International Journal of Engineering
Journal Homepage: www.ije.ir

# Addressing the Freight Consolidation and Containerization Problem by Recent and Hybridized Meta-heuristic Algorithms

E. Nasiri[a], A. J. Afshari[a], M. Hajiaghaei-Keshteli*[b]

[a] *Department of Industrial Engineering, Shomal University, Amol, Iran*
[b] *Department of Industrial Engineering and Management System, University of Science and Technology of Mazandaran, Behshahr, Iran*

*PAPER INFO*

*Paper history:*
Received 11 January 2017
Received in revised form 02 February 2017
Accepted 09 February 2017

*Keywords*:
Freight Consolidation and Containerization Problem
Red Deer Algorithm
Stochastic Fractal Search
Hybridized Methods
Intensification and Diversification Phases

*ABSTRACT*

Nowadays, in global free market, third-party logistics providers (3PLs) are becoming increasingly important. Hence, this study aims to develop the freight consolidation and containerization problem, which consists of loading items into containers and then shipping these containers to different warehouse they are delivered to their final destinations. In order to handle the proposed problem, this research not only uses the traditional and recent algorithms, but also the two new hybridized methods are introduced in order to strengthen the advantages of recent ones. In this regard, this study considers the two important phases in meta-heuristic to develop new ones. Besides, Taguchi experimental design method is utilized to set and estimate the proper values of the algorithms' parameters to improve their performance. For the purpose of performance evaluation of the proposed algorithms, various problem sizes are employed and the computational results of the algorithms are compared with each other. Finally, the impacts of the rise in the problem size on the performance of the proposed algorithms are investigated.

***doi**: 10.5829/idosi.ije.2017.30.03c.10*

## 1. INTRODUCTION

The management of containers is an important problem in global logistics networks. The freight consolidation and containerization problem (FCCP) deals with the loading of items into containers that are shipped to different possible locations, from where they are sent to their final destinations [1]. It is assumed that third-party logistics providers will take care of the transportation. This problem was introduced in the literature [2] in the context of the transportation of textile products for children.

Some important aspects to be considered in global logistics networks are the loading followed by the transportation of the containers [3]. The complexity of the loading varies depending on the type of items transported. A simple type of containerization corresponds to the NP-*hard* bin packing problem [4], which consists of packing one-dimensional items into the minimum amount of bins. The more general multi-capacity one-dimensional bin packing is a special case of the FCCP, in which there is only one location to which the bins can be shipped.

The transportation of containers or goods can be categorized as long or short transportation. Long transportation modes include maritime and rail transport; a review on maritime routing can be found in the literature [5]. Short transportation usually occurs after (or before) a long transportation between the port or rail station and a warehouse, usually giving rise to pickup and delivery problems [6]. A short transportation was studied in some works [7]. In some situations, as in the context of the problem we are considering in this paper, a long transportation of containers (*i.e.* from one country to another), the goods still have to be delivered to their final destination using a short transportation mode after the containers are unloaded.

The symmetric characteristic of certain problems (with bin packing among them), has challenged the performance of optimization approaches [8]. Some

*Corresponding Author's Email: mostafahaji@mazust.ac.ir (M. Hajiaghaei-Keshteli)

Please cite this article as: E. Nasiri, A. J. Afshari, M. Hajiaghaei-Keshteli, Addressing the Freight Consolidation and Containerization Problem by Recent and Hybridized Meta-heuristic Algorithms, International Journal of Engineering (IJE), TRANSACTIONS C: Aspects Vol. 30, No. 3, (March 2017) 403-410

authors have coped with symmetry by providing problem specific alternatives [9] for the vertex coloring problem [10], for the bin packing and cutting stock problems [11], for the partition coloring problem and finally, [12] for the job grouping problem. For the interested reader, a survey on symmetry in integer programming was presented [2] that studied the FCCP and proposed a standard formulation by two evolutionary meta-heuristics, GA and MA used for handling the problem. But, this paper develops the FCCP by various sizes for the containers in large scale network for the first time. Besides, to handle the proposed model, Red Deer algorithm and Stochastic Fractal Search are firstly used in an engineering problem as well. In addition, since most of the recent works usually used hybridized methods [13] in meta-heuristic literature, by considering the advantages of the recent algorithms and the two important phases in meta-heuristic, the two new hybridized methods are developed.

This manuscript is organized as follows. In Section 2 we formally describe the problem in MIP formulations. In Section 3, our solution approaches are presented. Computational results are investigated in Section 4. Finally, the results and suggestions for future works are presented in Section 5.

## 2. THE PROBLEM FORMULATION AND DESCRIPTIONS

Our proposed FCCP can be modeled in the following manner. Let $S$ and $R$ be the sets of shipments and shipping routes, respectively. Each shipment $k \in S$ may consist of one or multiple items, and set $I$ comprises all items, each having a size $v_i$. The shipment containing item $i \in I$ is denoted by $S$ ($i$). The collection of all containers is denoted by set $B$, and the attributes of each container $j \in B$ are shipping route $R$ ($j$), price $p_j$ and capacity $V_j$. We assume different sizes for containers and each shipping route may have different price, too. The parcel delivery cost associated with assigning item $i$ to container $j$ is given by $r_{ij}$. Note that once an item is assigned to a shipping route, its parcel delivery cost is determined since this cost is only related to the locations of the destination hub and its store. In particular, assigning an item to any container in a given shipping route would incur the same parcel delivery cost $r_{ij}$.

We define three types of binary decision variables in proposed model. Let $x_{ij}$ be a binary decision variable that is equal to 1 if item $i$ is loaded into container $j$, and 0 otherwise; let $z_{kl}$ be equal to 1 if shipment $k$ is assigned to route $l$, and 0 otherwise; and let $y_j$ be equal to 1 if container $j$ is used, and 0 otherwise. The following is an IP model for the FCCP:

$$\text{obj} = \min \sum_{j\in B} y_j p_j + \sum_{i\in I}\sum_{j\in B} r_{i,j} x_{i,j} \tag{1}$$

$$\text{s.t. } \sum_{j\in B} X_{i,j} = 1, \text{for all } i \in I \tag{2}$$

$$X_{i,j} \le y_j, \text{for all } i \in I \text{ and } j \in B \tag{3}$$

$$X_{i,j} \le z_{k,l}, \text{for all } i \in I : S_{(i)} = k \text{ and } j \in B: R_{(j)} = l \tag{4}$$

$$\sum_{l\in R} z_{k,l} = 1, \text{for all } k \in S \tag{5}$$

$$\sum_{i\in I} v_i X_{i,j} \le V_j, \text{for all } j \in B \tag{6}$$

$$X_{i,j}, y_j \in \{0,1\}, \text{for all } i \in I \text{ and } j \in B \tag{7}$$

$$z_{k,l} \in \{0,1\}, \text{for all } k \in S \text{ and } l \in R \tag{8}$$

The objective (1) is to minimize the total transportation cost. Constraint (2) guarantee that each item is loaded into only one container. The container into which any item is loaded must be marked as “used”, which is realized by Constraint (3). Constraint (4) states that if item $i$ is loaded into container $j$, the shipment that includes item $i$ must be assigned to the shipping route that provides container $j$. Constraint (5) ensures that each shipment must be assigned to only one shipping route. Constraint (6) requires that the capacity limitations of the containers are not violated.

## 3. SOLUTION APROACH

In this paper, researchers have some good plans to solve the proposed model. First of all, we presented two powerful recent meta-heuristics. These methods are used repeatedly in the recent papers to solve the NP-*hard* problems. As mentioned earlier, most of papers in FCCP used two successful methods (*i.e*. GA and SA). So, researchers also utilize these two well-known methods to compare with the newly proposed methods. In addition, two new hybridization methods are developed from both groups in order to enrich the algorithms and use their advantages. In the following subsections, the proposed methods are detailed to address the problem.

**3. 1. Encoding Scheme** In order to handle the algorithms, we need to define an array to code the algorithms. Hence, in our approach, the information coded into a chromosome simultaneously specifies the assignment of shipments to shipping routes and the loading sequence of items into containers. A gene consists of a 2-tuple and the number of genes in a chromosome is equal to the number of items, i.e., $n = |I|$. Formally, we express a chromosome as a vector $x = \big((\tau_1, \sigma_1), \dots, (\tau_i, \sigma_i), \dots, (\tau_n, \sigma_n)\big)$, where $\tau_i$ and $\sigma_i$ represent the shipping route and loading order of item $i$, respectively. The genes in a chromosome are grouped into $|S|$ blocks, each corresponding to a shipment. Since our chromosome encoding does not contain information on how the items are loaded into containers, to transform a chromosome into a feasible solution of the FCCP requires the solutions of several one-dimensional VSBPPs with a pre-determined item loading sequence; the VSBPP requires that a set of items is loaded into a number of containers with different volumes and costs such that the total cost of containers used is minimized.

An instance chromosome is shown in Figure 1. This chromosome involves shipments 1, 2 and 3, which contain item sets {1; 2}, {3; 4; 5; 6} and {7; 8}, respectively. Items 1–6 and 7–8 are assigned to shipping routes 1 and 2, respectively, and the loading sequence of items 1–8 is (3, 8, 2, 4, 7, 6, 5, 1). Note that the item loading sequence is relative; for instance, on shipping route 1 we sequentially load items 3, 1, 4, 6, 5, 2 into containers since their loading orders have the relationship 2 < 3 < 4 < 6 < 7 < 8. As the items of each shipment must be transported along a unique shipping route, the first elements of all 2-tuples in the same block are identical. To obtain a feasible solution from this chromosome, we need to solve two VSBPPs with item sequences (3, 1, 4, 6, 5, 2) and (8, 7).

| | Shipment 1 | | Shipment 2 | | | | Shipment 3 | |
|---|---|---|---|---|---|---|---|---|
| Item | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| chromosome | (1, 3) | (1, 8) | (1, 2) | (1, 4) | (1, 7) | (1, 6) | (2, 5) | (2, 1) |

**Figure 1.** The encoding scheme

**3. 2. Red Deer Algorithm (RDA)** Although many methods have been developed in the recent two decades, but just only a few of them considered and discussed the two important phases; exploration and exploitation, and their trade-off. Red Deer algorithm presented by Fathollahi Fard and Hajiaghaei-Keshteli [14] is one of first methods in recent meta-heuristics to give the opportunity to a user to make a balance between intensification and diversification. This algorithm explores the Red Deer's characteristics in breeding season and simulates their main behaviors in this specially time of year. The Scottish Red Deer (*Cervus Elaphus Scoticus*) is a subspecies of Red Deer and lives in British Isles. The males roar loudly and repeatedly during the breeding season and females prefer a high to a low roaring rate. The males want to increase their territory and the number of hinds in their harems. So, the course of fight is unavoidable. Although it is possible that a male has no territory and harem, hence, they prefer to mate with a handy hind. In a nutshell, RDA starts with an initial population, called Red Deers (RD). They are divided into two types: hinds and male RDs.

Besides, a harem is a group of female RDs, and the competition of male RDs to get the harem with more hinds via roaring and fighting, and their mating behavior is the basis of the proposed evolutionary algorithm. In continue, the steps of the algorithm are detailed in the pseudo-code as shown in Figure 2.

**3. 3. Stochastic Fractal Search (SFS)** Stochastic Fractal Search (SFS) introduced by Salimi [15] is one of population-based and stochastic optimization techniques and inspired by the natural phenomenon of fractal's growth. Two main processes occurring in the SFS are the diffusing process and the updating process. In the first process, similar to Fractal Search, each particle diffuses around its current position to satisfy intensification (exploitation) property. This process increases the chance of finding the global minima, and also prevents being trapped in the local minima. In the latter process, the algorithm simulates how a point in the group updates its position based on the position of other points in the group.

Initialize the Red Deers population.
Calculate the fitness and sort them and form the hinds ($N_{hind}$) and male RDs ($N_{male}$).
X*=the best solution.
***while*** ($t$< maximum number of iteration)
    ***for*** each male RD
        A local search near his position.
        Update the position if better than the prior ones.
    ***end for***
    Sort the males and also form the stags and the commanders .
    ***for*** each male commander
        Fight between male commander and stag.
        Update the position of male commander and stag.
    ***end for***
    Form harems.
    ***for*** each male commander
        Mate male commander with the selected hinds of his harem randomly.
        Select a harem randomly and name it $k$.
        Mate male commander with some of the selected hinds of the harem.
    ***end for***
    ***for*** each stag
        Calculate the distance between the stag and all hinds and select the nearest hind.
        Mate stag with the selected hind.
    ***end for***
    Select the next generation with roulette wheel selection.
    Update the X* if there is better solution.
    $t$=$t$+1
***end while***
return X*

**Figure 2.** The pseudo-code of RDA

Unlike the diffusing phase in FS which causes a dramatic increase in the number of participating points, we consider a static diffusion process for SFS. It means that the best generated particle from the diffusing process is the only particle that is considered, and the rest of the particles are discarded. In addition to efficient exploration of the problem space, SFS uses some random methods as updating process. In the other word, updating process in SFS leads us to diversification (exploration) properties in meta-heuristic algorithms.

**3. 4. Genetic Algorithm (GA)** Holland [16] has developed the GA to solve the huge and complex problems, for the first time. GA is inspired by genetic evolutionary. GAs are the special type of EAs and include so many methods in this classification (*i.e*. Genetic programming [17], Scatter search [18] and different evolution [19]. Chromosomes are the structure of cells in animals, plants and humans. In GA, we define an array of variables called chromosome. Chromosomes are altered by two operators: mutation and crossover. So, some new solutions are created by these two mentioned operators [20].

**3. 5. Simulated Annealing (SA)** SA introduced by Kirkpatrick et al. [21], is based on the annealing process of metals. Researchers know that SA is an intelligent single-solution method. In addition, SA is a kind of a local search algorithm. In this probabilistic algorithm, SA starts with an initial random solution. The neighbor of this solution is made by some suitable techniques.

**3. 6. Hybridized RDA & GA (RDGA)** In this section, by hybridized RDA and GA, a new meta-heuristic is developed. This algorithm obtains RDA as main loop and GA as a local search. It seems that RDA is very good at intensification phase by two different operators to perform it. In this method, roaring and fighting process are saved and instead of mating process, algorithm obtains the GA by using crossover operator. In order to code this, each commander and all hinds in his harem are mated by crossover operator. This modified RDA can reduce the process time and does the diversification phase better than the general RDA in these special steps about mating process. As illustrated in Figure 3, the pseudo-code of the proposed hybridized algorithm is presented. This idea is probed to solve the problem in comparison to its original algorithms.

**3. 7. Hybridized SFS & SA (SFSA)** As mentioned in SFS, this algorithm has two main steps to do exploitation and exploration phases. It seems that this algorithm has not any special plan to escape from local optima.

In order to improve the SFS, this method is hybridized with SA to cover the disadvantages. So, a new approach is proposed. This approach uses SA to evaluate the new generation of fractals. As detailed in Figure 4, the pseudo-code of proposed algorithm is explained.

Initialize the Red Deers population.
Calculate the fitness and sort them and form the hinds ($N_{hind}$) and male RDs ($N_{male}$).
X*=the best solution.
***While*** (t< maximum number of iteration)
  ***for*** each male RDs.
    A local search near his position.
    Update the position if better than the prior ones.
  ***end for***
  Sort the males and also form the stags and the commanders.
  ***for*** each male commanders
    Fight between male commanders and stags.
    Update the position of male commanders and stags
  ***end for***
  ***for*** each male commanders
    Select a hind with roulette wheel selection.
    Specify this commander and mentioned hind as parents.
    Perform crossover and generate two new solutions.
  ***end for***
  Select the next generation with roulette wheel selection.
  Update the X* if there is better solution.
  t=t+1
***end while***
return X*

**Figure 3.** The pseudo-code of RDGA

Initialize random solutions.
Select the best solution X.
***while*** (t<maximum time number of iteration)
  ***for*** each fractal
    Do exploration phase by searching new position for new fractals.
    Calculate the fitness of these positions.
    ***if*** New fractal better the prior
      Replace the new position
    ***else***
      Calculate tetta.
      Create a random probability by rand
      ***if*** the random probability is lower than exp(-tetta/T)
        Replace this new solution instead of prior
      ***end if***
    ***end if***
    Update X.
  ***end for***
  T=T*(1-alpha);
  t=t+1;
***end while***

**Figure 4.** The pseudo-code of SFSA

## 4. COMPUTATIONAL EXPERMENTS

**4. 1. Instances** In order to analyze and study the performance of algorithms in this paper, we must have a plan to generate tests problems. The problems are divided into three classes (*i.e*. small, mediate and large). In each class, four random solutions are initialized to design the tests problems. Table 1 shows the experimental design.

**4. 2. Parameter Setting** The parameters and their levels for the algorithms are shown in Table 2. Generally, the effectiveness of meta-heuristic algorithms depends on the correct choice of the parameters. So, we study the behavior of different parameters of the proposed algorithms [22]. When the number of factors significantly increases, this method does not seem to be effective. For instance, in RDA, there are 6 parameters and three levels for them. In addition, we run each algorithm for thirty times. So, the number of runs for algorithm is equal to $6 \times 3 \times 30 = 540$ times, and it is not possible to perform this work in each algorithm.

**TABLE 1.** Experimental design of test problem

| Size of problems | No. of problems | No. of shipments | No. of containers | No. of shipping routes | Total demands | Volume of shipments | | Volume of containers | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Lower limit | Upper limit | Lower limit | Upper limit |
| Small | P1 | 10 | 3 | 2 | 1000 | 10 | 30 | 10000 | 50000 |
| | P2 | 15 | 4 | 2 | 1200 | | | | |
| | P3 | 15 | 5 | 3 | 1500 | | | | |
| | P4 | 20 | 5 | 3 | 2500 | | | | |
| Medium | P5 | 40 | 10 | 5 | 2500 | | | | |
| | P6 | 50 | 10 | 6 | 3000 | | | | |
| | P7 | 55 | 12 | 6 | 4000 | | | | |
| | P8 | 60 | 15 | 8 | 5000 | | | | |
| Large | P9 | 70 | 20 | 10 | 10000 | | | | |
| | P10 | 80 | 25 | 12 | 15000 | | | | |
| | P11 | 90 | 25 | 14 | 20000 | | | | |
| | P12 | 100 | 30 | 16 | 30000 | | | | |

**TABLE 2.** Parameters and their levels for algorithms

| | nPop | MaxT | Sub-it | reduc T | init T | $P_C$ | $P_M$ | nMale | alpha | betta | gamma | walk | nDiff |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GA | 100 | 5 | | | | 0.5 | 0.02 | | | | | | |
| | 150 | 10 | | | | 0.6 | 0.05 | | | | | | |
| | 200 | 15 | | | | 0.7 | 0.1 | | | | | | |
| SA | | 5 | 20 | 0.9 | 200 | | | | | | | | |
| | | 10 | 30 | 0.99 | 300 | | | | | | | | |
| | | 15 | 50 | 0.999 | 500 | | | | | | | | |
| RDA | 100 | 5 | | | | | | 7 | 0.7 | 0.4 | 0.6 | | |
| | 150 | 10 | | | | | | 10 | 0.8 | 0.5 | 0.7 | | |
| | 200 | 15 | | | | | | 15 | 0.9 | 0.6 | 0.8 | | |
| SFS | 100 | 5 | | | | | | | | | | 0.3 | 2 |
| | 150 | 10 | | | | | | | | | | 0.5 | 5 |
| | 200 | 15 | | | | | | | | | | 0.7 | 10 |
| RDGA | 100 | 5 | | | | | | 7 | | | 0.6 | | |
| | 150 | 10 | | | | | | 10 | | | 0.7 | | |
| | 200 | 15 | | | | | | 15 | | | 0.8 | | |
| SFSA | 100 | 5 | | 0.9 | 200 | | | | | | | 0.3 | 2 |
| | 150 | 10 | | 0.99 | 300 | | | | | | | 0.5 | 5 |
| | 200 | 15 | | 0.999 | 500 | | | | | | | 0.7 | 10 |

Due to unpractical and often impossible omission of the noise factors, the Taguchi tends to both minimize the impact of noise and also find the best level of the influential controllable factors on the basis of robustness [23]. Moreover, Taguchi determines the relative importance of each factor with respect to its main impacts on the performance of the algorithm.

For GA, SA, SFS and RDGA, we have only 4 parameters to tune and the modified orthogonal array *L9* is used. Also the modified orthogonal array *L27* is used for tuning RDA and SFSA.

**4. 3. Experimental Results** In this section, a comprehensive analysis is done on algorithms. It should be noted that each algorithm is run for thirty times. Hence, the results are based on the best value among thirty runs. We also obtain an exact approach by LINGO software to compare the outputs of algorithms. This method is checked for satisfying outputs of metaheuristics. Table 3 shows the results of the experiments.

In addition, we use Gap to show the performance of the proposed algorithms. Gap explains the deviation of solutions from the best solution. In order to achieve this purpose, Table 4 is provided. Also, to illustrate this fact clearly Figure 5 shows the Gap behavior of proposed meta-heuristics.

**TABLE 3.** The final outputs for the methods (G=Global optimum, L=Local optimum)

| $P_i$ | SA | GA | RDA | SFS | RDGA | SFSA | LINGO | |
|---|---|---|---|---|---|---|---|---|
| P1 | 10640 | 11590 | 10870 | 11340 | 11250 | 11730 | 8710 | G |
| P2 | 12490 | 11780 | 11840 | 12490 | 12630 | 11920 | 10430 | G |
| P3 | 15420 | 15280 | 15130 | 15360 | 14590 | 14580 | 15320 | G |
| P4 | 18970 | 17460 | 17240 | 18470 | 18210 | 18190 | 16940 | G |
| P5 | 23560 | 22840 | 22580 | 22690 | 21870 | 22960 | 22370 | L |
| P6 | 27510 | 28690 | 26540 | 27160 | 27390 | 27430 | 26540 | G |
| P7 | 33260 | 32710 | 33720 | 33450 | 33180 | 34710 | 32460 | L |
| P8 | 41960 | 42560 | 40830 | 41590 | 42510 | 41620 | 41730 | L |
| P9 | 57690 | 56480 | 55420 | 54620 | 54170 | 55640 | 55490 | L |
| P10 | 63840 | 66120 | 64320 | 65640 | 64570 | 65910 | 63410 | L |
| P11 | 74580 | 76870 | 73620 | 72660 | 74320 | 77420 | 72295 | L |
| P12 | 85490 | 86740 | 84630 | 85910 | 84370 | 85490 | 84370 | L |

**TABLE 4.** The Gap value for each approach

| $P_i$ | SA | GA | RDA | SFS | RDGA | SFSA | LINGO |
|---|---|---|---|---|---|---|---|
| P1 | 0.221584 | 0.330654 | 0.247991 | 0.301952 | 0.291619 | 0.346728 | 0 |
| P2 | 0.197507 | 0.129434 | 0.135187 | 0.197507 | 0.21093 | 0.142857 | 0 |
| P3 | 0.057613 | 0.048011 | 0.037723 | 0.053498 | 0.000686 | 0 | 0.050754 |
| P4 | 0.119835 | 0.030697 | 0.01771 | 0.090319 | 0.07497 | 0.07379 | 0 |
| P5 | 0.077275 | 0.044353 | 0.032465 | 0.037494 | 0 | 0.04984 | 0.022862 |

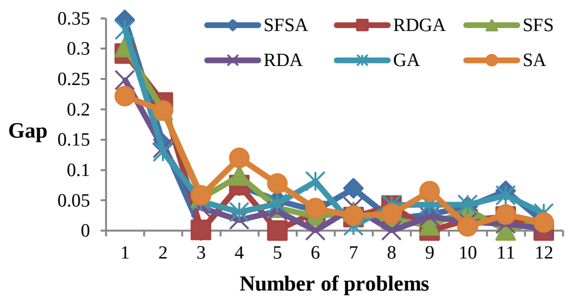

**Figure 5.** Gap behavior of algorithms

As obviously seen in the figure, the proposed RDA and its hybridized meta-heuristic show the best performance among all algorithms in this study.

In order to verify the statistical validity of the results, we have performed an analysis of variance (ANOVA) to accurately analyze the results. The results demonstrate that there is a clear statistically significant difference between performances of the algorithms. The means plot and LSD intervals (at the 95% confidence level) for six algorithms are shown in Figure 6.

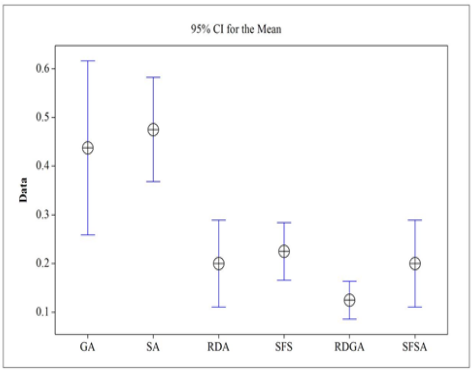


**Figure 6.** Means plot and LSD intervals for the algorithms

## 5. CONCLUSION AND FUTURE WORKS

This work studied the freight consolidation and containerization problem as an NP-*hard* problem. In order to solve the problem, the two recent meta-heuristics are used firstly in an engineering problem. Besides, the two traditional algorithms are obtained to compare the novel ones. In addition, this paper presents the two hybridized meta-heuristics based on proposed approaches for the first time as well. All of the parameters of the algorithms are tuned by Taguchi experimental design method. Finally, the performance and efficiency of algorithms are investigated and some important analyses are created to show the fact. The results explain that RDA and its proposed hybridized method have the best performance among all algorithms.

For future studies, to explore the algorithms exactly, more comprehensive analysis may be needed. In addition, some other real constraints can be proposed to develop the problems. Moreover, some real study cases can be tested on our model. At the end, more real scale optimization problems can be obtained to evaluate the two new hybridized algorithms.

## 6. REFERENCES


1. van Heeswijk, W., Mes, M., Schutten, J. and Zijm, W., "Freight consolidation in intermodal networks with reloads", ***Flexible Services and Manufacturing Journal***, (2016), 1-34.
2. Qin, H., Zhang, Z., Qi, Z. and Lim, A., "The freight consolidation and containerization problem", ***European Journal of Operational Research***, Vol. 234, No. 1, (2014), 37-48.
3. Jamrus, T. and Chien, C.-F., "Extended priority-based hybrid genetic algorithm for the less-than-container loading problem", ***Computers & Industrial Engineering***, Vol. 96, (2016), 227-236.
4. Garey, M.R. and Johnson, D.S., "Computers and intractability, wh freeman New York, Vol. 29, (2002).
5. Christiansen, M., Fagerholt, K. and Ronen, D., "Ship routing and scheduling: Status and perspectives", ***Transportation science***, Vol. 38, No. 1, (2004), 1-18.
6. Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I. and Laporte, G., "Static pickup and delivery problems: A classification scheme and survey", ***Top***, Vol. 15, No. 1, (2007), 1-31.
7. Escudero, A., MuñUzuri, J., Guadix, J. and Arango, C., "Dynamic approach to solve the daily drayage problem with transit time uncertainty", ***Computers in Industry***, Vol. 64, No. 2, (2013), 165-175.
8. Odchimar, A. and Hanaoka, S., "Intermodal freight network incorporating hub-and-spoke and direct calls for the archipelagic philippines", ***Maritime Economics & Logistics***, (2016).
9. Campelo, M., Campos, V.A. and Corrêa, R.C., "On the asymmetric representatives formulation for the vertex coloring problem", ***Discrete Applied Mathematics***, Vol. 156, No. 7, (2008), 1097-1111.
10. De Carvalho, J.V., "Lp models for bin packing and cutting stock problems", ***European Journal of Operational Research***, Vol. 141, No. 2, (2002), 253-273.
11. Frota, Y., Maculan, N., Noronha, T.F. and Ribeiro, C.C., "A branch-and-cut algorithm for partition coloring", ***Networks***, Vol. 55, No. 3, (2010), 194-204.
12. Jans, R. and Desrosiers, J., "Efficient symmetry breaking formulations for the job grouping problem", ***Computers & Operations Research***, Vol. 40, No. 4, (2013), 1132-1142.
13. Hasani, A., Soltani, R. and Eskandarpour, M., "A hybrid meta-heuristic for the dynamic layout problem with transportation system design", ***International Journal of Engineering-Transactions B: Applications***, Vol. 28, No. 8, (2015), 1175.
14. Fard, A.M.F. and Hajiaghaei-Keshteli, M., "Red deer algorithm (RDA); a new optimization algorithm inspired by red deer's mating", in International Conference on Industrial Engineering, IEEE., (2016 e), 33-34.
15. Salimi, H., "Stochastic fractal search: A powerful metaheuristic algorithm", ***Knowledge-Based Systems***, Vol. 75, No., (2015), 1-18.
16. Goldberg, D.E. and Holland, J.H., "Genetic algorithms and machine learning", ***Machine Learning***, Vol. 3, No. 2, (1988), 95-99.
17. Alemdag, S., Gurocak, Z., Cevik, A., Cabalar, A. and Gokceoglu, C., "Modeling deformation modulus of a stratified sedimentary rock mass using neural network, fuzzy inference and genetic programming", ***Engineering Geology***, Vol. 203, (2016), 70-82.
18. Glover, F., "Heuristics for integer programming using surrogate constraints", ***Decision Sciences***, Vol. 8, No. 1, (1977), 156-166.
19. Storn, R. and Price, K., "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces", ***Journal of Global Optimization***, Vol. 11, No. 4, (1997), 341-359.
20. Hajiaghaei-Keshteli, M., "The allocation of customers to potential distribution centers in supply chain networks: Ga and aia approaches", ***Applied Soft Computing***, Vol. 11, No. 2, (2011), 2069-2078.
21. Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., "Optimization by simulated annealing", ***Science***, Vol. 220, No. 4598, (1983), 671-680.
22. Engin, O., Ceran, G. and Yilmaz, M.K., "An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems", ***Applied Soft Computing***, Vol. 11, No. 3, (2011), 3056-3065.

23. Hajiaghaei-Keshteli, M. and Aminnayeri, M., "Solving the integrated scheduling of production and rail transportation problem by keshtel algorithm", ***Applied Soft Computing***, Vol. 25, (2014), 184-203.

# Addressing the Freight Consolidation and Containerization Problem by Recent and Hybridized Meta-heuristic Algorithms

E. Nasiri[a], A. J. Afshari[a], M. Hajiaghaei-Keshteli[b]

[a] *Department of Industrial Engineering, Shomal University, Amol, Iran*
[b] *Department of Industrial Engineering and Management System, University of Science and Technology of Mazandaran, Behshahr, Iran*

*P A P E R I N F O*

*Paper history:*
Received 11 January 2017
Received in revised form 02 February 2017
Accepted 09 February 2017

*Keywords*:
Freight Consolidation and Containerization Problem
Red Deer Algorithm
Stochastic Fractal Search
Hybridized Methods
Intensification and Diversification Phases

**چکیده**

امروزه در بازارهای ازاد جهانی، اهمیت شبکه ی تهیه کنندگان سه بخشی بیش از بیش مورد توجه است. از این رو، این مطالعه در نظر دارد تا به بررسی مساله ی حمل و نقل کانتینری که شامل بسته بندی مجموعه ای از کانتینرها و نقل این کانتینرها به سمت تامین کنندگان مختلف که آنها را به سمت مقصد نهایی می رسانند. به منظور حل مساله پیشنهادی، این تحقیق نه تنها از الگوریتم های جدید و سنتی بهره بدره است بلکه به معرفی دو روش جدید ترکیبی به منظور تقویت مزایای الگوریتم های اخیر دارد، پرداخته است. در این رابطه، دو فاز مهم از فراابتکاری ها را برای الگوریتم های جدید مورد بررسی قرار داده است. علاوه بر این، از روش طراحی ازمایشات تاگوچی برای تخمین پارامترهای مناسب برای الگوریتم‌ها به منظور افزایش کارایی استفاده شده است. به منظور یک ارزیابی مناسب الگوریتم ها مسائل مختلفی که نشان دهنده ی نتایج محاسباتی هستند برای الگوریتم ها با یکدیگر مقایسه شده اند. و در پایان، تاثیر رشد مسائل در کارایی الگوریتم ها مورد بحث واقع شده اند.

***doi****: 10.5829/idosi.ije.2017.30.03c.10*