Journal of Optimization in Industrial Engineering 1(2018) 51- 65
DOI: 10.22094/JOIE.2017.594.1382

Location-Allocation and Scheduling of Inbound and Outbound Trucks in Multiple Cross-Dockings Considering Breakdown Trucks

Javad Behnamian[a], Seyed Mohammad Taghi Fatemi Ghomi[b,*], Fariborz Jolai[c], Pooya Heidary[d]

[a] *Assistant Professor, Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran*
[b] *Professor, Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran.*
[c] *Professor, Department of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran*
[d] *MSc, Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran*.

Received 30 June 2017; Revised 16 October 2017; Accepted 2 November 2017

Abstract

This paper studies multiple cross-dockings where the loads are transferred from origins (suppliers) to destinations (customers) through cross-docking facilities. Products are no longer stored in intermediate depots and incoming shipments are consolidated based on customer demands and immediately delivered to them to their destinations. In this paper, each cross-docking has a covering radius that customers can be served by at least one cross-docking provided. In addition, this paper considers the breakdown of trucks. We present a two-stage model for the location of cross-docking centers and scheduling inbound and outbound trucks in multiple cross-dockings. We work on minimizing the transportation cost in a network by loading trucks in the supplier locations and route them to the customers via cross-docking facilities. The objective, in the first stage, is to minimize transportation cost of delivering products from suppliers to open cross-docks and cross-docks to the customers; in the second-stage, the objective is to minimize the make spans of open cross-dockings and the total weighted summation of completion time. Due to the difficulty of obtaining the optimum solution to medium- and large-scale problems, we propose four types of metaheuristic algorithms, i.e., genetic, simulated annealing, differential evolution, and hybrid algorithms. The result showed that simulated annealing is the best algorithm between the four algorithms.

Keywords: Cross-docking, Transhipment, Location of cross-docking centers, Metaheuristic.

1. Introduction

The operation of a distribution center consists of five basic functions: receiving, sorting, storing, retrieving, and shipping. The best way to reduce cost and improve efficiency is not by simply improving a function, but by eliminating it if feasible (Yu and Egbelu, 2008).In a traditional distribution center, goods are first received, and then stored. When a customer requests an item, workers pick it from the storage and ship it to the destination. From these four major functions of warehousing, storage and order picking are usually the most costly ones. Storage is expensive due to the inventory holding costs; the same is true for order picking because it is labor-intensive. One approach to reduce costs could be to improve one or more of these functions or to improve how they interact. Cross-docking is a material handling and distribution concept in which items move directly from receiving dock to shipping dock without being stored in a warehouse or distribution center which has the potential of eliminating storage and recovery and the two most expensive warehousing operations. One innovative warehousing strategy that has great potential for controlling the logistics and distribution costs while simultaneously enhancing the level of customer service is cross-docking (Apte and Viswana than, 2000).As the timing of delivery and pickup is becoming increasingly crucial in a supply chain, the use of "cross-docking" has become synonymous with rapid consolidation and processing (Chen et al., 2009). Different from a warehouse, a cross-docking is a transshipment center rather than an inventory warehouse. Cargos can be delivered, consolidated, and picked up in across-docking; also, delayed transshipments are allowed, but the delay allowance is usually no more than 24 hours and even less than 1 hour in some cases (Li et al., 2004).

The location and scheduling problems with cross-docking identified new areas of research, which take account of two main components of cross-docking distribution networks, namely cross-docking centers location and scheduling of

*Corresponding author Email address: fatemi@aut.ac.ir

trucks. The location and scheduling problem involves the strategic (*i.e*., location) and tactical/operational (*i.e*., scheduling) decision levels in supply chain management.

This paper presents a two-stage mixed-integer programming (MIP) model for the location of cross-docking centers and scheduling trucks for the cross-docking distribution networks in the supply chain. In the strategic stage, we define the location of possible cross-docks and allocation suppliers to open cross-docks and cross-docks to customers considering coverage radius. In the tactical stage, we consider a bi-objective problem in which the first objective is to minimize the total weighted summation of completion time, and the second objective is to minimize the make spans of open cross-dockings. Due to the difficulty of obtaining the optimum solution to medium- and large-scale problems, we propose four types of metaheuristic algorithms.

2. Literature Review

Liao et al. (2012) proposed to minimize total operation time in inbound and outbound truck sequencing in cross-docking problem; they proposed two hybrid differential evolution algorithms. They also suggested an operational policy that leads to shorter make span than suggested by older studies. To minimize the total weighted tardiness, Liao et al. (2013) studied the simultaneous dock assignment and sequencing of inbound trucks for a multi-door cross-docking under a fixed outbound truck departure schedule. They proposed a new model and solved it by six different metaheuristic algorithms. In this study, for evaluating the total weighted tardiness associated with any given inbound-truck sequence and dock assignment, an operational policy was developed. Kuo (2013) considered multi-door cross-docking problem with both inbound and outbound truck sequencing and both inbound and outbound truck dock assignment and make span objective function. He/she proposed a model integrated with a variable neighborhood search (VNS) to optimize the sequence of all inbound and outbound trucks. Four simulated annealing algorithms were also adopted for comparison. Mohtashami (2015) proposed a genetic algorithm-based framework for scheduling inbound and outbound trucks in cross-docking systems with temporary storage of product items at shipping dock for the second defined scenario, such that it minimizes the total operation time.

Assadi and Bagheri (2016) considered just-in-time philosophy in truck scheduling problem in which interchangeability of products, ready times for both inbound and outbound trucks, and different transshipment time between receiving and shipping doors were considered. To minimize the total earliness and tardiness for outbound trucks, a mixed integer programming model and two metaheuristic algorithms were developed. Amini and Tavakkoli-Moghaddam(2016) used bi-objective linear mathematical model for truck scheduling problem in a cross-docking center, with breakdowns during their service times which have Poisson distribution function. They also employed non-dominated sorting genetic algorithm II, multi-objective simulated annealing, and multi-objective differential evolutionary algorithms to solve the problem. Rahmanzadeh Tootkaleh et al. (2016) proposed an inbound truck scheduling model based on fixed outbound trucks' departure times. This study assumed that delayed loads are stored in temporary storage until the next outbound trucks' departure time with the same destination. To solve the model, they proposed a heuristic algorithm.

3. Mathematical Modeling

This paper makes the assumptions as those of Yu (2002) in the following:

1. All inbound and outbound trucks are available at time zero.
2. All products received must be shipped out; long-term storage is not allowed.
3. The total number of units with a given product type must equal the number of units shipped of the same product type.
4. The unloading sequence of the products from an inbound truck can be determined.
5. It is permissible to unload only the required number of units of a particular product from an inbound truck.
6. Only one unit count of a product can be loaded into the outbound truck at a time. Therefore, loading multiple counts simultaneously from a conveyor and the temporary storage into an outbound truck are prohibited.
7. Truck changeover time is the same for all inbound and outbound trucks.
8. There is one receiving dock and one shipping dock, and the docks are separate.
9. The capacity of the temporary storage buffer is infinite.
10. The following information is assumed to be known as a priori:
    (*i*) Product types and the quantity of each product loaded in an inbound truck;
    (*ii*) Product types and the quantity of each product needed for an outbound truck;
    (*iii*) Loading and unloading times for the products;
    (*iv*) Moving time of products from the receiving dock to the shipping dock;
    (*v*) Truck changeover time.

In this paper, make span is defined as the total operating time of the cross-docking operation. The total operating time starts from the moment when the first product of the first scheduled inbound truck is unloaded onto the receiving dock to the moment when the last product of the last scheduled outbound truck is loaded from the shipping dock. The objective is to find the best truck docking sequences for both of the inbound and the outbound trucks to minimize the total cross-docking operation time (equivalent to maximizing the throughput rate) of the cross-docking process. The product assignments from inbound trucks to

outbound trucks as well as docking sequences of the inbound and outbound trucks are also determined, simultaneously (Yuand Egbelu, 2008).

I: Set of suppliers
P: Set of possible cross-dockings
K: Set of customer zones
L: Set of product families
D: Truck changeover time
V: Moving time of products for all trucks from the receiving dock to the shipping dock
C: Capacity of each truck
M: Big number
C_{ip}: Cost to transport product from supplier *i* to cross-docking*p*
C_{pk}: Cost to transport product from cross-docking p to customer *k*
F_p:Fixed operating cost to operate/open cross-docking center*p*
r_{ipl}: Number of units of product type l initially loaded in inbound truck *i* at cross- docking*p*
S_{jpl}: Number of units of product type l initially needed for outbound truck *j* atcross-docking *p*
φ^I_{ip} : Probability of breakdown inbound truck *i* in cross-dock *p*
φ^O_{jp} : Probability of breakdown outbound truck *j* in cross-dock *p*
G_{jp} :Due date of outbound truck *j* for outgoing of cross-dock *p*
B^I_{ip} :Average repair time of inbound truck *i* in cross-dock *p*
B^O_{jp} : Average repair time of outbound truck *j* in cross-dock *p*

Variables:

y_p: 1 If cross-docking *p* is open; 0 Otherwise

y_{ip}: 1 If supplier *i* is assigned to cross-docking*p*; 0 Otherwise

W_{pk}: 1 If customer *k* is assigned to cross-docking *p*; 0 Otherwise

a_{pk}: If demand point *k* can becovered by cross-docking *p* (distance between demand point *k* and candidate point *j* is less than covering radius) a_{pk} = 1, and it would be 0 otherwise

Continuous variables:

T_p: Makespan at cross-docking *p*
c_{ip}: Time at which inbound truck *i* enters the receiving dock of cross-docking *p*
F_{ip}: Time at which inbound truck *i* leaves the receiving dock of cross-docking *p*
d_{jp}:Time at which outbound truck *j* enters the shipping dock of cross-
L_{jp}: Time at which outbound truck *j* leaves the shipping dock of cross-

Integer variables:

x_{ijpl}: Number of units of product type *l* transferred from inbound truck *i* to outbound truck *j* at cross-

Binary variables:

v_{ijp}= 1 If any product is transferred from inbound truck *i* to outbound truck *j* at cross-; 0 Otherwise;

v_{ijp}= 1 If inbound truck *I* precedes inbound truck *j* in the inbound truck sequence at cross-docking*p*0 ; 0 Otherwise;

v_{ijp}= 1 If outbound truck *i* precedes outbound truck *j* in the outbound truck sequence at cross-docking *p* 0; 0 Otherwise;

Cross-docking centers' location (stage 1):

The location problem can be formulated as below:

$$Min\ Z_1 = \sum_{p=1}^{o} F_p z_p + \sum_{i=1}^{n}\sum_{p=1}^{o} C_{ip} y_{ip} + \sum_{p=1}^{o}\sum_{k=1}^{m} C_{pk} w_{pk} \quad (1)$$

subject to:

$$\sum_{p=1}^{o} y_{ip} = 1, \qquad \forall i \quad (2)$$

$$\sum_{p=1}^{o} w_{pk} \geq 1, \qquad \forall k \quad (3)$$

$$\sum_{p=1}^{o} z_p \geq 1 \quad (4)$$

$$\sum_{p=1}^{o} a_{pk} z_p \geq 1, \qquad \forall k \quad (5)$$

$$\sum_{p=1}^{o} F_p z_p \leq TC \quad (6)$$

$$y_{ip} \leq z_p, \qquad \forall i, p \quad (7)$$

$$w_{pk} \leq z_p, \qquad \forall p, k \quad (8)$$

$$y_{ip}, w_{pk}, z_p \in \{0,1\} \quad (9)$$

Objective function (1) minimizes the sum of fixed costs to open cross-docking centers and transportation costs from suppliers to cross-docking centers and from cross-docks to customers where *o* is the maximum number of cross-dock. Constraint (2) ensures that each supplier is only allocated to one of the open cross-docking centers. Constraint (3) ensures that each customer is metat least by one of the open cross-docking centers. Constraint (4) ensures that at least one cross-dock should be open. Constraint (5) presents the limitation related to the coverage radius. Constraint (6) ensures that the total cost paid for opening cross-docking centers could not be greater than a certain amount. Constraints (7) and (8) ensure that the allocation of suppliers to cross-docking center and of cross-docking center to customers can be executed only when the corresponding cross-docking center is open. Constraint (9) defines corresponding decision variables of the model.

Scheduling (stage 2):

$$MinZ_2 = \sum_{p=1}^{m} T_p \quad (10)$$

subject to:

$$T_P \geq L_{jp} \quad \forall j,p \quad (11)$$

$$r_{ipl} = \sum_{j=1}^{S_p} x_{ijpl} \qquad \forall i, p, l \quad (12)$$

$$S_{jpl} = \sum_{i=1}^{R_p} x_{ijpl} \qquad \forall j, p, l \quad (13)$$

$$x_{ijpl} \leq M v_{ijp} \qquad \forall i, j, p, l \quad (14)$$

$$F_{ip} \geq C_{ip} + \sum_{l=1}^{N} r_{ipl} + \varphi^{l}{}_{ip} B^{l}{}_{ip} \qquad \forall i, p \quad (15)$$

$$C_{jp} \geq F_{ip} + D - M(1 - P_{ijp}) \qquad \forall i, j, p, i \neq j \quad (16)$$

$$C_{ip} \geq F_{jp} + D - M(P_{ijp}) \qquad \forall i, j, p, i \neq j \qquad (17)$$

$$P_{iip}=0 \qquad \forall i,p \qquad (18)$$

$$L_{jp} \geq d_{jp} + \sum_{l=1}^{N} S_{jpl} + \varphi^{O}{}_{jp} B^{O}{}_{jp} \qquad \forall j, p \qquad (19)$$

$$d_{jp} \geq L_{ip} + D - M(1-q_{ijp}) \qquad \forall i, j, p, i \neq j \qquad (20)$$

$$d_{ip} \geq L_{jp} + D - M(q_{ijp}) \qquad \forall i, j, p, i \neq j \qquad (21)$$

$$q_{iip}=0 \qquad \forall i,p \qquad (22)$$

$$L_{jp} \geq C_{ip} + V + \sum_{l=1}^{N} x_{ijpl} - M(1-v_{ijp}) \qquad \forall i, j, p \qquad (23)$$

The objective function minimizes the sum of makespans of open cross-docks. Constraint (11) ensures that in each cross-docking, makespanis greater than or equal to the time the last scheduled outbound truck leaves the shipping dock. Constraint (12) ensures that in each cross-docking, the total number of units of product type transferred from inbound truck *i* to all outbound trucks is exactly the same as the number of units of product type initially loaded in inbound truck *i*. Similarly, Constraint (13) ensures that in each cross-docking, the total number of units of product type transferred from all inbound trucks to outbound truck *j* is exactly the same as the number of units of product type l needed for outbound truck *j*. Constraint (14) just compels the correct relationship between $x_{ijpl}$and v_{ijp} variables in each cross-docking. Constraints (15)–(17) make a right sequence in each cross-docking for arriving and departing times of the inbound trucks based on their order and with considering breakdown of inbound trucks. Constraint (18) ensures that in each cross-dock, no in bound truck can precede itself in the inbound truck sequence. Similar to Constraints (15)-(17) for inbound trucks, Constraints (19)–(21) make a right sequence in each cross-docking for arriving and departing times for the outbound trucks based on their order and with considering breakdown of outbound trucks. Similar to Constraint (18), Constraint (22) ensures that in each cross-docking, no outbound truck can precede itself in the outbound truck sequence. Constraint (23) connects the leaving time of an outbound truck to the arriving time of an inbound truck if any products or items are transferred between the trucks.

4. Metaheuristic Characteristics

The basic objective in stage 2in cross-docking problems to specify the best sequences of inbound and outbound trucks in sucha way that the make span will be minimized. Tosolve the problem in stage 2, we propose four algorithms: genetic algorithm (GA), simulated annealing (SA), differential evolution (DE), and hybrid GA-SA, respectively.

4.1. Genetic algorithm

Genetic algorithms (GAs) have become a well-known and powerful metaheuristic approach for hard combinatorial optimization problems. GAs are based on the idea of natural selection, and have been applied to numerous combinatorial optimization problems successfully. In this section, we first describe the essential components of our proposed GA for the scheduling problem, e.g., solution representation, the crossover operator, etc.

This algorithm resorts some features, such as population, chromosomes, genes, reproduction parameters, and generation in order to evolve its search procedures. Now, we describe the framework of our applied GA used by (Boloori Arabaniet al., 2011) as in the following steps:

(1) For the initial population, we create random chromosomes (sequences) separately for inbound and outbound trucks similar to Figure 1;
(2) As fitness evaluation, the makespan for each chromosome is calculated based on a theme presented by Yu(2002);
(3) For the selection operator, a chromosome is selected based on two parameters: roulette wheel selection and tournament selection;
(4) The reproduction scheme consists of three main approaches: elitism, crossover, and mutation;
(5) The probabilities of elitism, crossover and mutation (P_e, P_c, and P_m) in each of three levels are taken into account;
(6) Stopping criterion, which is the maximum number of generations.

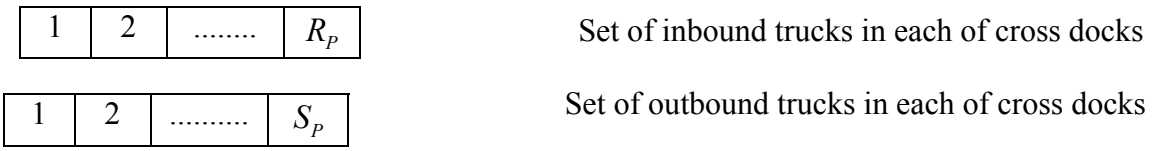

Fig. 1. Set of inbound and outbound trucks in each of cross docks

4.1.1. Crossover scheme

In the crossover phase, each chromosome is merged with another chromosome by a specific method. For this purpose, we apply 1-point and 2-point types of crossover for inbound and outbound permutation of trucks in each open cross-docking shown in Figure 2.

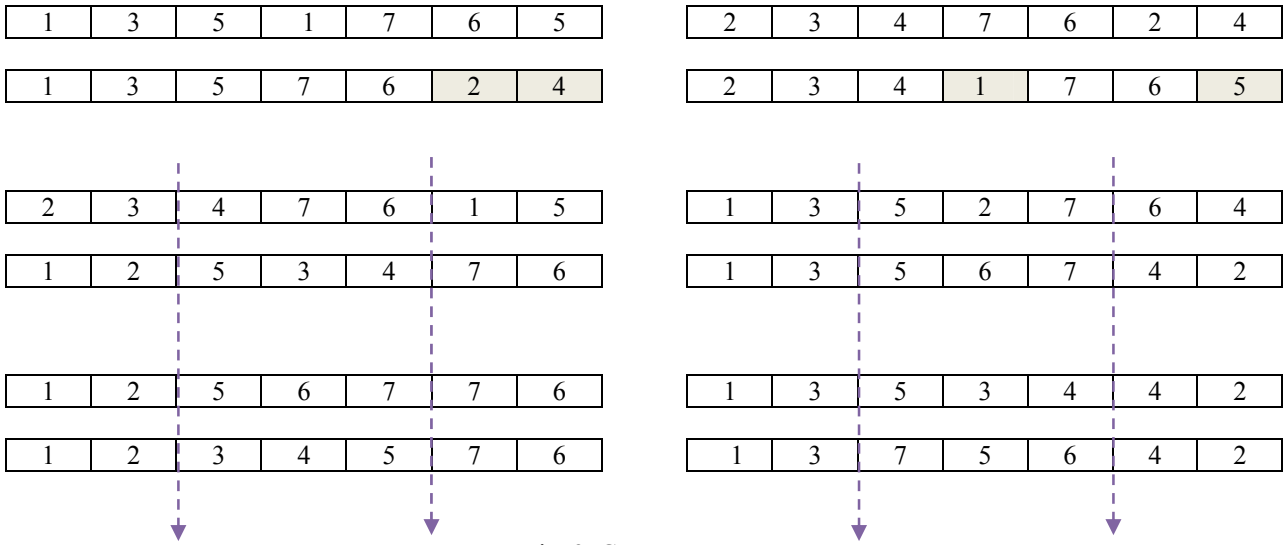

Fig. 2. Crossover operator

As is clear from the figure, we consider four schemes with 2 schemes for one-point and 2 schemes for two-point for crossover operations. In one-point crossover, at first, we select a crossover point randomly, and then all genes after crossover point from the second parent are replaced with all genes after crossover from the first parent. In order to acquire the correct sequence in duplicated genes, the place of these repeated genes should be changed in order by which they are already placed in their sequences.
In two-point crossover, first, we select two crossover points randomly, and then all genes between two points from the first parent are replaced by all genes between two points from second parent. For duplicated genes, we apply a method similar to one-point crossover.

4.1.2 Mutation scheme

As shown in Figure 3, for mutation operation, we propose three different methods, i.e., swap, reversion, and insertion. These methods function as follows:

1. Swap: two genes are selected randomly and then replaced with each other.
2. Reversion: two points from parent are chosen randomly, and then all genes between two points will be reversed.
3. Insertion: two points from parent are chosen randomly; the first gene is inserted after second gene.

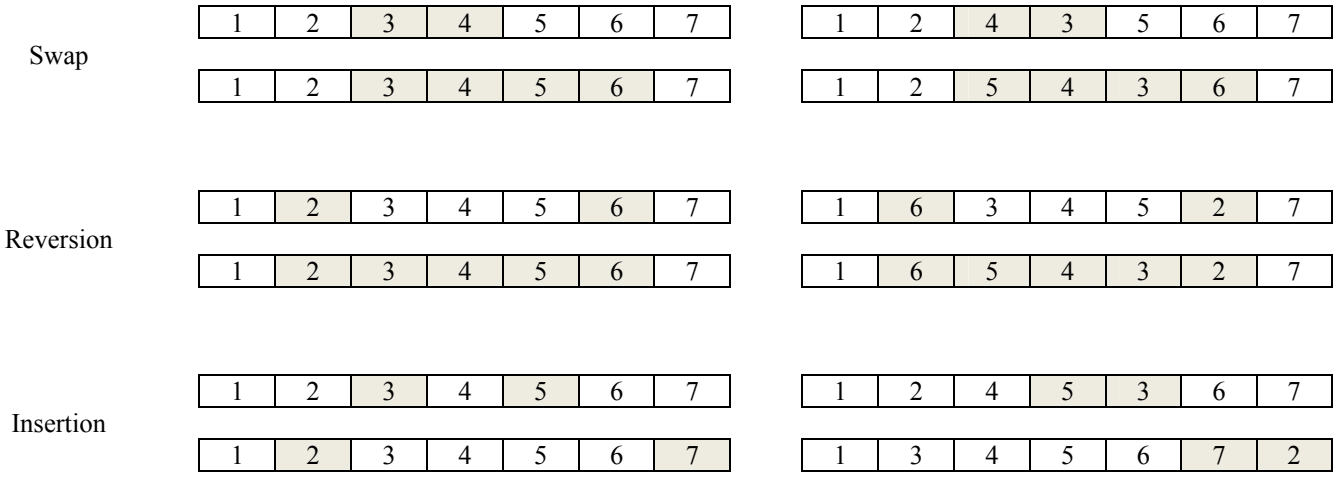

Fig. 3. Mutation operator

4.2. Simulated annealing approach

Simulated annealing is a computational process which attempts to solve hard combinatorial optimization problems through controlled randomization. This approach was popularized by Kirkpatrick et al. (1983) based on a work by Metropolis et al. (1953) (the so-called Metropolis algorithm) in statistical mechanics. Simulated annealing emulates the physical process of annealing (hence, the name of the heuristic) where the annealing process involves the following steps:

1. The temperature of the system is raised to a sufficient level.
2. The temperature of the system is maintained at this level for a prescribed amount of time.
3. The system is allowed to cool under controlled conditions until the desired energy state is attained.

The initial temperature (Step 1), the time the system remains at this temperature (Step 2) and the rate at which the system is cooled (Step 3) are referred to as the annealing schedule. The algorithm's transitions can be modeled as a collection of finite-length Markov chains corresponding to each temperature level of the system. Hence, through selection of an appropriate probability distribution and through control of its parameters, the algorithm's rate of convergence is controlled. The general procedure to implementa simulated annealing algorithm is as follows (Chen et al., 1996):

Step1. Select an initial temperature, *t*, and an initial solution, X_0 . Let $f_0 = f(X_0)$ denote the corresponding objective value. Set *i* = 0 and go to Step 2.

Step2. Set $i = i + 1$. Randomly generate a new solution, X_i, and evaluate $f_i = f(X_i)$.

Step3. If $f_i > f_{i-1}$, go to Step 5. Otherwise, accept f_i as anew solution with probability $Pr = e^{\frac{f_i - f_{i-1}}{t}}$.

Step4. If f_i was rejected as the new solution instep 3, set $f_i = f_{i-1}$. Go to Step 5.

Step5. If satisfied with the current objective value (f_i), stop. Otherwise, adjust the temperature, *t*, according to the annealing schedule and go to Step 2.

4.3. Differential evaluation

Differential evaluation (DE), as developed by Storn and Price (1995), is one of the population-based evolutionary metaheuristic. The DE algorithm generally consists of six ingredients, and because of these elements, it has been configured as a method in which the main idea is based on generating random vectors. Actually, in order to adapt this algorithm to our cross-docking problem, it should be said that each solution (each sequence of trucks) is shown by a vector. Now, we are going to discuss each of these six ingredients separately.

4.3.1. Population framework

In order to have an initial population, a set of truck sequences is generated at random. In this algorithm, it should be observed that the population is configured as a matrix and consists of vectors, called target vectors, as follows:

$$P_X(t) = (X_{ij}(t)) = \begin{pmatrix} x_{11} & \cdots & x_{1R} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mR} \end{pmatrix} \quad (24)$$
$$i = 1,2,...,n; \quad j = 1,2,....,R$$

where $P_X(t)$ is the population matrix at iteration *t*; $X_{ij}(\mathrm{t})$ represents sequence*i*; *m* is the population size or the number of sequences that must be primarily identified.

4.3.2. Initialization of the population

$$X_{ij}(t) = \left[rand[0,1) \times (\mathrm{U}_L - L_L)\right] + L_L \quad (25)$$

where U_L and L_L are the upper and lower bounds of each vector, respectively, and rand [0, 1) is a random number generated in [0, 1).By this method, $X_{ij}(\mathrm{t})$ will probably take floating-point values in [0, 1). So, Lichtblau (2002) proposed an alternative approach called the relative position indexing by which the target vectors can obtain the decimal point values.

4.3.3. Elitism scheme

$$V_{ij}(t) = X_{best-j}(t) + F\left[X_{aj}(t) - X_{bj}(t)\right]; \quad i \neq a,b \quad (26)$$

where $V_{ij}(t)$ corresponds to the mutant vectors. In these equations, $X_{best-j}(t)$ represents the best vector (having the best fitness function)among all vectors of the current population; $X_{aj}(\mathrm{t})$, $X_{bj}(\mathrm{t})$ are the vectors in iteration *t*; meanwhile, it should be noted that these vectors must be different from vector $V_{ij}(t)$ and be selected at random; *F* is a real-valued factor ranged in[0, 1).

$$D_{ij}(t) = \begin{cases} V_{ij}(t) & if\ rand[0,1) \le Cr \\ X_{ij}(t) & otherwise \end{cases} \quad (27)$$

where*Cr* is called crossover parameter created at the beginning of the algorithm. The above equation indicates that if the value of the random number, created from [0, 1),is less than or equal to Cr, mutant vectors ($V_{ij}(t)$) will constitute trial vector ($D_{ij}(t)$). Otherwise, the mutant vectors will take the value of target vectors ($X_{ij}(\mathrm{t})$).

The DE algorithm proposes a method by which the target vectors can be changed to the vectors taking the value from domain [0, 1) (Price et al., 2005):

The best vectors (with the best objective functions) will be copied to the population of the next iteration.

4.3.4. Mutation vector

Now, all the target vectors have identified their floating-point frameworks. Therefore, in order to generate new vectors, called mutant vectors, a method is applied according to the following strategy:

4.3.5. Diversify via crossover

In order to create a new vector, called trial vector, the mutant vectors and the target vectors are combined with each other by means of the following equation (Price et al., 2005):

4.3.6. Selection operation

Finally, the vectors of the new population will be created by making a comparison between the target vector and the trial vector. In other words, we differentiate these vectors by their fitness evaluations (f) as follows (Price et al., 2005):

$$P_X(t+1)=\begin{cases} D_{ij}(t) & if\ f(D_{ij}) \le f(X_{ij}) \\ X_{ij}(t) & otherwise \end{cases} \quad (28)$$

where $P_X(t+1)$ is the specific sub-population at iteration t+1. Actually, the above equation makes a comparison between the trial vectors and the target vectors by means of their fitness (objective) functions.
Finally, in order to determine the rest of the vectors in the new population, all of the foregoing six steps are followed up and repeated until the stopping criterion is satisfied. Eventually, the best vector in the final iteration will be recognized as the best vector and must be converted to its equivalent sequence based on the relative position indexing method described before.

4.4. Hybrid algorithm

The structure of the basic proposed hybrid algorithm is designed as shown in Algorithm.

Algorithm: Main body of the hybrid algorithm

Begin

Step 1. Initialization
Input data set;
Initialize parameters: Number of initial population;
Crossover rate; Mutation rate; Elitism rate; Selection rate;
Stopping criteria;
Initial population generation: Randomly generate an initial population of P_{size} chromosomes;
Objective function evaluation;

Step 2. Main loop
while stopping criteria is met **do**
Crossover;
Mutation;
Solution improvement:
{
$S^* \leftarrow$ GA solution()
for *iterations* $\leftarrow$ 1 to a maximum number of *iterations* **do**
$S \leftarrow S^*$;
Perform a local search on $N_l(S)$ to find a solution S';
if $f(S') \le f(S^*)$ **then**
$S^* \leftarrow S'$;
Else
Accept S' as new solution with probability *Pr*
end if
end for
}
Objective function evaluation;
Elitism scheme;
Selection operation;
Generate next generation;
end while
Step 3. Report results

End

5. Parameters Setting

As we know, each metaheuristic requires its own parameters to solve problems and always one of the most important questions when using the method is how to determine the appropriate level of each of these parameters.
In this paper, to tune the parameters and estimate the optimal influencing process parameters, the response surface methodology (RSM) is applied. In this method, the regression equation is used to evaluate different levels of parameters. This means that a series of different levels of algorithm parameters based on the input parameters (usually, the objective function value is used) are checked; with respect to the best-fit regression equation on different levels of the parameters, optimal values are suggested for the tuning parameters. For each parameter of the two levels, we consider: -1, when the level of the parameter is low; +1, when the level of the parameter is high.
The different levels of coding parameters are selected using the following equation.

$$X_i = \frac{r_i - \left(\frac{h+l}{2}\right)}{\left(\frac{h-l}{2}\right)} \quad (29)$$

Where h and l are high and low levels of parameters, and X_i and r_i are the coded and actual values for the parameters.

Table 1
Parameters setting

State	Problem size	Method
1	Small	GA
2	Large	GA
3	Small	SA
4	Large	SA
5	Small	DE
6	Large	DE
7	Small	SA-GA
8	Large	SA-GA

Table 2
Levels of parameters when GA is used

State	High level				Low level			
	MuR	CrR	MaxIt	nPop	MuR	CrR	MaxIt	nPop
1	0.50	0.70	250	120	0.20	0.30	100	50
2	0.50	0.70	500	250	0.20	0.30	200	100
3	0.50	0.70	250	120	0.20	0.30	100	50
4	0.50	0.70	500	250	0.20	0.30	200	100
5	0.50	0.70	250	120	0.20	0.30	100	50
6	0.50	0.70	250	120	0.20	0.30	100	50
7	0.50	0.70	250	120	0.20	0.30	100	50
8	0.50	0.70	500	250	0.20	0.30	200	100

Table 3
Levels of parameters when SA is used

State	High level		Low level	
	P	θ	P	θ
1	8	0.95	5	0.75
2	10	0.95	6	0.75
3	8	0.95	5	0.75
4	10	0.95	6	0.75

Table 4
Levels of parameters when DE is used

State	High level				Low level			
	CrR	F	MaxGen	nPop	CrR	F	MaxGen	nPop
1	0.50	0.50	120	120	0.20	0.30	50	50
2	0.60	0.60	250	250	0.25	0.30	100	100
3	0.50	0.50	120	120	0.20	0.30	50	50
4	0.60	0.60	250	250	0.25	0.30	100	100

After Design Expert software is applied for using response surface method, the optimal levels of each of the 16 states are as in Tables (5-7).

Table 5
Results of RSM method for GA algorithm

State	Actual				Coding			
	MuR	CrR	MaxIt	nPop	MuR	CrR	MaxIt	nPop
1	0.49	0.30	100	120	0.94	-0.99	-1.00	1.00
2	0.23	0.58	200	250	-0.99	0.42	-1.00	1.00
3	0.38	0.56	100	50	0.20	0.29	1.00	-1.00
4	0.31	0.70	202	161	0.17	1.00	-0.99	-0.18
5	0.41	0.70	250	98	0.40	1.00	1.00	0.36
6	0.49	0.70	247	120	0.99	1.00	0.96	1.00
7	0.50	0.30	100	120	0.99	-0.98	-1.00	1.00
8	0.49	0.61	200	250	0.94	0.58	-1.00	1.00

Table 6
Results of RSM method for SA algorithm

State	Actual		Coding	
	P	θ	P	θ
1	8	0.91	1	0.59
2	10	0.89	1	0.33
3	7.93	0.94	0.95	0.93
4	6	0.95	-1	1

Table 7
Results of RSM method for DE algorithm

State	Actual				Coding			
	CrR	F	MaxGen	nPop	CrR	F	MaxGen	nPop
1	0.27	0.50	50	120	-0.54	1	-1	1
2	0.58	0.60	100	203	0.95	1	-1	0.37
3	0.34	0.50	82	120	-0.06	1	-0.1	1
4	0.26	0.60	100	240	-0.93	1	-1	0.87

6. Computational Results

Computational tests in this section are generated to verify and evaluate the performance of the proposed metaheuristic algorithms to solve the proposed two-stage MIP model for the location of cross-docking centers and scheduling problems in the distribution network.

6.1. Evaluation of the MIP on small-sized instances

In this subsection, we are going to evaluate the proposed MIP model on ten instances. The general performance of the MIP model is evaluated with a set of small-sized instances which are randomly generated. The results obtained from MIP solved by CPLEX are compared with those of SA-GA and DE. The results are shown in Table 8.

Table 8
Computational results on small-sized instances

Problem No.	Mathematical model		Metaheuristics		
	CPLEX	RT (s)	SA-GA	DE	RT (s)
1	764	200	789	924	<1
2	820	450	830	925	<1
3	461	500	604	562	2
4	883	500	903	1070	2
5	599	500	641	711	2
6	691	700	699	769	5
7	392	700	415	430	6
8	700	900	894	817	6
9	481	900	647	537	10
10	823	900	1171	992	13

***RT (s):** running time (second)

This table also shows that metaheuristics have appropriate performance.

6.2. Evaluation of the metaheuristics on large-sized instances

Here, 18 test problems in the supply chain environment with various sizes are generated at random in small- and large-scale cases. In order to verify the statistical validity of the results shown in Table 9 and to confirm which algorithm is the best, a design of experiments and an analysis of variance (ANOVA) have been performed in which different algorithms are considered as a factor, and the response variable is RPD

.

Table 9
Running times for the second stage

Problem No.	SA-GA	DE	SA	GA
1	1.556	2.395	1.093	2.005
2	1.716	2.274	1.298	1.557
3	2.4	4.026	1.176	2.541
4	2.015	7.621	1.841	3.673
5	1.995	7.055	1.922	2.281
6	3.703	6.208	1.596	3.765
7	9.97	22.003	7.943	14.133
8	8.484	11.935	8.603	22.965
9	14.215	25.664	12.399	31.199
10	15.326	43.278	12.525	32.392
11	6.711	8.503	5.886	19.456
12	6.952	14.718	2.459	13.632
13	13.813	24.880	12.110	27.318
14	10.542	35.815	8.394	24.178
15	17.582	17.039	11.075	11.531
16	14.074	9.675	12.961	9.874
17	10.285	26.410	11.367	19.509
18	15.755	29.214	12.458	23.302

For better judgment, the running times in small- and large-sized test problems for the second stage are also reported in Table 10.

Table 10
Average relative percentage deviation (RPD) for algorithms

Problem no.	SA-GA	DE	SA	GA
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0.0248	0	0.0194	0
8	0	0	0	0
9	0	0	0	0
10	0.0223	0	0.0124	0
11	0	0	0	0
12	0.0173	0	0.0272	0
13	0.6032	0.1380	0.7286	0.1201
14	0.1290	0.3668	0.2892	0.0378
15	0.7577	0.4484	0.7132	0.0545
16	0.0109	0.0606	0.0520	0.0767
17	0.2575	0.2295	0.0998	0.1076
18	0.2555	0.0077	0.1102	0.0162

where RPD is obtained by the following formula:

$$RPD = \frac{A\lg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \tag{30}$$

where Alg_{sol} is objective function obtained by a given algorithm. Clearly, lower values of RPD are preferable.
The results demonstrate that there is a clear, statistically significant difference between the performances of the algorithms.

The mean plots and least significant difference (LSD) intervals (at 95% confidence level) for four algorithms are shown in Figure 4. As it can be seen, the hybrid SA-GA works better than others.

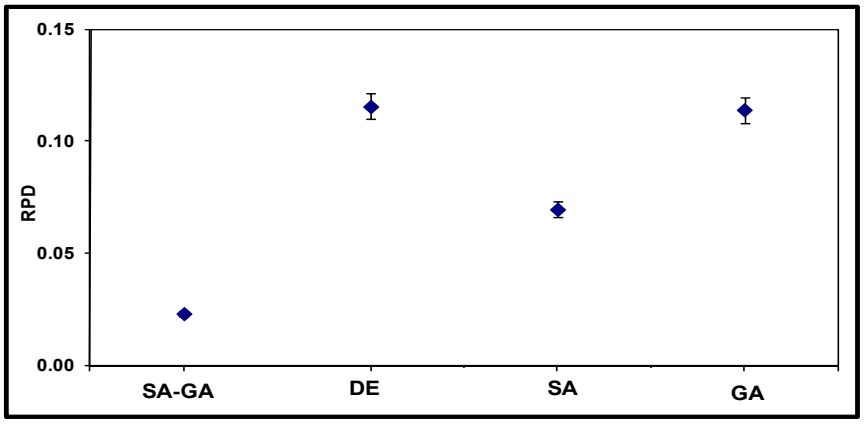

Fig. 4. Plot of $\overline{RPD}$ for the type of algorithm factor

In addition, to evaluate the performance of metaheuristic methods, the obtained RPDs of algorithms for each problem are shown in Figure 5.

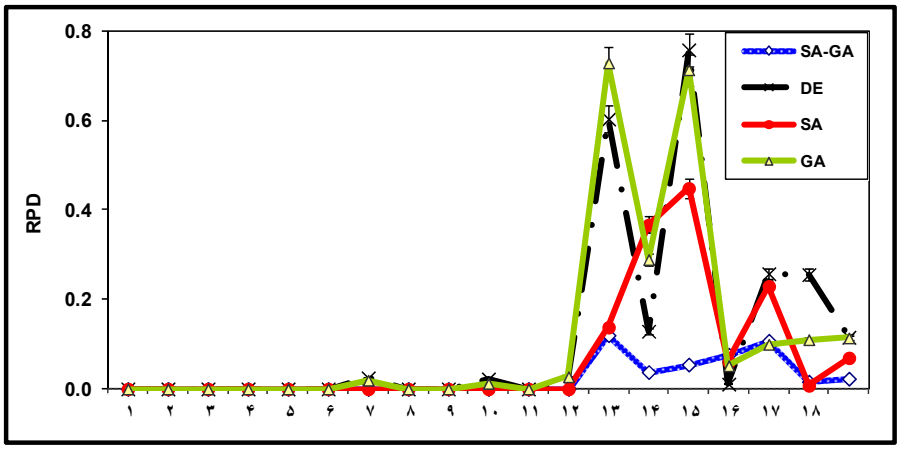

Fig. 5. RPD's of algorithms

Comparing the running time as we see from Figure 6, simulated annealing algorithm is the best algorithm, because it requires minimum time to obtain the near-optimum solution.

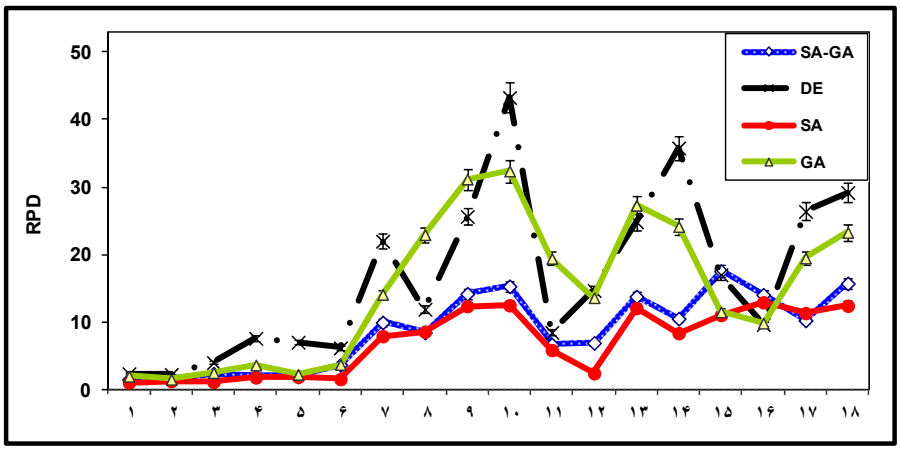

Fig. 6. Running times of algorithms

7. Conclusions and Future Researches

This paper proposed the location-allocation and scheduling models in cross-docking distribution networks. To solve the presented models, four metaheuristic algorithms, such as simulated annealing (SA), genetic algorithm (GA), differential algorithm (DE), and hybrid algorithm of SA and GA, were applied. The result showed that simulated annealing is the best algorithm between four algorithms. Future research can be recommended in a few directions. It is interesting to consider time-window constraints in the proposed cross-docking distribution network. Another extension is to take uncertain parameters into account such as covering radius.

References

Amini, A.&Tavakkoli-Moghaddam, R. (2016). A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks, Computers & Industrial Engineering, 96, 180-191.

Apte, U.&Viswanathan, S. (2000). Effective cross-docking for improving distribution efficiencies, International Journal of Logistics Research and Applications, 3 (3), 291-302.

Assadi, M.T.&Bagheri, M. (2016).Differential evolution and Population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems, Computers & Industrial Engineering, 96, 149-161.

BolooriArabani, A. Zandieh, M. &FatemiGhomi, S.M.T. (2011).Multi-objective genetic-based algorithms for a cross-docking scheduling problem, Applied Soft Computing, 11(8), 4954-4970.

Chen, R. Fan, B.& Tang, G. (2009). Scheduling problems in cross docking, Combinatorial Optimization and Applications, Lecture Notes in Computer Science, 5573, 421-429.

Kirkpatrick, S. Gelatt, C.D.&Vecchi. M. P. (1983).Optimization by simulated annealing. Science, 220, 671–680.

Kuo, Y. (2013) Optimizing truck sequencing and truck dock assignment in a cross docking system, Expert Systems with Applications, 40(14), 15, 5532-5541.

Li, Y. Lim, A.& Rodrigues, B. (2004).Crossdocking - JIT scheduling with time windows, Journal of Operational Research Society, 10(1057), 1-10 .

Liao, T.W. Egbelu, P.J.& Chang P.C. (2012). Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations, Applied Soft Computing, 12(11), 3683-3697.

Liao, T.W. Egbelu, P.J.& Chang, P.C. (2013). Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations, International Journal of Production Economics, 141(1), Pages 212-229.

Lichtblau, D. (2009).Relative Position Index Approach.Davendra, D. and Onwubolu G (eds.) Differential Evolution A Handbook for Global Permutation-Based Combinatorial Optimization, pp. 81–120, Springer, Heidelberg.

Marín Á.&Salmerón, J. (1996). A simulated annealing approach to the railroad freight transportation design problem, International Transactions in Operational Research, 3(2), 139–149.

Metropolis, N. Rosenbluth, A. W. Rosenbluth, M. N. Teller, A. H.& Teller, E. (1953).Equation of state calculations by fast computing machines. Journal of Chemical Physics, 21, 1087-1092.

Mohtashami, A., (2015). Scheduling trucks in cross docking systems with temporary storage and repetitive pattern for shipping trucks, Applied Soft Computing, 36, 468-486.

Price, K.V. Storn, R.M.&Lampinen, J.A. (2005). Differential Evolution. A Practical Approach to Global Optimization. Springer-Verlag, Berlin-Heidelberg.

RahmanzadehTootkaleh, S. FatemiGhomi, S.M.T.&Sheikh Sajadieh, M. (2016).Cross dock scheduling with fixed outbound trucks departure times under substitution condition, Computers & Industrial Engineering, 92, 50-56.

Storn, R.& Price, K.V. (1995).Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR95-012, International Computer Science Institute (ICSI).

Yu, W. (2002) Operational strategies for cross docking systems. Dissertation, Iowa state University. Ames, IA, USA.

Yu, W.&Egbelu, P.J. (2008).Scheduling of inbound and outbound trucks in cross-docking systems with temporary storage, European Journal of Operational Research, 184(1), 377-396.

This article can be cited: FatemiGhomi, S.M.T., Behnamian J., F. Jolai&Heidary, P.(2018). Location-Allocation and Scheduling of Inbound and Outbound Trucks in Multiple Cross-Dockings Considering Breakdown Trucks.*Journal of Optimization in Industrial Engineering.*11 (1), 51-65.

URL: http://www.qjie.ir/article_535407.html
DOI: 10.22094/JOIE.2017.594.1382