CARPATHIAN J. MATH. 32 (2016), No. 2, 225 - 232
Online version at http://carpathian.ubm.ro
Print Edition: ISSN 1584 - 2851 Online Edition: ISSN 1843 - 4401

# A hybrid based genetic algorithm for solving a capacitated fixed-charge transportation problem

PETRICA POP [1], OLIVIU MATEI [2], CORINA POP SITAR [3] and IOANA ZELINA [1]

ABSTRACT. This paper is focusing on an important transportation application encountered in supply chains, namely the capacitated two-stage fixed-charge transportation problem. For solving this complex optimization problem we 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. The proposed algorithm was implemented and tested on an often used collection of benchmark instances and the computational results obtained showed that our proposed hybrid heuristic algorithm delivered competitive results compared to the state-of-the-art algorithms for solving the considered capacitated two-stage fixed-charge transportation problem.

## 1. INTRODUCTION

Supply chains (SCs) are universal networks containing the following individuals: suppliers, manufacturers, distribution centers, retailers and customers. The classic SC accomplishes the functions of acquisition of raw materials, transformation of those into intermediate and finished products and finally the distribution of the resulted products to customers and its main goal is to fulfill the customer requirements [13].

Supply chain management (SCM) concerns the management of the flow of products starting from suppliers and ending to customers. SCM is an important and crucial process for many companies, and many companies are struggling to achieve an optimized supply chain because this translates to lower costs for the company.

Network design is playing an important and central role in realizing an efficient and effective management of supply chain systems. Usually, the supply chains can be modeled and represented as a form of multi-stage based structure, whose optimal design has been recognized as an NP-hard optimization problem [4].

The fixed cost transportation problems are natural extensions of the classical transportation problem described for the first time by Schaffer and O'leary [17]. These problems have been motivated by the real world applications and their main characteristics are presence of two kinds of costs: the distribution costs and the fixed charge costs. At the beginning there have been considered single-stage problems and nowadays multi-stage distribution problems are investigated.

The classical transportation problem is static in the sense that all the information relevant is known apriori, before the process begins. The dynamic version of the problem is a transportation problem over time. A good survey on these problems was provided by Bookbinder and Sethi [3]. A different dynamic transportation problem was studied by Lupşa et al. [8]. Some other dynamic optimization problems have been investigated in [14, 15].

The two-stage transportation problems have been introduced by Geoffrion and Graves [6] in 1974, but even nowadays these complex transportation problems are representing

Received: 28.03. 2015. In revised form: 22.10.2015. Accepted: 04.11.2015
2010 *Mathematics Subject Classification.* 90B06, 92D10, 05C85.
Key words and phrases. *Fixed-charge transportation problem, genetic algorithms, hybrid algorithms* .
Corresponding author: Petrica Pop; petrica.pop@cunbm.utcluj.ro

a challenging research area. Since then several solution approaches based on exact and heuristic algorithms have been proposed such as the tabu search approach described by Sun et al. [18], a spanning tree-based genetic algorithm presented by Syarif et al. [19], a genetic algorithm described by Raj and Rajendran [1], etc.

In the current paper, we are focusing on a particular supply chain network design problem, namely the capacitated fixed-cost transportation problem in a two-stage supply chain network involving one manufacturer, a set of distribution centers (DC's) and a set of customers and which consists on opening an optimal number of DC's and finding the distribution routes in order to meet the specific demands from customers such that the total transportation costs are minimized. In this form, the problem was introduced by Molla-Alizadeh-Zavardehi et al. [11]. The same authors presented as well an integer programming model of the problem and proposed a spanning tree-based genetic algorithm with a Prüfer number representation and an artificial immune algorithm for solving it. Some comments concerning the mathematical model of the problem were published by El-Sherniny [5]. Recently, Pintea and Pop [13] developed an improved hybrid algorithm combining the Nearest Neighbor search heuristic with a powerful local search procedure, which was tested on a collection of benchmark instances and in a preliminary version, Pintea et al. [12] described some hybrid classical approaches for solving the problem and Pop et al. [16] proposed an efficient reverse distribution system for solving the problem.

Our paper is organized as follows: in the second section we define the capacitated fixed-cost transportation problem in a two-stage supply chain network with one manufacturer. Section 3 describes our novel developed hybrid based genetic algorithm for solving the problem. The proposed algorithm is applied in Section 4 to a set of benchmark instances taken from Pintea and Pop [13] and the obtained results are presented and analyzed. Finally, in the last section, we summarize the obtained results in this paper and future research directions are presented.

## 2. Definition of the fixed-charge transportation problem

The considerd capacitated fixed-cost transportation problem in a two-stage supply chain network is defined as follows: given a manufacturer, a set of $m$ distribution centers (DC's) and a set of $n$ customers satisfying the following properties:

- the manufacturer can ship to any distribution center at a transportation cost $c_i$, $i \in \{1, ..., m\}$,
- each DC can ship to any customer at a transportation cost $c_{ij}$ from DC $i \in \{1, ..., m\}$ to customer $j \in \{1, ..., n\}$, plus a fixed-cost $f_{ij}$ for operating the route,
- the opening costs for a potential DC $i$ are denoted by $f_i$, $i \in \{1, ..., m\}$,
- the manufacturer has a given number units of supply, each DC $i \in \{1, ..., m\}$ has $SC_i$ units of stocking capacity and each customer $j \in \{1, ..., n\}$ has a demand $D_j$,

we want to determine which DC's and routes are going to be opened and the size of the shipments on those routes such that the total distribution costs satisfying the supply constraints in order to meet the demands of the customers is minimized.

An illustration of the considered fixed-cost transportation problem is presented in the next figure.

## 3. The hybrid genetic algorithm

In this section, we describe our novel hybrid genetic based algorithm for solving the considered capacitated fixed-charge transportation problem.

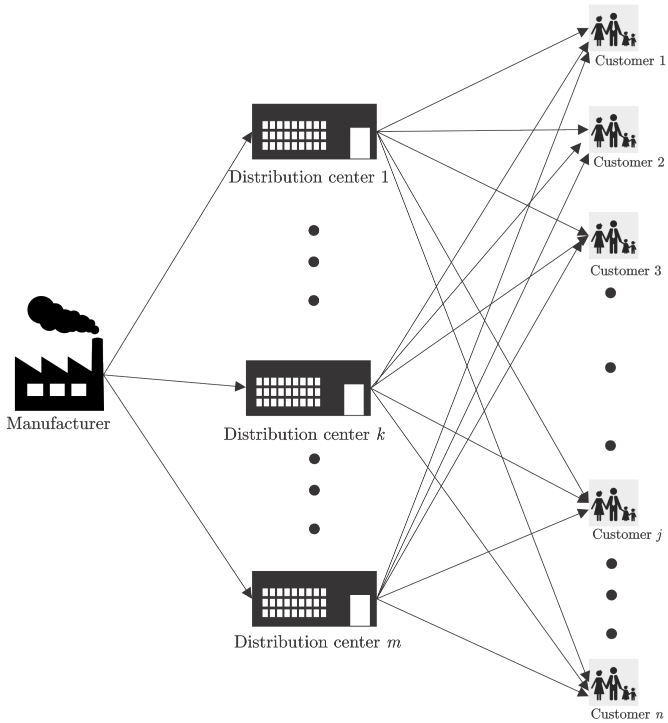


FIGURE 1. Illustration of the two-stage supply chain network design [13]

3.1. **Genetic Representation.** As the manufacturer is always the same and brings no intrinsic added value, actually only the distribution centers (DC's) and the customers need to be modeled in an individual, therefore a straightforward representation is a hash table as defined in [20] and depicted in Figure 2, in which the keys are the DC's and the values are the customers. Each customer is allocated to a DC. If a key has no associated values, it means that specific DC is not selected, therefore the costs associated with it are null.

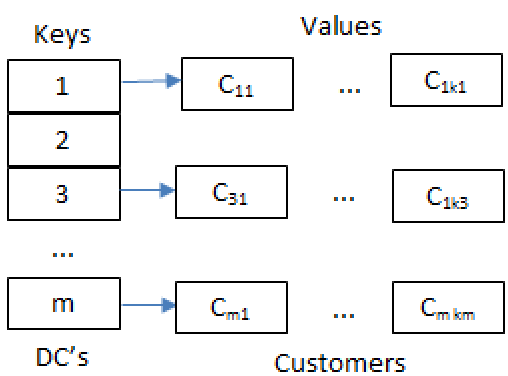


FIGURE 2. The hash table representation of an individual

Mathematically, an individual can be denoted as:

(3.1)
$$I = (DC_1(C_{11}, C_{12}, ..., C_{1k_1}), DC_2(C_{21}, C_{22}, ..., C_{2k_2}),\ ...,\ DC_n(C_{m1}, C_{m2}, ..., C_{mk_m}))$$
where by $C_{pq}$, $p \in \{1, ..., m\}$, $q \in \{k_1, ..., k_m\}$ we denoted the customers served by a specific DC and $k_1 + ... + k_m = n$.

Another possible representation of an individual is as a tree with the depth three, where the root is the manufacturer, the second level consists of the distribution centers (DC) and the leaves are the customers. The representation is very similar to the one proposed by Koza in [7]. In Java, the implementation of an individual is a class implementing the `TreeModel`. Another implementation is based on Prüfer numbers, as proposed by Molla et al in [11]. Of course, for all individuals, the root is the same, therefore all the genetic operators apply only from the root downwards. The main drawback of this representation is that each individual carries a redundant root, with absolutely no added value. On the other hand, the genetic operators for tree individuals are well defined in [7].

3.2. **Genetic operators. Crossover**

Two parents are selected from the population by the binary tournament method, i.e. the individuals are chosen from the population at random and undergo recombination (crossover).

Offspring are produced from two parent solutions using the following classic crossover procedure (see for example [10]). It is implemented by selecting a random cut point between the keys (DC's). The first offspring is made of the first part of the first parent, respectively the second part of the second parent. The other offspring is made of the second sequence of the first parent, respectively the first sequence of the first parent.

Given the two parents:

$$P_1 = \left(DC_1(C^1_{11}, ..., C^1_{1k_1}), DC_2(C^1_{21}, ..., C^1_{2k_2}), |\ ...,\ DC_m(C^1_{m1}, ..., C^1_{mk_m})\right) \tag{3.2}$$
$$P_2 = \left(DC_1(C^2_{11}, ..., C^2_{1k_1}), DC_2(C^2_{21}, ..., C^2_{2k_2}), |\ ...,\ DC_m(C^2_{m1}, ..., C^2_{mk_m})\right) \tag{3.3}$$

and the cutting point defined by "|", the offspring are:

$$O_1 = \left(DC_1(C^1_{11}, ..., C^1_{1k_1}), DC_2(C^1_{21}, ..., C^1_{2k_2}),\ ...,\ DC_m(C^2_{n1}, ..., C^2_{mk_m})\right) \tag{3.4}$$
$$O_2 = \left(DC_1(C^2_{11}, ..., C^2_{1_k 1}), DC_2(C^2_{21}, ..., C^2_{2_k 2}),\ \ ...,\ DC_m(C^1_{m1}, ..., C^1_{mk_m})\right) \tag{3.5}$$

If any of the customers appears twice in one of the offspring, one of them is discarded at random.

We emphasis the fact that the cutting point refers only the DC's, not to the customers. This means that the crossover operator just changes the associations between the DC's and the customers.

**Mutation**

The mutation is implemented as a swap of a random number of customers served by a specific distribution center with a random number of customers served by another distribution center. If the following distribution centers are selected to undergo mutation $DC_r(C_{r1}, C_{r2}, C_{r3}, ..., C_{rk_r})$ and $DC_s(C_{r1}, C_{r2}, C_{r3}, ..., C_{rk_r})$ and the two random numbers are $R_r$ and $R_s$, then $R_r$ values of the $DC_r$ are moved to $DC_s$ and $R_s$ customers of $DC_s$ are moved to $DC_r$. The new number of customers of $DC_r$ is $|DC_r| = k_r - R_r + R_s$, whereas the new number of customers of $DC_s$ is $|DC_s| = k_s - R_s + R_r$. The values of the other keys (DC's) are not affected.

There are several specific cases:

- if $R_r = R_s$ then the number of the associated customers is unchanged after mutation;
- if the number of one of the DC's is null, e.g. $|DC_s| = 0$, then there is a chance that after mutation $|DC_r| = 0$, but not necessarily.

Of course, for a valid mutation, $R_i < k_i$, where $R_i$ is the random number of customers to be moved out of the $DC_i$ and $k_i$ is the number of customers associated to $DC_i$.

**Selection**

The selection process is deterministic. Usually there are two approaches, called $(\mu, \lambda)$, respectively $(\mu + \lambda)$. In both cases, $\mu$ parents produce $\lambda$ offspring. However, in the former scenario, the next generation is constituted from the best $\mu$ individuals out of the $\lambda$ offspring, which means that the parents die out after one epoch. The latter scenario, which has been used also in our research, assumes that the parents and the offspring form a pool of individuals out of which the best $\mu$ are selected to form the next generation. The advantages of this process is that each population inherits the gain of the previous generation. On the other hand, there is a high risk that the population gets stuck in local optima. For avoiding this, when the next generation is created, only distinct individuals are selected, so that the new population will have all individuals different. The method has proved its efficiency in [9].

TABLE 1. The experimental results

| Replication | Number of DC's | Number of customers | Hybrid algorithm [13] | Proposed Genetic algorithm | | | Proposed Hybrid algorithm | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Best values | Average values | Std. deviation | Best values | Average values | Std. deviation |
| 1 | 10 | 10 | 21980 | 20450 | 21430 | 564,56 | **20400** | 21320 | 500,26 |
| 2 | 10 | 10 | 12160 | 11240 | 11850 | 401,09 | **11220** | 11740 | 398,33 |
| 3 | 10 | 10 | 14000 | 14100 | 14620 | 292,54 | **14040** | 14520 | 280,9 |
| 1 | 10 | 20 | 36000 | 35400 | 36200 | 461,27 | **35380** | 35860 | 240,83 |
| 2 | 10 | 20 | 39660 | 37840 | 38470 | 435,3 | **37800** | 38250 | 318,91 |
| 3 | 10 | 20 | 36060 | **36000** | 36110 | 118,27 | **36000** | **36000** | 0 |
| 1 | 10 | 30 | 55660 | 52700 | 54880 | 1085,67 | **52650** | 53700 | 606,1 |
| 2 | 10 | 30 | 55380 | 54650 | 55640 | 510,62 | **54540** | 54880 | 228,45 |
| 3 | 10 | 30 | 49860 | 48580 | 49470 | 699,1 | **48540** | 49240 | 431,92 |
| 1 | 15 | 15 | 26680 | **25420** | 27640 | 1052,22 | **25420** | 26710 | 819,16 |
| 2 | 15 | 15 | 29100 | **28600** | 29230 | 375,37 | **28600** | 28940 | 168,32 |
| 3 | 15 | 15 | 29200 | 28840 | 29470 | 338,24 | **28750** | 29120 | 218,87 |
| 1 | 50 | 50 | 92400 | 91550 | 92410 | 561,29 | **91500** | 92140 | 348,12 |
| 2 | 50 | 50 | 116500 | 114660 | 117440 | 2070,67 | **114150** | 115420 | 701,4 |
| 3 | 50 | 50 | **105000** | **105000** | 107400 | 1718,91 | **105000** | 106480 | 841,24 |

3.3. **The fitness function.** The fitness function is also the function to be optimized, namely the total cost of the distribution (fixed costs and the transportation costs).

3.4. **Genetic parameters.** It is well known that the genetic parameters are very important for the success of a GA, equally important as the other aspects, such as the representation of the individuals, the initial population and the genetic operators. Based on preliminary experiments, we have used the following parameter settings in our GA:

- the population size $\mu$ has been set to twice the number of DC's multiplied by the number of customers, therefore $\mu = 2 \cdot m \cdot n$.
- the intermediate population size $\lambda$ was chosen three times the size of the population: $\lambda = 3 \cdot \mu$.
- the mutation probability was set at $10\%$.
- the maximum number of epochs to run our GA was set to 1000.

3.5. **Local search.** In order to improve the quality of the obtained solutions by the GA, we consider a local search procedure. Our procedure consists on three LS operators introduced by Pintea and Pop [13] and applied sequentially:

1. **Insert DC.** This operator replaces a distribution center from the network with another one which was not open yet.
2. **Relocate DC.** Given two distribution centers $i_1, i_2 \in \{1, ..., m\}$, this operator interchanges the customers served by the DC $i_1$ with the the customers served by the DC $i_2$.
3. **Relocate customers.** Given two customers $j_1$ and $j_2$ served by the DC's $i_1$ and $i_2$, this operator assigns the customer $j_1$ to DC $i_2$ and the customer $j_2$ to DC $i_1$.

Our proposed hybrid algorithm stops when there are no improvements in the population over 15 consecutive generations or after the maximum number of epochs otherwise.

## 4. Computational results

In order to asses the effectiveness of our proposed hybrid genetic algorithm, we conducted our computational experiments on a set of 20 benchmark instances introduced by Pintea and Pop [13]. These instances were generated randomly as in [11], but unfortunately the data used by Molla-Alizadeh-Zavardehi et al. have not been available. Our algorithm has run 10 times on each instance and the best value was recorded, respectively the average and the standard deviation were computed.

The achieved results are summarized in Table 1. The first column represents the number of the instance (there are three for each combination of DC's and customers). Column *Hybrid algorithm* shows the best values reported by Pintea and Pop in [13]. The next six columns provide the results achieved by our proposed approaches: the genetic algorithm alone and the hybrid algorithm. The values in bold indicate the best existing solution with respect to that problem instance.

Analyzing the computational results reported on Table 1, we can observe that both proposed approaches compared favorably in terms of the provided quality solutions in comparison to the hybrid algorithm described by Pintea and Pop [13]: in 14 out of 15 instances improving the best values and for the last instance achieving the same best solution. As well, we remark that embedding the genetic algorithm with a local search procedure the resulted hybrid algorithm improved the quality of the solutions in 11 out of the 15 considered instances.

## 5. Conclusions

In this paper we consider a particular supply chain network design problem, namely the capacitated fixed-cost transportation problem in a two-stage supply chain network with one manufacturer. We provided a genetic algorithm for solving the problem, which was embedded with a local search procedure, obtaining in this way an efficient hybrid heuristic algorithm.

The computational results for an often used collection of benchmark instances provided by Pintea and Pop [13] show that our proposed approaches delivered competitive results compared to the state-of-the-art algorithms for solving the considered two-stage fixed-charge transportation problem. In addition, the developed hybrid algorithm obtained by incorporating within the GA a local search procedure, provides better solutions than those of the genetic algorithm alone.

In the future we plan to strengthen our developed hybrid heuristic algorithm by considering some other local search operators and in addition in order to asses its generality and scalability, we will test it on larger instances.

## REFERENCES


[1] Antony Arokia Durai Raj, K. and Rajendra, C., *A genetic algorithm for solving the fixed-charge transportation model: two-stage problem*, Computers & Operations Resaerch, **39** (2012), 2016–2032

[2] Back, T., Hoffmeister, F. and Schwefel, H.-P., A survey of evolution strategies, in *Proc. of the 4th International Conference on Genetic Algorithms*, San Diego, CA, July, 1991

[3] Bookbinder, J. H. and Sethi, S., *The Dynamic Transportation Problem: A Survey, Naval Research Logistics Quarterly*, 1980, 27(1). DOI: 10.1002/nav.3800270107

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

[5] 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 (2011)*, Expert Systems with Applications, **39** (2012), 11321–11322

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

[7] Koza, J. R., *Concept formation and decision tree induction using the genetic programming paradigm*, in Parallel Problem Solving from Nature - Proceedings of 1st Workshop, 1989, 768–774

[8] Lupşa, L., Duca, D. I., Chiorean, I. and Neamiu, L., *Dynamic transport problems of cost type and time type*, Creat. Math. Inform., **17** (2008), No. 3, 452–459

[9] Matei, O. and Pop, P. C., An efficient genetic algorithm for solving the generalized traveling salesman problem, in *Proc. of the IEEE International Conference on Intelligent Computer Communication and Processing*, 2010, 87–92

[10] Michalewicz, Z., *Genetic algorithms + data structures = evolution programs*, Springer Science & Business Media, 2013

[11] 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

[12] Pintea, C.-M., Pop Sitar, C., Hajdu-Macelaru, 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

[13] 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

[14] Pop, P. C., *Generalized Network Design Problems. Modeling and Optimization*, De Gruyter Series in Discrete Mathematics and Applications, Germany, 2012

[15] Pop, P. C., Pintea, C. M., Pop Sitar, C. and Dumitrescu, D., *A Bio-Inspired Approach for a Dynamic Railway Problem*, IEEE Proceedings of the 9th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 449–453, IEEE Computer Society Press, Timisoara, Romania, September 26–29, 2007

[16] Pop, P. C., Pintea, C. M., Pop Sitar, C. and Hajdu-Macelaru, M., *An efficient reverse distribution system for solving a supply chain network design problem*, J. Appl. Logic, Elsevier, Vol. **13** (2015), No. 2, Part A, 105–113

[17] Schaffer, J. R. and O'leary, D. E., *Use of penalties in a branch and bound procedure for the fixed transportation problem*, European Journal of Operational Research, **43** (1989), 305–312

[18] Sun, M., Aronson, J. E., Mckeown, P. G. and Drinka, D. A., *A tabu search heuristic procedure for the fixed charge transportation problem*, European Journal of Operational Research, **106** (1998), 441–456

[19] Syarif, A., Yun, Y. and Gen, M., *Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm*, Computers and Industrial Engineering, **43** (2002), 299–314

[20] Wirth, N., Algorithms and data structures, 1986

[1] DEPARTMENT OF MATHEMATICS AND INFORMATICS
TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
NORTH UNIVERSITY CENTER AT BAIA MARE
BAIA MARE, ROMANIA
*E-mail address*: petrica.pop@cunbm.utcluj.ro
*E-mail address*: ioanazelina@yahoo.com

[2] DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
NORTH UNIVERSITY CENTER AT BAIA MARE
BAIA MARE, ROMANIA
*E-mail address*: oliviu.matei@holisun.com

[3] DEPARTMENT OF ECONOMICS
TECHNICAL UNIVERSITY OF CLUJ-NAPOCA
NORTH UNIVERSITY CENTER AT BAIA MARE
BAIA MARE, ROMANIA
*E-mail address*: corina.pop.sitar@cunbm.utcluj.ro