CARPATHIAN J. MATH.
Volume **35** (2019), No. 2,
Pages 153 - 164

Online version at `https://semnul.com/carpathian/`
Print Edition: ISSN 1584 - 2851; Online Edition: ISSN 1843 - 4401
DOI: https://doi.org/10.37193/CJM.2019.02.04

Dedicated to Prof. Juan Nieto on the occasion of his 60^{th} *anniversary*

An efficient iterated local search heuristic algorithm for the two-stage fixed-charge transportation problem

OVIDIU COSMA[1], PETRICĂ C. POP[1] and CORINA POP SITAR[2]

ABSTRACT. This paper concerns the two-stage transportation problem with fixed charges associated to the routes and proposes an efficient multi-start Iterated Local Search (ILS) procedure for the total distribution costs minimization. Our heuristic approach constructs an initial solution, uses a local search procedure to increase the exploration, a perturbation mechanism and a neighborhood operator in order to diversify the search. Computational experiments were performed on two sets of instances: one that consists of 20 benchmark instances available in the literature and a second one that contains 10 new randomly generated larger instances. The achieved computational results prove that our proposed solution approach is highly competitive in comparison with the existing approaches from the literature.

1. INTRODUCTION

The transportation problem has a venerable history in mathematics and economics. The problem was considered for the first time in 1781 by Monge, who encountered it in a practical application, namely the soil-transportation problem. In the standard transportation problem, given a number of sources and a number of destinations, we look for an optimum transportation schedule keeping in mind the minimization of the transportation costs. Hitchcok stated this kind of problem in 1941 and Kantorovich considered the continous case of the problem.

The importance and famousness of the transportation problem originates from several facts: there exist very efficient algorithms for solving the standard version of the problem, it has several real-world applications and it has been successfully applied to many other optimization problems. However, for many researchers the extensions of the classical transportation problem are more attractive, being more difficult and having many practical applications.

This work deals with a variant of the transportation problem, namely the fixed-cost transportation problem (FCTP) 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 main characteristic of the fixed-charge transportation problem 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 two-stage transportation problem was introduced by Geoffrion and Graves [8]. Since then several variants of the problem have been studied and several methods, based on exact and heuristic algorithms, have been proposed for solving them.

Received: 25.11.2018. In revised form: 20.03.2019. Accepted: 27.03.2019
2010 *Mathematics Subject Classification.* 90C08.
Key words and phrases. *transportation system design, two-stage fixed-charge transportation problem, iterated local search.*
Corresponding author: Petrică C. Pop; petrica.pop@cunbm.utcluj.ro

Marin and Pelegrin [12] proposed a solution approach using 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 [14] presented a mixed integer programming formulation and provided lower bounds of the optimal objective values using 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 [18] investigated a multi-commodity, multi-plant, capacitated facility location version of the problem and described a mixed integer programming model and a solution approach using a Lagrangian relaxation of the problem. The same authors in [11] extended their model by taking into consideration the acquisition of raw material and described a heuristic algorithm that uses the solution generated by a Lagrangian relaxation of the problem. Amari [1] studied 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. Calvete et al. [2] investigated a bi-level optimization problem that deals with the planning of a distribution system that permits to take into account the manner in which decisions made at the distribution stage of the supply chain can affect and be affected by decisions made at the manufacturing stage. They proposed a mixed integer formulation of the considered transportation problem and as well a hybrid solution approach that combines the use of an evolutionary algorithm to control the supply of distribution centers with optimization techniques to determine the delivery from distribution centers to customers and the supply from manufacturers to distribution centers.

In one of these variants, there exists only one manufacturer and this version was considered by Molla et al.[14]. They described an integer programming mathematical formulation of the problem and they proposed also a spanning tree-based genetic algorithm with a Prüfer number representation and an artificial immune algorithm for solving it. Some comments concerning the mathematical formulation of the problem were published by El-Sherbiny [6]. Pintea et al. [15] described some hybrid classical approaches and Pintea and Pop [17] developed an improved hybrid algorithm combining the Nearest Neighbor search heuristic with a local search procedure for solving the two-stage transportation problem with fixed costs. Pop et al. [20] described a novel hybrid heuristic approach obtained by combining a genetic algorithm based on a hash table coding of the individuals with a powerful local search procedure. Recently, Cosma et al. [5] proposed an efficient hybrid heuristic approach that constructs an initial solution while using a local search procedure whose aim is to increase the exploration and for the purpose of diversifying the search, a neighborhood structure is used.

Another version of the two-stage transportation problem takes into consideration its impact on the environment by limiting the greenhouse gas emissions and was introduced by Santibanez-Gonzales [24] for dealing with a practical application from the public sector. For this variant of the problem, Pintea et al. [16] provided a set of classical hybrid heuristic approaches and Pop et al. [19] proposed an efficient reverse distribution system for solving it.

Raj and Rajendran [22] considered two scenarios of the problem: the first scenario (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 (Scenario 2) which takes into consideration the opening costs of the distribution centers in addition to unit transportation costs. Recently, Calvete et al. [3], in the case of Scenario 2, proposed a hybrid evolutionary algorithm whose main characteristic is the use of a chromosome encoding that provides information about the distribution centers used within the distribution system.

In the form considered in our paper, the problem was defined by Jawahar and Balaji [10]. They described a genetic algorithm (GA) with a specific coding scheme suitable for two-stage problems. The same authors introduced a set of 20 benchmark instances and their computational results have been compared to lower bounds and approximate solutions obtained from a relaxation. Raj and Rajendran [22] called this variant Scenario-1 and developed a two-stage genetic algorithm (TSGA) in order to solve the problem. They also proposed a solution representation that allows a single-stage genetic algorithm (SSGA) [23] to solve it. The major feature of these methods is a compact representation of a chromosome based on a permutation. Recently, Pop et al. [21] described a hybrid algorithm that combines a steady-state genetic algorithm with a local search procedure.

Our paper is organized as follows. In Section 2, we define the two-stage fixed-charge transportation problem. The developed multi-start Iterated Local Search 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 point out the results obtained in this paper and present future research directions.

2. Definition of the two-stage transportation problem with fixed costs associated to the routes

In order to define the considered two-stage fixed-cost transportation problem, we start by defining the related sets, decision variables and parameters:

p	the number of manufacturers
q	the number of distribution centers
r	the number of customers
i	manufacturer identifier, $i \in \{1, ..., p\}$
j	distribution center identifier, $j \in \{1, ..., q\}$
k	customer identifier, $k \in \{1, ..., r\}$
D_k	the demand of customer k
S_i	capacity of manufacturer i
f_{ij}	the fixed transportation charges for the link from manufacturer i to distribution center j
g_{jk}	the fixed transportation charges for the link from distribution center j to customer k
b_{ij}	unit cost of transportation from manufacturer i to distribution center j
c_{jk}	unit cost of transportation from distribution center j to customer k
x_{ij}	the number of units transported from manufacturer i to distribution center j
y_{jk}	the number of units transported from distribution center j to customer k
Z	the total cost of the distribution solution
out_i	the number of units delivered by manufacturer i
ind_j	the number of units supplied to distribution center j
inc_k	the number of units supplied to customer k

Given a set of p manufacturers, a set of q distribution centers (DC's) and a set of r customers with the following properties:

- Each manufacturer may ship to any of the q distribution centers at a transportation cost b_{ij} per unit from manufacturer i, where $i \in \{1, ..., p\}$, to DC j, where $j \in \{1, ..., q\}$, plus a fixed-charge f_{ij} for operating the route.
- Each DC may ship to any of the r customers at a transportation cost c_{jk} per unit from DC j,where $j \in \{1, ..., q\}$, to customer k, where $k \in \{1, ..., r\}$, plus a fixed-charge g_{jk} for operating the route.

- Each manufacturer $i \in \{1, ..., p\}$ has S_i units of supply and each customer $k \in \{1, ..., r\}$ has a demand D_k.

The aim of the two-stage fixed-cost transportation problem is to determine the routes to be opened and corresponding shipment quantities on these routes, such that the customer demands are fulfilled, all shipment constraints are satisfied, and the total distribution costs are minimized.

By introducing the linear variables:

x_{ij} representing the number of units transported from manufacturer i to DC j,
y_{jk} representing the number of units transported from DC j to customer k,

and the binary variables:

z_{ij} is 1 if the route from manufacturer i to DC j is used and 0 otherwise,
w_{jk} is 1 if the route from DC j to customer k is used and 0 otherwise,

then the two-stage transportation problem with fixed costs associated to the routes can be modeled as the following mixed integer problem described by Raj and Rajendran [22]:

$$\min \quad \sum_{i=1}^{p}\sum_{j=1}^{q}(b_{ij}x_{ij} + f_{ij}z_{ij}) + \sum_{j=1}^{q}\sum_{k=1}^{r}(c_{jk}y_{jk} + g_{jk}w_{jk})$$

$$\text{(2.1)} \qquad s.t. \quad \sum_{j=1}^{q} x_{ij} \leq S_i, \ \forall\, i \in \{1, ..., p\}$$

$$\text{(2.2)} \qquad \sum_{j=1}^{q} y_{jk} = D_k, \ \forall\, k \in \{1, ..., r\}$$

$$\text{(2.3)} \qquad \sum_{i=1}^{p} x_{ij} = \sum_{k=1}^{r} y_{jk}, \ \forall\, j \in \{1, ..., q\}$$

$$\text{(2.4)} \qquad x_{ij} \geq 0, \ \forall\, i \in \{1, ..., p\}, \ \forall\, j \in \{1, ..., q\}$$

$$\text{(2.5)} \qquad y_{jk} \geq 0, \ \forall\, j \in \{1, ..., q\}, \forall\, k \in \{1, ..., r\}$$

$$\text{(2.6)} \qquad z_{ij} \in \{0, 1\}, \ \forall\, i \in \{1, ..., p\}, \ \forall\, j \in \{1, ..., q\}$$

$$\text{(2.7)} \qquad w_{jk} \in \{0, 1\}, \ \forall\, j \in \{1, ..., q\}, \forall\, k \in \{1, ..., r\}$$

The objective function minimizes the total distribution cost: the fixed costs and transportation per-unit costs. Constraints (2.1) guarantee that the quantity shipped out from each manufacturer does not exceed the available capacity, constraints (2.2) guarantee that the total shipment received from DCs by each customer is equal to its demand and constraints (2.3) are the flow conservation conditions and they guarantee that the units received by a DC from manufacturers are equal to the units shipped from the distribution centers to the customers. The last four constraints (2.4)-(2.7) ensure the integrality and non-negativity of the decision variables.

The presence of the fixed-charge associated to the routes makes the problem more difficult. Hirsch and Dantzig [9] showed that even the fixed-charge transportation problem is an NP-hard problem and therefore our considered two-stage fixed-cost transportation problem which is an extension of the FCTP is as well NP-hard. Since the considered transportation problem is NP-hard, the necessary computational time in order to obtain an exact solution of the problem increases in an exponential manner and very rapidly becomes extremely difficult and long as the size of the problem increases. That is why in order to tackle the two-stage fixed-cost transportation problem we proposed an efficient multi-start Iterated Local Search algorithm.

An illustration of the investigated two-stage fixed-charge transportation problem is presented in the next figure.

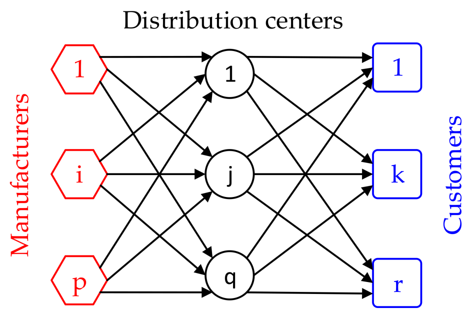

FIGURE 1. Illustration of the two-stage fixed-charge transportation problem

3. DESCRIPTION OF THE ITERATED LOCAL SEARCH HEURISTIC

The following terms and abbreviations will be used in the description of our proposed algorithm:

Manufacturer i will be called *not depleted*, if $s_i - out_i > 0$.

Customer k will be called *completed*, if $d_k - ink = 0$

A transportation link from manufacturer i to DC j, denoted L_{ij}, will be called nonzero, if $x_{ij} > 0$.

A transportation link from DC j to customer k, denoted L_{jk} will be called nonzero, if $y_{jk} > 0$

A route R_{ijk} is a set of two transportation links: L_{ij} and L_{jk}. A route R_{ijk} connects manufacturer i to customer k through distribution center j. The capacity of route R_{ijk} , denoted xy_{ijk}, is the smallest value of x_{ij} and y_{jk}, i.e. $xy_{ijk} = min\{x_{ij}, y_{jk}\}$. Route R_{ijk} will be called *nonzero* if $xy_{ijk} > 0$.

The operation of reserving a units on the link L_{ij} is defined as follows:

$reserve(link\ L_{ij},\ amount\ a)$
1. $x_{ij} \leftarrow x_{ij} + a$
2. $out_i \leftarrow out_i + a$
3. $ind_j \leftarrow ind_j + a$

The operation of canceling a units from the link L_{ij} is defined as follows:

$cancel(link\ L_{ij},\ amount\ a)$
1. $reserve(L_{ij}, -a)$

The operation of reserving a units on the route R_{ijk} is defined as follows:

$reserve(route\ R_{ijk},\ amount\ a)$
1. $reserve(L_{ij}, a)$
2. $y_{jk} \leftarrow y_{jk} + a$
3. $inc_k \leftarrow inc_k + a$
4. $ind_j \leftarrow ind_j - a$

The operation of canceling a units from the route R_{ijk} is defined as follows:

$cancel(route\ R_{ijk},\ amount\ a)$
1. $reserve(R_{ijk}, -a)$

The algorithm consists of a multi-start heuristic, mainly based on an Iterated Local Search (ILS). Initial solutions are constructed using a minimum cost procedure applied for each customer, combined with a quick neighborhood operator, while local search is performed using a powerful neighborhood operator. When a solution is trapped in a local optimum, a perturbation mechanism is applied that builds a new solution from scratch.

The main module of our ILS executes a fixed number of iterations which was previously established based on the number of customers. At each iteration, the customers are placed in a random order that has not been used before. For this purpose, we used the Fischer-Yates shuffle algorithm and the resulting order is placed in a hash set, so that its uniqueness can be efficiently checked. Next an initial solution is built, supplying the customers in the order previously set. Then the initial solution is enhanced by several local search operations, that continue as long as the solution improves. When improvement is no longer possible, a new solution is being built, processing the customers in a different order.

The process of supplying customer k involves the following operations:

$supply(customer\ k)$
1. $Solve\ the\ problem\ defined\ by\ relations$ (3.8) to (3.12), $resulting\ the\ pair\ j, i\ and\ the\ available\ quantity\ \tilde{d}_k$
2. $perform\ a\ reserve(R_{ijk}, \tilde{d}_k)\ operation$
3. $if\ inc_k < d_k,\ then\ continue\ with\ step$ 1

$$\min_{j,i}(c_{jk} + \tilde{g}_{jk} + b_{ij} + \tilde{f}_{ij}) \tag{3.8}$$

$$\tilde{g}_{jk} = \begin{cases} 0, y_{jk} \neq 0 \\ \frac{g_{jk}}{\tilde{d}_k}, y_{jk} = 0 \end{cases} \tag{3.9}$$

$$\tilde{f}_{ij} = \begin{cases} 0, x_{ij} \neq 0 \\ \frac{f_{ij}}{\tilde{d}_k}, x_{ij} = 0 \end{cases} \tag{3.10}$$

$$\tilde{d}_k = min\{s_i - out_i, d_k - inc_k\} \tag{3.11}$$

$$\tilde{d}_k > 0. \tag{3.12}$$

The demand of each customer k is satisfied by solving the problem defined by relations (3.8) to (3.12). The result is the j, i pair and the available amount $\tilde{d}_k$. Then, a $reserve(R_{ijk}, \tilde{d}_k)$ operation is performed. If after the first iteration $inc_k < d_k$, then satisfying the demand of customer k requires more iterations which are carried out as described above. Relations (3.11) and (3.12) guarantee that only un depleted manufacturers will be taken into account. The process finishes when the customer becomes completed.

If there were no fixed costs for opening transport routes and if the production capacities of the manufacturers were not limited, the initial solution building process described above would find the optimal solution in the first attempt. But this is unlikely to happen, because of the distribution system restrictions.

After ensuring the demand of each customer, the distribution system changes as part of the manufacturers production was consumed and new transport routes could be opened. A quick neighborhood operator was designed to verify whether previously established routes can be improved under the new conditions. The quick search is performed when the processing of a customer finishes. Experiments showed that for less complex distribution systems, this step can significantly increase the efficiency of the optimal solution search process.

The quick search operator systematically destroys older parts of the solution and then rebuilds them according to the current configuration of the distribution system, seeking a better alternative. Previously established routes are canceled first, and then they are rebuilt. All nonzero transportation routes R_{ijk} are reconsidered as follows:

$quickSearchMove(route\ R_{ijk})$
1. $a \leftarrow xy_{ijk}$
2. $cancel(R_{ijk}, a)$
3. $reserve(R_{ijk}, a)$.

Customers are processed in the same order they were processed in the initial solution building process. The first reconsidered routes are the ones connected to the first supplied customer. Since usually there is a relatively small number of nonzero routes, the quick search operator works fast and can be called after the supply of each customer finishes. The number of operations involved in quick search increase as the number of supplied customers increases. If the quick search does not improve the solution, all the changes are abandoned, thus restoring the previous solution.

In Figures 2 and 3 we present a distribution system under construction, and the possible effect of a $quickSearchMove$ operation. Within the nodes we marked the quantities available at the manufacturers s_i and the demands of the customers d_k and on the arcs, we marked the number of transported units x_{ij} and y_{jk}. The parameter R_{ijk} of the $quickSearchMove$ operation is shown in figure 2. Figure 3 shows the new R_{ijk} route discovered by $quickSearchMove$, and the final state of the distribution system.

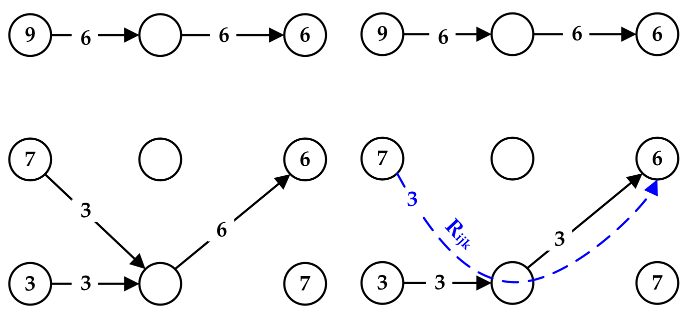

FIGURE 2. Illustration of a distribution system under construction and a route to be removed by the $quickSearchMove$ operation

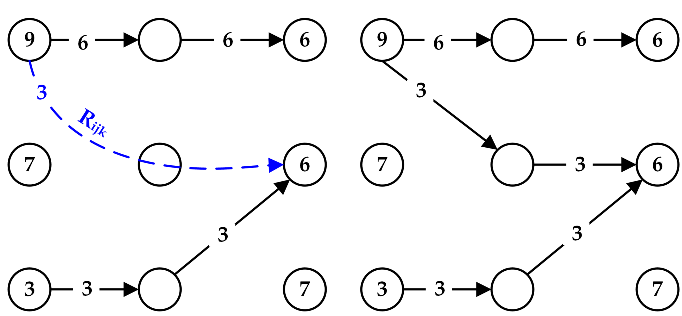

FIGURE 3. Illustration of a new alternative route discovered by the $quickSearchMove$ operation for the removed route and the new distribution system structure

The local search operator is useful in the case of higher complexity distribution systems, for which the quick search fails to find the optimal solution. The local search operator uses the same search principle as the quick search operator: destroy and rebuild. The difference lies in the fact that the local search operator is applied after the initial solution construction is finished, and it destroys larger portions of the solution. This increases the search area, thus increasing the probability of improving the solution.

Our proposed local search is an iterative process. At each iteration a nonzero link L_{ij} and two nonzero routes $R'_{i'j'k'}$ and $R''_{i''j''k''}$ are picked randomly. The selection process is designed so that all possible variants are tested only once. This operation is possible, because the number of nonzero links and routes is relatively small. Each selected set $(L_{ij}, R'_{i'j'k'}, R''_{i''j''k''})$ is processed as follows:

$localSearchMove(linkL_{ij}, routeR'_{i'j'k'}, routeR''_{i''j''k''})$
1. $a \leftarrow x_{ij};\ a' \leftarrow xy_{i'j'k'};\ a'' \leftarrow xy_{i''j''k''}$
2. $cancel(L_{ij}, a)$
3. $cancel(R'_{i'j'k'}, a')$
4. $cancel(R''_{i''j''k''}, a'')$
5. $reserve(R'_{i'j'k'}, a')$
6. $reserve(R''_{i''j''k''}, a'')$
7. $solve\ the\ problem\ defined\ by\ relations$ (3.13) to (3.16). $The\ result\ is\ the\ manufacturer\ identification\ number\ i,\ and\ the\ supported\ quantity\ \tilde{a}$
8. $reserve(L_{ij}, \tilde{a})$
9. $if\ ind_j\ <\ 0\ then\ continue\ with\ step$ 7
10. $if\ the\ solution\ worsened,\ then\ restore\ the\ initial\ solution.$

$$\min_i (b_{ij} + \tilde{f}_{ij}) \tag{3.13}$$

$$\tilde{f}_{ij} = \begin{cases} 0, x_{ij} \neq 0 \\ \frac{f_{ij}}{\tilde{a}}, x_{ij} = 0 \end{cases} \tag{3.14}$$

$$\tilde{a} = min\{s_i - out_i,\ -ind_j\} \tag{3.15}$$

$$\tilde{a} > 0. \tag{3.16}$$

Each localSearchMove operation uses 3 parameters: A nonzero link L_{ij} and two nonzero routes $R'_{i'j'k'}$ and $R''_{i''j''k''}$. After canceling the link L and the two routes R' and R'', the two routes are reserved again. The new reservations could find better alternatives in the new conditions. The cancellation of the L_{ij} link aims to increase the likelihood of finding a more advantageous variant for replacing R' and R'' routes.

However, the cancellation of this link has diminished the number of units entering DC j, without changing the number of units leaving this DC. Thus ind_j becomes negative. Next, the most advantageous variant for correcting the solution is sought as follows:

The number of units with which the stock of DC j decreased after cancelling the L_{ij} link is -ind_j. The correction involves solving the previously defined optimization problem. If -$ind_j \leq s_i$-out_i , then the correction finishes in one step. Otherwise, more iterations are required to complete the initial stock of DC j.

Because the order in which customer demands are resolved is decisive for the result, it is useful to perform an extra $localSearchMove$ operation with the two routes in reversed order.

In figures 4 and 5 we show the way a solution can be modified by the $localSearchMove$ operation. Figure 4 presents an initial solution, a link L_{ij} and two routes $R'_{i'j'k'}$ and

$R''_{i''j''k''}$ that could be picked for a $localSearchMove$ operation. Figure 5 shows the new discovered routes, the necessary correction and the final distribution system. Such a transformation cannot be accomplished by quick search moves. Therefore, local search moves can find solutions that the quick search moves cannot reach.

The local search operations can generate a significant number of duplicate solutions. For the efficient removal of duplicates, a hash code is calculated for each new solution, that is stored in a hash set. However, this technique cannot be applied in the case of large instances.

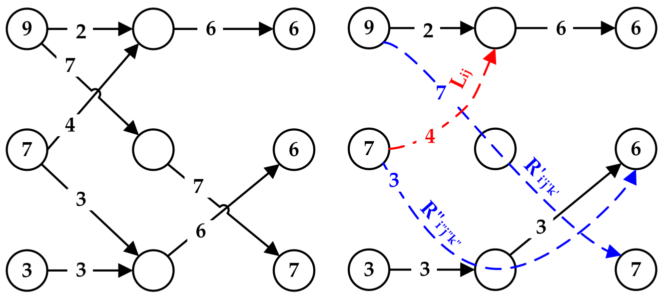

FIGURE 4. Illustration of an initial solution, a link and two routes that are transmitted to the $localSearchMove$ operation

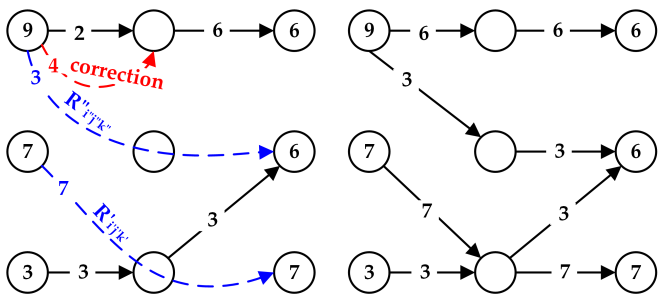

FIGURE 5. Illustration of the new routes and the new solution of the problem

4. Computational results

In order to analyze the performance of our proposed ILS heuristic algorithm, we tested it on two sets of instances. The first one was generated by Jawahar and Balaji [10] and it consists of 20 test problems of small sizes. The second set of problems contains 10 new randomly generated instances of larger sizes. The files of the two sets of 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 3.3GHz, 4GB RAM, Windows 10 Education 64 bit operating system.

Table 1 presents the computational results of our proposed heuristic ILS algorithm in comparison with the genetic algorithm described by Jawahar and Balaji [10], called JRGA, the two genetic algorithms introduced by Raj and Rajendran [22], denoted by TSGA and SSGA and the hybrid genetic algorithm (HGA) described by Pop et al. [21]. 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 [10], the two genetic algorithms introduced by Raj and Rajendran [22], the hybrid genetic algorithm described by Pop et al. [21] and our ILS heuristic algorithm and the rounded average number of evaluations necessary to obtain it. The results written in bold represent cases for which the obtained solution is the best existing from the literature.

TABLE 1. Computational results achived by our proposed ILS compared to existing approaches.

Instance	JRGA		TSGA	SSGA		HGA		ILS	
size	obj	#eval	obj	obj	#eval	obj	#eval	obj	#eval
2 x 2 x 3	**112600**	1444	**112600**	**112600**	637	**112600**	2	**112600**	2
2 x 2 x 4	**237750**	1924	**237750**	**237750**	857	**237750**	2	**237750**	2
2 x 2 x 5	**180450**	2404	**180450**	**180450**	1214	**180450**	319	**180450**	6
2 x 2 x 6	**165650**	2884	**165650**	**165650**	1354	**165650**	324	**165650**	12
2 x 2 x 7	**162490**	3364	**162490**	**162490**	1889	**162490**	335	**162490**	16
2 x 3 x 3	**59500**	2164	**59500**	**59500**	1503	**59500**	317	**59500**	5
2 x 3 x 4	**32150**	2884	**32150**	**32150**	1859	**32150**	339	**32150**	9
2 x 3 x 6	69970	4324	67380	**65945**	2577	**65945**	356	**65945**	5
2 x 3 x 8	263000	5764	**258730**	**258730**	5235	**258730**	546	**258730**	341
2 x 4 x 8	80400	7684	84600	**77400**	5246	78550	1039	**77400**	28
2 x 5 x 6	94565	7204	80865	**75065**	3574	80865	430	**75065**	3
3 x 2 x 4	**47140**	2884	**47140**	**47140**	1429	**47140**	321	**47140**	6
3 x 2 x 5	178950	3604	178950	**175350**	2061	178950	320	**175350**	25
3 x 3 x 4	**57100**	4324	61000	**57100**	3060	**57100**	354	**57100**	12
3 x 3 x 5	**152800**	5404	156900	**152800**	4555	**152800**	335	**152800**	3
3 x 3 x 6	**132890**	6484	**132890**	**132890**	2981	**132890**	3	**132890**	3
3 x 3 x 7 (a)	104115	7564	106745	**99095**	7095	103815	1330	**99095**	78
3 x 3 x 7 (b)	287360	7564	295060	**281100**	7011	**281100**	380	**281100**	15
3 x 4 x 6	77250	8644	81700	**76900**	7105	77250	373	**76900**	22
4 x 3 x 5	**118450**	7204	**118450**	**118450**	4227	**118450**	394	**118450**	19

Analyzing the computational results reported in Table 1, we can observe that our ILS approach achieved for all the instances the optimal solution, obtained by using CPLEX 12.7 to solve exactly the mathematical model. Our algorithm has a better computational performance compared to JRGA [10] and TSGA [22] and compared to the SSGA [22], it delivers the same solution in all the 20 considered instances, but the number of solution evaluations is significantly lower compared to the number of solutions enumerated to obtain the corresponding solutions by the SSGA approach. In comparison with the HGA [21], our ILS heuristic algorithm achieved better solution in 5 out of 20 instances and the same solutions for the remaining instances. Regarding the number of solution evaluations, our approach uses considerable less in 17 out of 20 instances and equal in the case of the other 3 instances. We would like to emphasize the fact that each achieved solution was obtained in under 1ms time.

Due to the small sizes of the benchmark instances considered in the literature, in Table 2 we present the computational results achieved by our proposed heuristic ILS algorithm in the case of the new randomly generated instances of larger sizes. The first column contains the size of the instances and the next four columns contain the following results achieved by our ILS heuristic algorithm: the best solution, the average solution, the average computational time in seconds and the average numbers of evaluations necessary to obtain it.

TABLE 2. Computational results achieved by our proposed ILS in the case of the new randomly generated instances of larger sizes

Instance size	ILS			
	Best sol.	Avg. sol.	Avg. time	Avg. eval.
10 x 15 x 20	232036	232036	93.95	65558.52
10 x 16 x 25	299089	299089	288.82	167533.81
10 x 20 x 30	302575	302575	58.82	19853.43
10 x 20 x 40	411915	411915	1.08	252.75
10 x 25 x 50	511980	511990	4593.41	851928.64
10 x 25 x 60	558094	558094	332.94	53312.22
10 x 30 x 60	508558	508558	5.33	807.78
10 x 30 x 70	585378	585378	21.70	3274.26
10 x 30 x 90	767341	767341	7415.09	934253.60
10 x 30 x 100	799825	804548	7314.74	792530.24

The achieved computational results presented in Table 2 were obtained by performing 10 independent runs for each randomly generated instance. Analyzing the computational results reported in Table 2, we can observe that in 8 out 10 instances the obtained best solutions coincide with the achieved average solutions, proving the robustness of our proposed ILS heuristic algorithm.

5. CONCLUSIONS

This paper proposes an efficient multi-start Iterated Local Search (ILS) procedure for solving the two-stage fixed-charge transportation problem which models an important transportation systems design from manufacturers to customers through distribution centers. Our heuristic approach constructs an initial solution, uses a local search procedure to increase the exploration, a perturbation mechanism and a neighborhood operator in order to diversify the search. Computational results on two sets of instances show that our iterated local search algorithm is robust and compares favorably to existing approaches. The novel solution approach provides optimal solutions in run-times of under one millisecond for the 20 benchmark instances available in the literature and high-quality solutions for the new proposed larger instances.

In the future, we plan to use our code as the basis for a parallel implementation and to test our ILS heuristic algorithm on larger instances.

REFERENCES

[1] Amiri, A., *Designing a distribution network in a supply chain system: Formulation and efficient solution procedure*, European Journal of Operational Research, **171** (2006), No. 2, 567–576

[2] Calvete, H., Galé, C. and Iranzo, J., *Planning of a decentralized distribution network using bilevel optimization*, Omega, **49** (2014), 30–41

[3] Calvete, H., Gale, C. and Iranzo, J., *An improved evolutionary algorithm for the two-stage transportation problem with fixed charge at depots*, OR Spectrum, **38** (2016), 189–206

[4] Chen, S., Zheng, Y., Cattani, C. and Wang, W., *Modeling of Biological Intelligence for SCM System Optimization*, Computational and Mathematical Methods in Medicine, Volume **2012**, Article ID 769702, (2012), 10 pp.

[5] Cosma, O., Pop, P. C., Matei, O. and Zelina, I., *A hybrid iterated local search for solving a particular two-stage fixed-charge transportation problem*, in Proc. of HAIS 2018, Lecture Notes in Computer Science, **10870** (2018), 684-693

[6] El-Sherbiny, M. M., *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*, Expert Systems with Applications, **39** (2012), 11321–11322

[7] Gen, M., Altiparmak, F. and Lin, L., *A genetic algorithm for two-stage transportation problem using priority based encoding*, OR Spectrum, **28** (2006), 337–354

[8] Geoffrion, A. M. and Graves, G. W., *Multicommodity distribution system design by Benders decomposition*, Management Science, **20** (1974), 822–844

[9] Hirsch, W. M. and Dantzig, G. B., *The fixed charge problem*, Naval Research Logistics Quarterly, **15** (1968), 413—424

[10] Jawahar, N. and Balaji, A. N., *A genetic algorithm for the two-stage supply chain distribution problem associated with a fixed charge*, European Journal of Operational Research, **194** (2009), 496–537

[11] Jayaraman, V. and Pirkul, H., *Planning and coordination of production and distribution facilities for multiple commodities*, European Journal of Operational Research, **133** (2001), No. 2, 394–408

[12] Marin, A. and Pelegrin, B., *A branch-and-bound algorithm for the transportation problem with locations p transshipment points*, Computers & Operations Research, **24** (1997), No. 7, 659–678

[13] Marin, A., *Lower bounds for the two-stage uncapacitated facility location problem*, European Journal of Operational Research, **179** (2007), No. 3, 1126–1142

[14] Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Kesteli, M. and Tavakkoli-Moghaddam, R., *Solving a capacitated fixed-cost transportation problem by artificial immune and genetic algorithms with a Prüfer number representation*, Expert Systems with Applications, **38** (2011), 10462–10474

[15] Pintea, C.-M., Pop Sitar, C., Hajdu-Măcelaru, M. and Pop, P. C., *A Hybrid Classical Approach to a Fixed-Charge Transportation Problem*, In Proc. of HAIS 2012, Part I, Editors E. Corchado et al., Lecture Notes in Computer Science, **7208** (2012), 557–566

[16] Pintea, C.-M., Pop, P. C. and Hajdu-Măcelaru, M., *Classical Hybrid Approaches on a Transportation Problem with Gas Emissions Constraints*, Advances in Intelligent and Soft Computing, **188** (2013), 449–458

[17] Pintea, C. M. and Pop, P. C., *An improved hybrid algorithm for capacitated fixed-charge transportation problem*, Logic Journal of IJPL, **23** (2015), No. 3, 369–378

[18] Pirkul, H. and Jayaraman, V., *A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution*, Computers & Operations Research, **25** (1998), No. 10, 869–878

[19] Pop, P. C., Pintea, C.-M., Pop Sitar, C. and Hajdu-Măcelaru, M., *An efficient Reverse Distribution System for solving sustainable supply chain network design problem*, J. Appl. Log., **13** (2015), No. 2, 105–113

[20] Pop, P. C., Matei, O., Pop Sitar, C. and Zelina, I., *A hybrid based genetic algorithm for solving a capacitated fixed-charge transportation problem*, Carpathian J. Math., **32** (2016), No. 2, 225–232

[21] Pop, P. C., Sabo, C., Biesinger, B., Hu, B. and Raidl, G.,*Solving the two-stage fixed-charge transportation problem with a hybrid genetic algorithm*, Carpathian J. Math., **33** (2017), No. 3, 365–371

[22] Raj, K. A. A. D. and Rajendran, C., *A genetic algorithm for solving the fixed-charge transportation model: Two-stage problem*, Computers & Operations Research, **39** (2012), No. 9, 2016–2032

[23] Raj, K. A. A. D. and Rajendran, C., *A Hybrid Genetic Algorithm for Solving Single-Stage Fixed-Charge Transportation Problems*, Technology Operation Management, **2** (2011), No. 1, 1–15

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

[1] Department of Mathematics and Computer Science
North University Center at Baia Mare
Technical University of Cluj-Napoca
Victoriei 76, 430122 Baia Mare, Romania
Email address: petrica.pop@cunbm.utcluj.ro
Email address: ovidiu.cosma@yahoo.com

[2] Department of Economics
North University Center at Baia Mare
Technical University of Cluj-Napoca
Victoriei 76, 430122 Baia Mare, Romania
Email address: corina.pop.sitar@cunbm.utcluj.ro