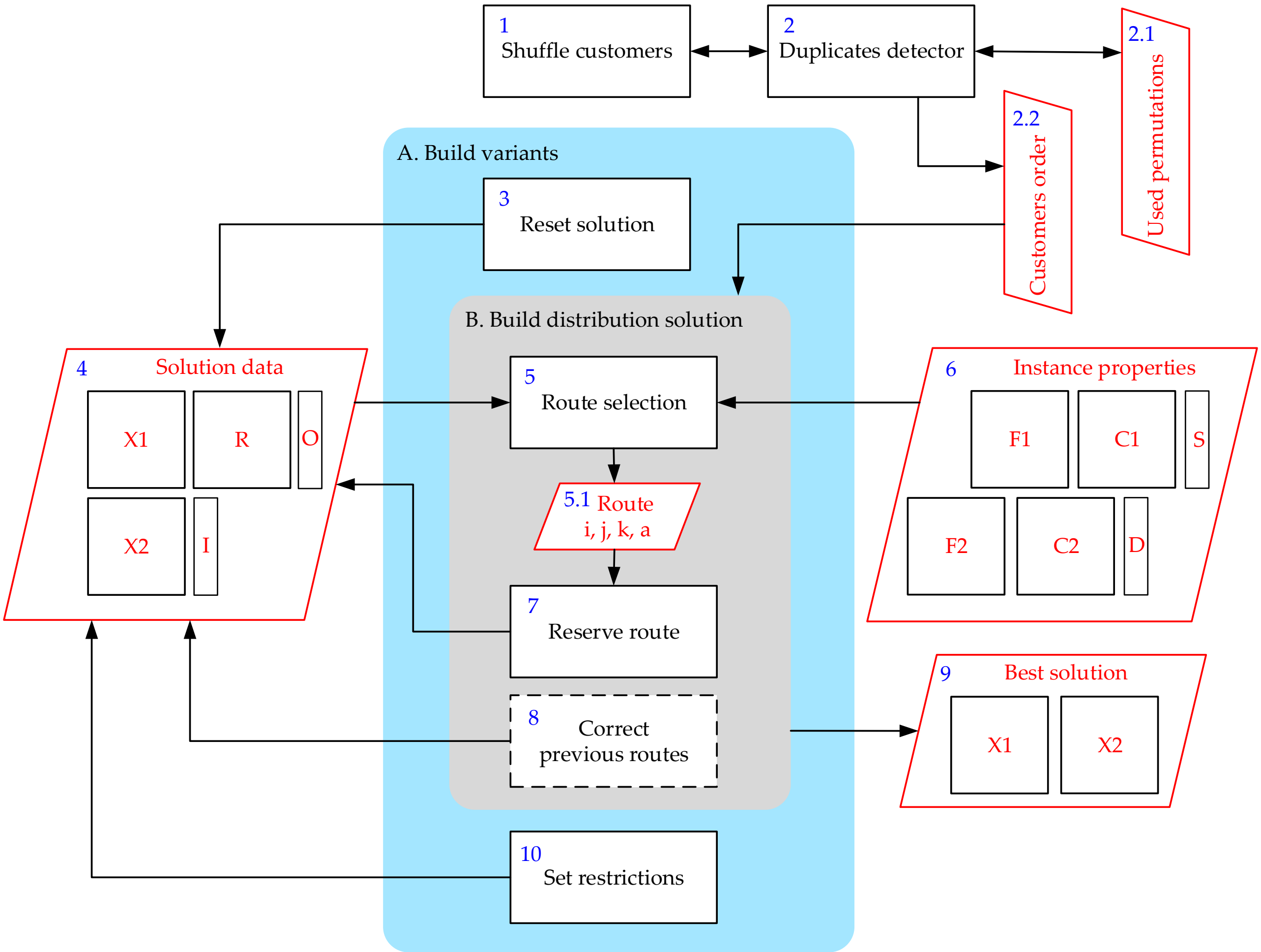

Fig. 2. The operation of the proposed heuristic algorithm

If there were no limitations on the capacities of the manufacturers and if there were no fixed costs for opening the transport routes, then modules 5 and 7 would build the optimal solution from the first attempt. But because of these restrictions, it is not certain that the optimal supply of a customer that can be found at any given time in the process of building a solution leads to the optimization of the entire distribution system. Any decision to supply a customer involves consuming a quantity of the production capacity of the manufacturers and possibly opening new transport routes. This will influence all the decisions to be taken within the algorithm. Consequently, the order in which customers are processed determines the final result. At the end of each customer's processing, an optional step of reviewing all previous decisions is applied, using the *Correct previous routes* module (8). This module deletes all previously reserved routes one after the other, and then attempts to replace them with variants that are more advantageous in the new conditions created by processing the last customer demand. The old routes are modified only if the change leads to a better solution. Using this module reduces on average the number of iterations needed to find the optimal solution, but the runtime of the algorithm increases as the complexity of the iterations increases. For comparison, the results obtained with and without this module will be presented in the *Computation Results* section (IV).

For certain distribution systems, it is not possible to reach the optimal solution simply by changing the customer processing order (using the *Shuffle customers* module 1) and the corrections performed by the *Correct previous routes* module (8). This is because the *Route selection* module (5) processes a single client at a time. This will always choose the optimal decision for each client, not the decisions optimizing the entire distribution system. Consequently, for these distribution systems, it is necessary to introduce new restrictions, which change the way decisions are made in the

Route selection module (5). These restrictions are fixed in the *Set restrictions* module (10). This module marks at each iteration one of the manufacturer-distribution center routes as mandatory. This route will be used with priority in the construction of the distribution solution until the manufacturer's production capacity is depleted. Thus, a search is made around the initial solution, by building other $p \cdot q$ variants of distribution systems.

In order to illustrate the operation of the algorithm, let's consider the example in figure 3. It represents a distribution system with one manufacturer (*M1*), two distribution centers (*DC1* and *DC2*) and two customers (*U1* and *U2*). The manufacturer's production capacity is 8 units, and *U1* and *U2* customer's demands are for 3 and 4 units, respectively. The transport costs are as follows: *C1 = {5, 3}, F1 = {10, 20}, C2[j, k] = 6, F2[j, k] = 7, j = {1, 2}, k = {1, 2}*. For this distribution system, there are two possible permutations for the customer set (*{U1, U2}*, and *{U2, U1}*), and the *Build Distribution Solution* module will attempt to build two variants of distribution systems.

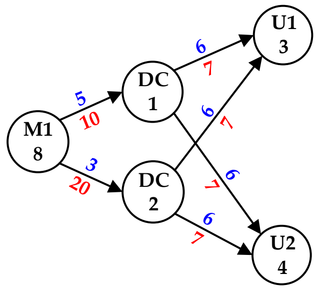

Fig. 3. A small example of a distribution system

In the first variant, the customer *U1* demand will be processed first. Of the two possible routes *M1-DC1-U1* and *M1-DC2-U1*, the first will be chosen, because it is more advantageous, resulting in a cost of (5 + 6) * 3 + 10 + 7 = 50. The cost of the second route is (3 + 6) * 3 + 20 + 7 = 54. Next the most advantageous route for the *U2* customer is chosen, which is *M1-DC1-U2*, resulting in a cost of (5 + 6) * 4 + 7 = 51. Thus, the total cost of the distribution solution is 50 + 51 = 101.

In the second variant, the most advantageous route for the U2 customer is chosen first. This is *M1-DC1-U2*, resulting in a cost of (5 + 6) * 4 + 10 + 7 = 61. Next the most advantageous route for the *U1* customer is chosen, which is *M1-DC1-U1*, resulting in a cost of (5 + 6) * 3 + 7 = 40. In conclusion, for both possible permutations, the same distribution solution is obtained in this example, which is suboptimal. This result is due to the fact that the choice of routes is always looking for the optimal solution for a particular customer in certain conditions, not the solution optimizing the whole distribution system.

The optimal solution can be reached through the action of the *Set restrictions module*, as follows: If the *M1-DC2* route is set as mandatory, it will have to be used with priority until the *M1* manufacturer's capacity is exhausted. Thus, if the customer *U1* is processed first, then the route *M1-DC2-U1* will be chosen, resulting in a cost of (3 + 6) * 3 + 20 + 7 = 54. Next the route *M1-DC2-U2* will be chosen for the *U2* customer, resulting in a cost of (3 + 6) * 4 + 7 = 43. Thus, the total cost of the distribution solution is 54 + 43 = 97.

IV. COMPUTATIONAL RESULTS

In order to analyze the performance of our proposed heuristic approach, we tested it on a set of 20 test instances that was generated by Gen et al. [3]. The files of the instances are available at the following address: https://sites.google.com/view/tstp-instances/.

Our algorithm was coded in Java 8 and we performed 30 independent runs for each instance on a PC with Intel Core i5-4590 processor at 3.3GHz, 4GB RAM and Windows 10 Education 64 bit operating system.

In Table 1, we show the computational results of our proposed heuristic algorithm in comparison with the genetic algorithm described by Jawahar and Balaji [4], called *JRGA*, the two genetic algorithms introduced by Raj and Rajendran [18], denoted by *TSGA* and *SSGA* and the hybrid genetic algorithm (*HGA*) described by Pop et al. [16]. The first column in the Table 1 gives the size of the instances, the next columns provide the solution achieved by the genetic algorithm described by Jawahar and Balaji [4], the two genetic algorithms introduced by Raj and Rajendran [18], the hybrid genetic algorithm described by Pop et al. [16] and the number of solution evaluations necessary to obtain it. The results written in bold represent cases for which the obtained solution is the best existing in literature.

The following columns show the results of our heuristic algorithm, obtained with and without the correction module (8). For each test instance, the running time and the number of solutions evaluated until the best solution is found are presented. Both the best values and the averages calculated for all 30 runs are presented.

Our algorithm finds the best known solution for each of the 20 test instances at each of the 30 runs, which demonstrates its robustness. In the variant without the correction block, all 20 instances are resolved in less than 1ms. In the version with the correction block, the average number of evaluated solutions is lower than in the case of the other known algorithms for all of the test instances. This is true also for the variant without the correction block, with only one exception.

REFERENCES

[1] A. Amiri, "Designing a distribution network in a supply chain system: Formulation and efficient solution procedure," Eur. J. of Oper. Res., vol. 171(2), pp. 567-576, 2006.

[2] H. Calvete, C. Gale, and J. Iranzo, "An improved evolutionary algorithm for the two-stage transportation problem with fixed charge at depots," OR Spectrum, vol. 38, pp. 189-206, 2016.

[3] M. Gen, F. Altiparmak, and L. Lin, "A genetic algorithm for two-stage transportation problem using priority based encoding," OR Spectrum, vol. 28, pp. 337-354, 2006.

[4] A.M. Geoffrion and G.W. Graves, "Multicommodity distribution system design by Benders decomposition," Management Sci., vol. 20, pp. 822-844, 1974.

[5] N. Jawahar, and A.N. Balaji, "A genetic algorithm for the two-stage supply chain distribution problem associated with a fixed charge," Eur. J. of Oper. Res., vol. 194, pp. 496-537, 2009.

[6] V. Jayaraman, and H. Pirkul, "Planning and coordination of production and distribution facilities for multiple commodities," Eur. J. of Oper. Res, vol. 133(2), pp. 394-408, 2001.

[7] A. Marin, and B. Pelegrin, "A branch-and-bound algorithm for the transportation problem with locations p transshipment points," Comp. & Oper. Res, vol. 24(7), pp. 659-678, 1997.

[8] A. Marin, "Lower bounds for the two-stage uncapacitated facility location problem," Eur. J. of Oper. Res., vol. 179(3), pp. 1126-1142, 2007.

[9] S. Molla-Alizadeh-Zavardehi, M. Hajiaghaei-Kesteli, and R. Tavakkoli-Moghaddam, "Solving a capacitated fixed-cost transportation problem

by artificial immune and genetic algorithms with a Prüfer number representation," Exp. Syst. with Appl., vol. 38, pp. 10462-10474, 2011.

[10] C.-M. Pintea, C. Pop Sitar, M. Hajdu-Macelaru, and P.C. Pop, "A Hybrid Classical Approach to a Fixed-Charge Transportation Problem," In Proc. of HAIS 2012, Part I, Eds E. Corchado et al., Lecture Notes in Computer Science, vol. 7208, pp. 557-566, 2012.

[11] C.-M. Pintea, P.C. Pop, and M. Hajdu-Macelaru, "Classical Hybrid Approaches on a Transportation Problem with Gas Emissions Constraints" In Proc. of SOCO 2012, Advances in Intelligent Systems and Computing, vol. 188, pp. 449-458, 2013.

[12] C.M. Pintea, and P.C. Pop, "An improved hybrid algorithm for capacitated fixed-charge transportation problem," Logic J. of IJPL vol. 23(3), pp. 369-378, 2015.

[13] H. Pirkul, and V. Jayaraman, "A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution," Comp. & Oper. Res., vol. 25(10), pp. 869-878, 1998.

[14] P.C. Pop, C.-M. Pintea, C. Pop Sitar, and M. Hajdu-Macelaru, "An efficient Reverse Distribution System for solving sustainable supply chain network design problem," J. of Appl. Logic, vol. 13(2), pp. 105-113, 2015.

[15] P.C. Pop, O. Matei, C. Pop Sitar, and I. Zelina, "A hybrid based genetic algorithm for solving a capacitated fixed-charge transportation problem,"Carpathian J. Math., vol. 32(2), pp. 225-232, 2016.

[16] P.C. Pop, C. Sabo, B. Biesinger, B. Hu, and G. Raidl, "Solving the two-stage fixed-charge transportation problem with a hybrid genetic algorithm," Carpathian J. Math., vol. 33(3), pp. 365-371, 2017.

[17] K.A.A.D. Raj, and C. Rajendran, "A Hybrid Genetic Algorithm for Solving Single-Stage Fixed-Charge Transportation Problems," Technology Operation Management, vol. 2(1), pp. 1-15, 2011.

[18] K.A.A.D. Raj, and C. Rajendran, "A genetic algorithm for solving the fixed-charge transportation model: Two-stage problem," Comp. & Oper. Res., vol. 39(9), pp. 2016-2032, 2012.

[19] E. Santibanez-Gonzalez, R. Del, G. Robson Mateus, and H. Pacca Luna, "Solving a public sector sustainable supply chain problem: A Genetic Algorithm approach," In Proc. of Int. Conf. of Artificial Intelligence (ICAI), Las Vegas, USA, pp. 507-512, 2011.

[20] M.M. El-Sherbiny, "Comments on "Solving a capacitated fixed-cost transportation problem by artificial immune and genetic algorithms with a Prüfer number representation" by Molla-Alizadeh-Zavardehi, S. et al. Expert Systems with Applications (2011)," Exp. Syst. with Appl., vol. 39, pp. 11321-11322, 2012.

TABLE I. COMPUTATIONAL RESULTS OBTAINED BY OUR PROPOSED APPROACH IN COMPARISON WITH OTHER ALGORITHMS FROM LITERATURE

Instance size	JRGA [4]		TSGA [18]	SSGA [18]		HGA [16]		Our solution approach								
								obj	with Correction block				without Correction block			
									run time		#eval		run time		#eval	
	obj	#eval	obj	obj	#eval	obj	#eval		best	avg.	best	avg.	best	avg.	best	avg.
2x2x3	**112600**	1444	**112600**	**112600**	637	**112600**	2	**112600**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
2x2x4	**237750**	1924	**237750**	**237750**	857	**237750**	2	**237750**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
2x2x5	**180450**	2404	**180450**	**180450**	1214	**180450**	319	**180450**	<1	**<1**	1	**1.37**	<1	**<1**	1	8.87
2x2x6	**165650**	2884	**165650**	**165650**	1354	**165650**	324	**165650**	<1	**<1**	1	**3.27**	<1	**<1**	1	30.93
2x2x7	**162490**	3364	**162490**	**162490**	1889	**162490**	335	**162490**	<1	**<1**	1	**5.60**	<1	**<1**	1	52.77
2x3x3	**59500**	2164	**59500**	**59500**	1503	**59500**	317	**59500**	<1	**<1**	1	**3.50**	<1	**<1**	1	11.03
2x3x4	**32150**	2884	**32150**	**32150**	1859	**32150**	339	**32150**	<1	0.53	1	**2.10**	<1	**<1**	1	6.97
2x3x6	69970	4324	67380	**65945**	2577	**65945**	356	**65945**	<1	0.53	1	**19.43**	<1	**<1**	1	22.00
2x3x8	263000	5764	**258730**	**258730**	5235	**258730**	546	**258730**	<1	5.17	1	**435.47**	<1	**<1**	36	728.77
2x4x8	80400	7684	84600	**77400**	5246	78550	1039	**77400**	<1	**<1**	1	**24.23**	<1	**<1**	10	661.00
2x5x6	94565	7204	80865	**75065**	3574	80865	430	**75065**	<1	**<1**	1	**1.00**	<1	**<1**	1	6.50
3x2x4	**47140**	2884	**47140**	**47140**	1429	**47140**	321	**47140**	<1	0.53	1	**1.37**	<1	**<1**	1	1.50
3x2x5	178950	3604	178950	**175350**	2061	178950	320	**175350**	<1	1.03	1	**20.50**	<1	**<1**	6	45.80
3x3x4	**57100**	4324	61000	**57100**	3060	**57100**	354	**57100**	<1	**<1**	1	**1.73**	<1	**<1**	1	18.93
3x3x5	**152800**	5404	156900	**152800**	4555	**152800**	335	**152800**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
3x3x6	**132890**	6484	**132890**	**132890**	2981	**132890**	3	**132890**	<1	**<1**	1	**1.00**	<1	**<1**	1	**1.00**
3x3x7(a)	104115	7564	106745	**99095**	7095	103815	1330	**99095**	<1	2.07	11	175.70	<1	**<1**	1	**1.00**
3x3x7(b)	287360	7564	295060	**281100**	7011	**281100**	380	**281100**	<1	**<1**	1	9.93	<1	**<1**	1	**1.00**
3x4x6	77250	8644	81700	**76900**	7105	77250	373	**76900**	<1	**<1**	1	**8.73**	<1	**<1**	1	8.87
4x3x5	**118450**	7204	**118450**	**118450**	4227	**118450**	394	**118450**	<1	**<1**	1	**9.57**	<1	**<1**	1	30.93