Indian Journal of Science and Technology, *Vol 8(12), IPL010, June 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Hybrid Approach for Cross-Docking Scheduling

Tokhmehchi Nafiseh[1*] and Makui Ahmad[2]

[1]Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran; nafiset@ymail.com

[2]Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran; amakui@iust.ac.ir

Abstract

Background/Objectives: Cross-docking is an innovative logistical strategy, which means discharging the cargos from in-bound trucks into the warehouse directly and saved automatically. This article investigates the scheduling of a distribution system consisting of suppliers, cross-docks and the ultimate destinations. The aim of analyzing this problem is presenting a scheduling model so that lateness, earliness and holding costs will be minimized. **Methods/Statistical Analysis:** Restrictions are applied on the number of arrival and exit doors, departure time, release time, earliness, lateness, and the required space for discharging. Since the offered model is NP-Hard and highly complicated, the hybrid form of firefly and genetic algorithms is applied to solve the model. **Results:** Tuning the parameters performed and several numerical examples shows the appliance of this model. Comparisons via ANOVA and interval plot show that hybrid form can act better than GA and FF in large sizes. **Conclusion/Application:** The innovation of this article is thanks to its model type and solution method.

Keywords: Crossdocking, Hybrid Firefly and Genetic Algorithms, Logistics, Metaheuristic, Scheduling

1. Introduction

Cross-docking is a creative distribution strategy for the goods that are time-sensitive. This method works as follows. The in-bound trucks are discharged in a cross-dock, and directly reloaded to the out-bound trucks with minimum standby time (standby time is commonly shorter than 24 hours). The following costs are reduced in this strategy: Warehousing, maintenance, transportation, and human resources. This method also leads to shorter lateness in delivery, improving client satisfaction, needing less warehousing room, smoother work flow, as well as lower risk and loss. The benefits of cross-docking system are more conspicuous in cases with higher distribution costs. A good example holds for car manufacturers. In order to meet their needs of their assembly lines and distribution centers, they have been increasingly attracted to cross-docks. These docks are usually run by logistical companies in order to manage on-time delivery of goods.

Efficient operations of cross-docks require accurate and simultaneous operations of in-bound and out-bound trucks by contriving proper scheduling approaches. For this purpose, several scheduling procedures have been introduced over the recent years. In general, scheduling helps taking advantage of limited resources in a certain time period so to optimize the objectives.

The scheduling process for the trucks commonly divides transportation procedure into two parts: 1) Discharging in-bound trucks, 2) Loading out-bound trucks. Of course there is a time lag for conveying the cargos inside the cross-dock terminal. This delay includes the required time for investigating, categorizing and carrying the materials.

Each terminal has several doors. The doors are scheduled for the trucks. The distance between each pair of doors is already determined; hence, there is another delay for carrying the materials between the pairs of doors as well[1].

On the whole, scheduling is a new horizon in cross-docking systems. It helps the manager's transport and distributes their cargos faster. This improves the overall efficiency of the organization.

The remainder of the paper is organized in the following manner. In Section 2 we briefly discuss the relevant literature of cross-docking scheduling. Section 3 presents the problem and mathematical formulation of this.

**Author for correspondence*

Comprehensive explanations of solution methodology based on hybridization of genetic algorithm and firefly algorithm is discussed in section 4. Section 5 is dedicated to tuning the parameters and numerical examples. Finally in section 6 conclusions is presented and some directions for future work are discussed.

2. Literature Review

There are not many researches about cross-docking. For example up until 2005, no article has been published about short-term scheduling for the trucks[1]. The current part of this study has a brief shot about several researches on cross-docking. McWilliams et al.[2] were considered in-bound truck scheduling for cross-docks of postal parcels in postal industries. In this system, the parcels were conveyed to out-bound trucks using fixed conveyor belt. The objective of this scheduling model was minimizing the time between discharging the first parcel till loading the last postal parcel. Also, in order to detect the amount of delays, a solution based on genetic algorithm was presented. In the end, numerical calculations showed that their suggested method has lead to significant reduction of delays. Yu and Egbelu[3] studied a cross-docking system with one arrival and one exit door. The objective was minimizing the total operation time. The cargos were considered to be interchangeable. Assigning the cargos to trucks and the travel-time between doors were supposed to be already determined. The only differences were the truck interchange time, and optional sequence for discharging the in-bound trucks. In the end, the problem was formulated with a mixed-integer model, and solved using a heuristic algorithm. In Wang and Regan[4] a scheduling model was contrived for in-bound trucks. Several dispatching rules, required for real dynamic environments, were presented as follows: Among the few in-bound trucks that are waiting for discharging their cargos, only the first truck was considered. The next trucks were selected with "First-com first-served" policy. This method pays attention only to the standby time of in-bound trucks, which does not influence the whole system significantly. In the end, 2 algorithms were introduced for solving the problems. Chen and Lee[5] investigated the problem of cross-docking flow shop scheduling aimed to find the optimum sequence of in-bound and out-bound trucks that could minimize makespan. The discharge and loading times were considered to be different for each truck. In some cases, this included travel time as well. Also, it was assumed that all the trucks were available in the beginning of scheduling horizon. The authors of this article proved that the problem in question is NP Hard; hence, a heuristic approached based on *Johnson's* law was suggested for solving it.

Vahdani and Zandieh[6] considered the previous problem and used 5 meta-heuristic approaches for solving the model. The answer of Problem[3] was used as the initial answer. It was proved that these meta-heuristic methods improve the results, with slightly longer run time.

Boysen & Fliedner[1] were presented an optimizing model. This model assigns fixed schedule for out-bound trucks, with the aim to minimize the number of delayed transportations. This model considered the travel time between arrival and exit doors as well. They also proved that this model is NP Hard.

In order to reduce the number of goods passing standby area, a sequence was presented for in-bound and out-bound trucks by Forouharfard and Zandieh[7]. Also Imperialist Competitive Algorithm was used for solving the problem. Vahdani et al.[8] were considered a similar problem, and formulated it as a mixed-integer programming model. They used genetic algorithm and electro-magnetism algorithm for solving the problem. Two meta-heuristic algorithms, including simulated annealing, and variable neighborhood algorithm were used for solving the similar problem by Soltani and Sajadi[9]. Arabani et al.[10] assumed that all out-bound trucks have due date. The target function aimed to minimize the earliness and lateness for trucks. Three meta-heuristic approaches were demonstrated for solving the problem. These approaches included genetic algorithm, evolutionary algorithm, and particle swarm algorithm. A similar problem was considered by Arabani et al[11]. The time span was sectioned into several parts. This problem was formulated as an NP Hard model. It used break down approach, including two segments: 1. Finding a fixed sequence for in-bound trucks; 2. Determining the sequence of out-bound trucks. Those subsets could find a near-optimum answer with heuristic algorithms, and the accurate answer with Dynamic Planning Approach. Alpan et al.[12] were considered a cross-dock with several arrival and exit doors. The aim of analyzing this problem was finding an optimum or near-optimum answer for the scheduling model, so to minimize the total costs of conveyance operations. Agustina et al.[13] developed a model for scheduling a cross-dock with delivery time window and temporary storage. Their target function aims to minimize distribution cost and holding cost. The resulted model was Mixed-Integer. They used Lingo for solving the model.

Li et al. (2012) introduced 3 different aspects of optimum planning, including trucks-grouping, clustering, and in the end determining the sequence of trucks and assigning the trucks to the docks in cross-docking system. McWilliams and McBride[13] used beam search algorithm for solving the problem of scheduling postal parcel distribution centers, with the aim to minimize the time span of the transfer operation in the terminal. They finally compared their suggested algorithm with other approaches. Bellanger et al.[14] were used a compound 3-step flow shop model with aim to minimize completion time. They also used branch and bound algorithm. This article investigates the problem of truck scheduling in uncertainty conditions of truck arrival for in-bound trucks. Buijs et al.[15] focused on the problem of correlation among different aspects of cross-docking problem, with the aim to support and develop the future practical decisions. They also presented an overall classification of decision trends in this area. Miao et al.[16] used adaptive tabusearch to solve the problem of contriving a cross-docking management system. Some numerical tests proved that the suggested solution, both with regards to effectiveness and efficiency, overcomes CPLEX solution.

In previous models of cross-dock scheduling, the discharge area of in-bound trucks was considered to be unlimited, but in the real industrial world such condition is completely impossible. In order to address this research gap, a mathematical model was presented. The mathematical model of this article is a development to[13] model. Since the previous model was NP Hard, the new one is obviously NP Hard as well. We can use metaheuristic algorithms for solving this problem. The difference of the mathematical model presented in this article, with the one suggested by [13] was considering the assumption of having limited discharge space. This model minimized the earliness and lateness costs, as well as holding costs. Additionally, it considers the condition of limited discharge space.

In the next part we will introduce the mathematical model and its conditions.

3. The Suggested Mathematical Model

In the beginning of this plan's time-scope, the retailers (ultimate targets) order their requirements to the suppliers. The suppliers will also send the cargos from the docks using trucks. These trucks, after arriving in the warehouse environment, either discharge their cargo to be directly reloaded to outgoing trucks, or stay in the warehouse for a period, usually less than 24 hours. Other conditions and assumptions in this problem are as follows:

Unlike the time of exiting from warehouse that is variable, the arrival time to the warehouse is already determined.

The warehouse space for discharging the incoming trucks is limited, but it is unlimited for outgoing trucks.

The number of incoming and outgoing trucks is limited.

3.1 Notations, Parameters and Variables

The following notations, parameters, and variables are used for mathematical model of the problem:

i Index of items $i = 1, 2, \dots, I$

n Index of inbound doors $n = 1, 2, \dots, N$

m Index of outbound doors $m = 1, 2, \dots, M$

z_i Earliest delivery time for item i (hours)

d_i Latest delivery time for item i (hours)

v_i Arrival time for item i (hours)

α Earliness cost rate

β Lateness cost rate

δ Inventory holding cost rate

H Planning horizon $h = 1, 2, \dots, H$ (hours)

f_i Required space for item i (m^2)

F Total space in in-bound yard (m^2)

Decision Variables:

r_i Release timetime for item i (hours)

a_i Departure time for item i (hours)

e_i Earliness for item i (hours)

t_i Lateness for item i (hours)

x_{inh} If item i loaded at inbound dock door n at time h = 1;

x_{inh} = 0, otherwise

x_{imh} If item i unloaded at outbound dock door m at time h =1; x_{imh} = 0, otherwise

3.2 Formation of the Model

The mathematical model of the problem becomes:

Minimize

$$TC = \alpha\sum_{i=1}^{I} e_i + \beta\sum_{i=1}^{I} t_i + \delta\sum_{i=1}^{I}(a_i - r_i) \tag{1}$$

ST:

$$\sum_{h=1}^{H}\sum_{n=1}^{N} x_{inh} = 1 \quad i = 1, 2, \dots, I \tag{2}$$

$$\sum_{h=1}^{H}\sum_{m=1}^{M} x_{imh} = 1 \quad i = 1, 2, \ldots, I \tag{3}$$

$$\sum_{n=1}^{N}\sum_{i=1}^{I} x_{inh} \le N \quad h = 1, 2, \ldots, H \tag{4}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I} x_{imh} \le M, \quad h = 1, 2, \ldots, H \tag{5}$$

$$\sum_{n=1}^{N}\sum_{i=1}^{I} f_i x_{inh} \le F \quad h = 1, 2, \ldots, H \tag{6}$$

$$a_i \le H, \quad i = 1, 2, \ldots, I \tag{7}$$

$$v_i \le r_i \le a_i, \quad i = 1, 2, \ldots, I \tag{8}$$

$$a_i - d_i \le t_i - e_i, \quad i = 1, 2, \ldots, I \tag{9}$$

$$z_i - a_i \le t_i - e_i, \quad i = 1, 2, \ldots, I \tag{10}$$

$$x_{inh}, x_{imh} = 0, 1 \tag{11}$$

$$r_i, a_i, e_i, t_i \ge 0 \tag{12}$$

The objective function of this problem is regarded as cost, and consists of 3 elements: Earliness penalty cost, lateness penalty cost, and cargo warehousing cost. Equation 2 indicates that every cargo uses only one arrival door for discharging.

Constraint 3 guarantees that each cargo will use only one exit door for reloading. Constraint 4 shows that the numbers of discharging procedures are fewer than the total number of arrival doors. Constraint 5 makes sure that the number of reloading procedures will not exceed the number of exit doors.

Next restrictions guarantee that the space required for discharging in-bound cargos is limited. Constraint 7 demonstrates departure time is within the time scope of the plan. Constraint 8 displays that release time is a number between arrival time and departure time. Constraints 9 and 10 are respectively used for determining earliness and lateness. Equations 11 and 12 display the range of decision variables.

4. Solution Method

Like mentioned before, the mathematical model of this problem is NP Hard (Garey & Johnson, 1979). It has complicated search space. In other words, accurate solution can be available only for small scales of this problem. For larger scales, other solutions must be contrived. This article uses Hybrid form of Firefly Algorithm and Genetic Algorithm as a solution method; therefore, in this section a brief description of each innovative solution will be raised. In each section, the solution structure will be reviewed. In the end the hybrid solution will be suggested.

4.1 Genetic Algorithm

Genetic Algorithm (GA) is a Search_algorithm heuristic belong to the larger class of evolutionary algorithms (Gen, 1997). In this approach, a random solution will be selected initially. Afterwards, the fitness function value will be calculated for each chromosome of the initial population. A suitable number of chromosome pairs shall be selected in this phase according to their fitness levels, so to be used in later steps.

In the next phase, crossover operator with probability of P_c will act upon the parent chromosomes, and combines them to create new chromosomes (offspring). Then mutation with probability of P_m will happen on the chromosomes created from crossover operation. Actually, by changing the bits of these chromosomes, a new way for entering new data will be achieved. Then, in order to evaluate the offspring, the fitness level of new chromosomes shall be calculated.

In the next level, a new population will be selected for the input of next phase. If the end conditions of algorithm happen, the algorithm will end up; otherwise, the available population will be used as the input population of the next phase.

4.1.1 Initial Population

In this paper, just like many genetic problems, the initial population is created randomly. The reason is creating a large range of solutions. Of course the information from system's users can be used as well. Their suggestions and comments might be applied to the initial population. An accurate solution shall also help.

4.1.2 Chromosome Structure

According to the decision variables of this model, the following chromosome structure is suggested. Each row indicates the variable values for one type of product. For example the elements of second row are respectively: Release time, departure, earliness, and lateness of product 2. The last 2 indexes show having or not having discharged

$$\begin{Bmatrix} r_1 & a_1 & e_1 & t_1 & x_{1nh} & x_{1mh} \\ r_2 & a_2 & e_2 & t_2 & x_{2nh} & x_{2mh} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_I & a_I & e_I & t_I & x_{Inh} & x_{Imh} \end{Bmatrix}$$

Figure 1. The chromosomal structure.

product 2 in door number n and in the time horizon h. The next member means having or not having reloaded at door number n and in time horizon h.

4.1.3 Survivability of Each Chromosome

Since the Fitness Function is a negative aspect, it must be minimized. In order to calculate survivability, number 1 is divided by Fitness Function Tc. The result will be divided by the sum of each and every quotient. The quotient is obtained from dividing 1 by every Tc.

$$P_i = \frac{\frac{1}{TC_j}}{\sum_{j=1}^{pop-size} \frac{1}{TC_j}} \qquad j = 1, \ldots, pop - size \tag{13}$$

4.1.4 Producing the Next Generation

Three operators are reviewed here as follows:

1) Copy: A% of the best chromosomes from previous generation with the highest survivability will be copied without any change.
2) Cross-over: B% of chromosomes in each generation will be selected from the best of previous generation, and crossed over each other in two-by-two couples. This way, B offspring will be produced. Among B offspring and B parents, B chromosomes with the optimum Fitness Function will be selected.

   The cross-over method in this article is as follows: A chromosome consisting of 0 and1 will be generated randomly. Then the locations from each of the two parents, which are associated with number 1, will be substituted with each other. Other elements will remain fixed. In result, two new offspring will be obtained.
3) Mutation**:** C% of next generation chromosomes will be generated with mutation. There are numerous methods for mutating.

This study generates a random chromosome using the genes associated with numbers between 0 and 1. Then the genes with mutation probability less than P_m will be selected. The parent chromosomes in similar conditions will change randomly.

4.1.5 Investigation of Stop Condition

In this article, the algorithm stops when a certain number of loops happen.

4.2 Firefly Algorithm

Firefly algorithm is an evolutionary model, based on Swarm intelligence algorithms, originated from nature. This algorithm was introduced by Yang in 2007. Its main idea arose from the optical relation between fireflies. In this algorithm, cooperation (and probably competition) of simple and low-intelligence members lead to higher levels of intelligence, that could not be achieved by any of the members alone.

This algorithm is based on getting to optimum limit when investigating behaviors and characteristics of fireflies. It consists of 3 main attributes: 1) Firefly becomes lighter and more attractive when it moves randomly. 2) The beauty and attractiveness of fireflies are proportional to their brightness and light level. This attraction decreases by increasing the distances (of course this comparison is from the view point of other fireflies). 3) The light intensity decreases with the rise in light absorptive factor. It is controlled with a certain scale.

Fireflies produce rhythmic and short flashes of light. The flashing pattern of each firefly is different from the others. The search module of firefly algorithm is as follows. Each firefly is compared with each and every other firefly. If the sample firefly has lower light than the compared firefly, it goes toward the one with stronger light (the problem of finding maximum point). This action results in concentration of the particles around the particles with stronger light. In the next loop of algorithm, if a particle with stronger light appears, again the other particles will move toward it. There is a failing in this method. The movements of the particles are adjusted only according to the light level of one firefly, which is in fact the local optimum. The global optimum has no effect on the search module of this algorithm; therefore, the problem space will not be searched optimally, and more loops will be required for finding the optimum point.

4.2.1 Steps of Firefly Algorithm

The details of applied algorithm are as follows:

A- Producing the initial solution and appraising them (Note that the structure of each firefly is similar to Figure 1.)

B- For every other firefly (like 'a') and for every firefly (like 'b'), we calculate light intensity according to (14).

$$I = I_0 e^{-\delta r^2} \tag{14}$$

In (14) I_0 is Intensity Coefficient Base Value, δ is Light Absorption Coefficient and r is the distance to light source. In addition light absorption coefficient typically changes from 0.01 to 100 (Yang, 2010).

C-If I_a is less than I_b the new answer generate according to (15).

$$x' = x_a + \beta_0 e^{-\delta r^2}(x_b - x_a) + \alpha\varepsilon \tag{15}$$

In (15) x_a, x_b show the position of firefly a,b.ε is a random vector and β_0 Attraction coefficient base value. In addition α is mutation coefficient and $\alpha \in [0,1]$

Note that attraction coefficient base value usually considered 1 and random vector is equal to [rand-1/2] (Yang, 2010).

D- Appraising of new population and selecting the best solution.

E- Loop to step 2, if the termination conditions are not met.

4.3 The Hybrid Form of Genetic Algorithm and Firefly Algorithm

The hybridization process of the two algorithms is like the process of humans bearing child, with the difference that a hybrid algorithm might have more than 2 parents. These algorithms inherit part of the parents' characteristics. The target of algorithms' hybridization is to create or improve new properties that are not existing in a singlealgorithm[17].

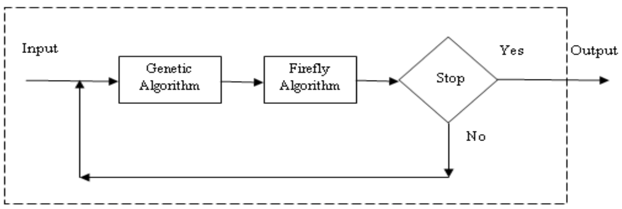

Figure 2. Iterative sequentialization hybrid.

Among different hybrids between algorithms, combining via genetic algorithm can improve the search performance and also it is a fortune to hybrid optimization to gain their objectives[18].

The range of hybrid algorithm methods is enormous. This article applies Hybrid Iterative Sequentialization to its problem, with the help of Matlab software scripting.

According to the operations and attributes of the two algorithms, firefly and genetic, considering the results of numerous tests, we found out that firefly algorithm demonstrates more intention to exploit space. It has similar characteristics to crossover in genetic algorithm; hence, the results from hybrid method of Iterative Sequentialization seem to have higher quality.

The hybrid form of Iterative Sequentialization algorithm acts as follows. First of all, the output of genetic algorithm will be used as the input of firefly algorithm. Then the output of firefly algorithm will be used as the input of genetic algorithm. This cycle continues until predetermined number of loops for calculating the target function will be passed. The following diagram shows a schematic view of Hybrid Iterative Sequentialization method.

4.3.1 The Step of HGF

The steps of Iterative sequentialization Hybrid of Genetic and Firefly algorithms (HGF) are shown below.

Produce an initial population (p) randomly.
While (not stop condition)
For p_g=1: P
Mate p_c percent of population and evaluate them.
Mutate p_m percent of population and evaluate them.
Combine population in a single set and rank them.
Shorten best members and produce new population of size P.
End for p_g
Set ultimate population as fireflies.
For p_f=1: P
For p_l=1: P
If (I_f<I_l)
Generate firefly and update it according to (15).
End if
End for p_l
End for p_f
Sort the solution and select the best.
End while.

5. Computational Result

This section divided in 3 parts. First general inputs are given. Next parts tunning the parameters via taguchi methods implemented and finally several examples in 3 sizes are given to illustrate the application of the proposed methodology in real-world environments.

5.1 General Inputs

Table 1 shows the general data to solve the problem.

5.2 Tunning the Parameters

In order to apply proposed algorithms in best mode, a parameter-tuning method named taguchi design was used (Taguchi et al., 2005). In these way parameters of 3 types of hybrid identified. Proposed hybrids are based on Genetic and Firefly Algorithms, so the vital factors derived from them. Because of considering 7 parameters with 3 levels, taguchi present 27 tests to be solved. Following Tables show the level of parameters and the

Table 1. General data

parameters	Value
H	20
v_i	Uniform(2,5)
Z_i	Uniform(5,8)
d_i	Uniform (8,12)
α	3000
β	12000
δ	2400
f_i	Uniform(1,4)
F	1000

Table 2. Levels of parameters

	parameter	1	2	3
Hybrid of GA and FF	C_{GA}: Crossover Percentage	0.3	0.5	0.8
	M_{GA}: Mutation Percentage	0.15	0.2	0.25
	I_{GA}: Number of GA Iterations	50	100	150
	S: Population size	50	80	100
	LC_{FF}:Light Absorption Coefficient	0.01	50	100
	M_{FF}: Mutation Coefficient	0.1	0.5	0.9
	I_{FF}: Number of FF Iterations	50	100	150

Table 3. Obtained values in parameter tuning

	C_{GA}	M_{GA}	I_{GA}	S	LC_{FF}	M_{FF}	I_{FF}	ISHGF
1	0.3	0.15	50	50	0.01	0.1	50	0.4736
2	0.3	0.15	50	50	50	0.5	100	0.5469
3	0.3	0.15	50	50	100	0.9	150	0.8978
4	0.3	0.2	100	80	0.01	0.1	50	0.3472
5	0.3	0.2	100	80	50	0.5	100	0.2571
6	0.3	0.2	100	80	100	0.9	150	0.3904
7	0.3	0.25	150	100	0.01	0.1	50	1.0334
8	0.3	0.25	150	100	50	0.5	100	0.9002
9	0.3	0.25	150	100	100	0.9	150	0.1787
10	0.5	0.15	100	100	0.01	0.5	150	0.9064
11	0.5	0.15	100	100	50	0.9	50	0.5974
12	0.5	0.15	100	100	100	0.1	100	0.3904
13	0.5	0.2	150	50	0.01	0.5	150	0.0000
14	0.5	0.2	150	50	50	0.9	50	0.2825
15	0.5	0.2	150	50	100	0.1	100	0.2067
16	0.5	0.25	50	80	0.01	0.5	150	0.0930
17	0.5	0.25	50	80	50	0.9	50	0.6035
18	0.5	0.25	50	80	100	0.1	100	0.9789
19	0.8	0.15	150	80	0.01	0.9	100	0.7103
20	0.8	0.15	150	80	50	0.1	150	0.9098
21	0.8	0.15	150	80	100	0.5	50	1.0844
22	0.8	0.2	50	100	0.01	0.9	100	0.4278
23	0.8	0.2	50	100	50	0.1	150	0.6776
24	0.8	0.2	50	100	100	0.5	50	0.9637
25	0.8	0.25	100	50	0.01	0.9	100	0.7795
26	0.8	0.25	100	50	50	0.1	150	0.2101
27	0.8	0.25	100	50	100	0.5	50	0.4736

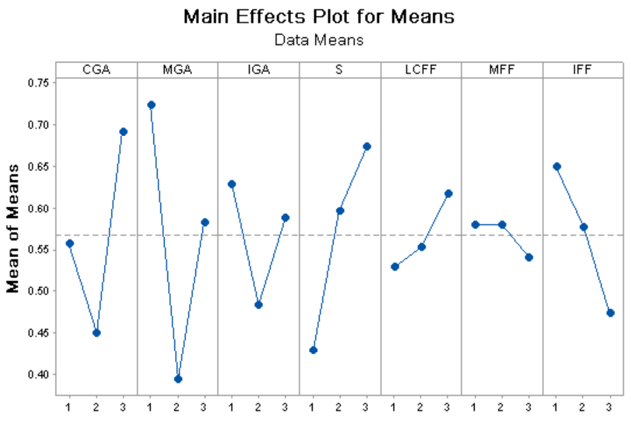

Figure 3. Main effects plot for means ISHGF.

results of 27 runs in related percentage deviation (RPD) form in (16).

$$RPD = \frac{|A\lg orihm\,Sol - Best\,Sol|}{Best\,Sol} \times 100 \qquad (16)$$

Finally, number achieved from 27 runs with different parameter values create following mean of means plot with aim to MINITAB 17 Software.

Lowest levels of every parameter in mean of means plot has the best effects on final solution. Considering this fact the best value reported in Table 4.

5.3 Computational Results

In this section 18 different problems in 3 sizes (small, medium and large) solved with HGF algorithm to show the appliance of proposed algorithm. In this way every test runs for 3 times and we report the average of them.

Table 4. The best parameter values

Algorithm	C_{GA}	M_{GA}	I_{GA}	S	LC_{FF}	M_{FF}	I_{FF}
ISHGF	0.5	0.2	100	50	0.01	0.9	150

Table 5. Final solution

Tests	I	N	M	HGF	GA	FF
Small size						
1	3	4	3	2.88	3.8743	3.1358
2	4	4	5	3.84	4.4844	4.2905
3	5	4	6	4.8	5.4919	5.4056
4	6	6	7	5.76	6.2629	6.0991
5	7	5	8	6.72	7.6728	7.6109
Medium size						
6	20	13	17	19.2001	26.2401	25.3363
7	22	13	12	21.1034	27.7803	27.0147
8	24	15	13	23.0301	30.0847	30.2837
9	26	15	15	24.961	33.7748	32.6843
10	28	16	14	26.7347	36.0363	33.6844
11	30	16	14	28.7928	43.1811	38.4040
12	32	18	16	30.7302	54.8773	53.3551
Large size						
13	45	20	21	43.2009	101.7391	98.6454
14	50	21	22	48.0006	121.3595	119.3855
15	60	25	24	57.6004	141.8014	134.4125
16	70	26	30	67.3142	162.8291	151.0233
17	80	30	35	76.8001	176.3561	172.1651
18	100	50	60	96.0318	221.7715	203.8707

5.4 Comparisons

In order to compare between 3 solutions, ANOVA Statistical Test was devised with SPSS Software in addition we consider each size separately to apply more focus on it. Following tables and diagrams show the comparisons between algorithms.

From the results obtained by ANOVA in Table 6 (sig>0.05) and interval plot in Figure 4 we can claim that there is not significant difference between 3 proposed solution in small size.

The results obtained by ANOVA in Table 7 shows there is significant difference between algorithms (sig<0.05) but because of overlapping between algorithms in Figure 5 we cannot certainly claim that ISHGF is the best in medium size.

Because of value of sig (<0.05) in Table 8 and not existence of overlapping in interval plot of large size we can report that ISHGF can act better among their parents.

Following diagram shows the convergance path of problem number 5.

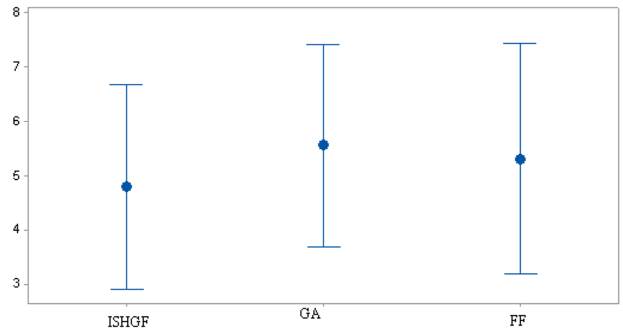

Figure 4. Interval plot for small size.

Table 6. ANOVA of small size

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1.490	2	.745	.299	.747
Within Groups	29.870	12	2.489		
Total	31.359	14			

Table 7. ANOVA of medium size

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	500.177	2	250.088	3.613	.048
Within Groups	1246.009	18	69.223		
Total	1746.186	20			

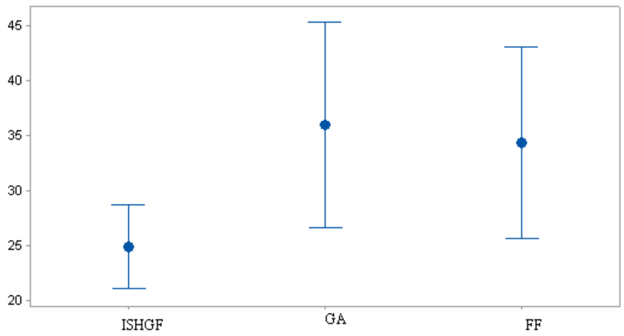

Figure 5. Interval plot for medium size.

Table 8. ANOVA of large size

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	14204.266	2	7102.133	3.888	.044
Within Groups	27402.139	15	1826.809		
Total	41606.404	17			

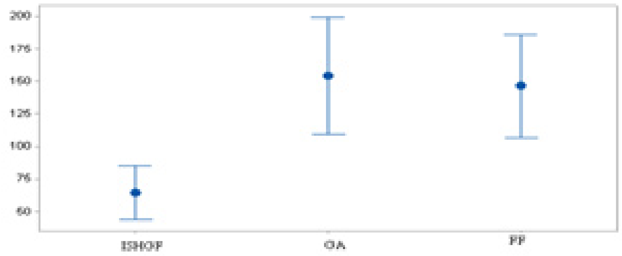

Figure 6. Interval plot for large size.

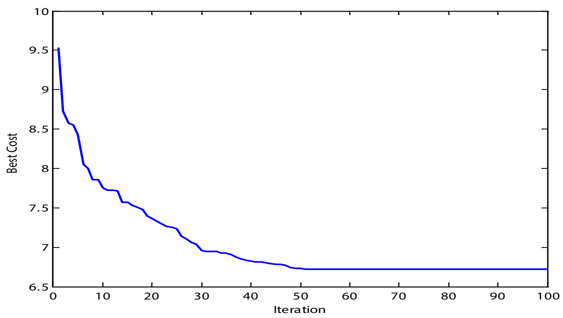

Figure 7. Convergancepath of ISHGF.

6. Conclusion and Suggestions for Future

This article reviewed a cross-docking system, in which the cargos are discharged from in-bound trucks and reloaded on out-bound trucks with minimum standby interval. This problem has been analyzed with the aim to develop a scheduling model, which minimizes lateness and earliness times, in common with holding cost. Restrictions have been applied to the number of arrival and exit doors, departure time, release time, earliness, lateness, and the space required for discharging.

Since the problem model is NP Hard, which is highly complicated, two metaheuristic algorithms were used: Genetic and Firefly. These algorithms were hybridized with Iterative Sequentialization to solve the problem. In addition to apply proposed algorithms in best mode, a parameter-tuning based on taguchi design was applied. Finally 18 test problems in 3 sizes solved via ISHGF, GA and FF. Comparisons via ANOVA and interval plot show that HGF can act better than GA and FF in large sizes.

Future studies might investigate the scheduling function becoming multi-objective. For example decreasing the number of delayed transportations can be another objective. In addition the model can extend with considering social or environmental issues. Changes in the solution approach can help develop the problem further.

7. References

1. Boysen N, Fliedner M. Cross-dock scheduling: Classification, literature review and research agenda. Omega. 2010; 38(6):413–22.
2. McWilliams DL, Stanfield PM, Geiger CD. The parcel hub scheduling problem: A simulation-based solution approach. Computers & Industrial Engineering. 2005; 49(3):393–412.
3. Yu W, Egbelu PJ. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. European Journal of Operational Research. 2008; 184(1):377–96.
4. Wang JF, Regan A. Real-time trailer scheduling for cross dock operations. Transportation Journal. 2008; 8:5–20.
5. Chen F, Lee CY. Minimizing the makespan in a two-machine cross-docking flow shop problem. European Journal of Operational Research. 2009; 193(1):59–72.
6. Vahdani B, Zandieh M. Scheduling trucks in cross-docking systems: Robust meta-heuristics. Computers & Industrial Engineering. 2010; 58(1):12–24.
7. Forouharfard S, Zandieh M. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. The International Journal of Advanced Manufacturing Technology. 2010; 51(9-12):1179–93.
8. Vahdani B, Soltani R, Zandieh M. Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. The International Journal of Advanced Manufacturing Technology.2010; 46(5-8):769–83.

9. Soltani R, Sadjadi SJ. Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. Transportation Research Part E: Logistics and Transportation Review. 2010; 46(5):650–66.
10. Arabani AB, Ghomi SF, Zandieh M. A multi-criteria cross-docking scheduling with just-in-time approach. The International Journal of Advanced Manufacturing Technology. 2010; 49(5-8):741–56.
11. Arabani AB, Ghomi SF, Zandieh M. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. Expert systems with Applications. 2011; 38(3):1964–79.
12. Alpan G, Larbi R, Penz B. A bounded dynamic programming approach to schedule operations in a cross docking platform. Computers & Industrial Engineering. 2011; 60(3):385–96.
13. Agustina D, Lee CKM, Piplani R. Cross-docking scheduling with delivery time window and temporary storage. 2011 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM); 2011 Dec. p. 131–5.
14. Bellanger A, Hanafi S, Wilbaut C. Three-stage hybrid-flowshop model for cross-docking. Computers & Operations Research. 2013; 40(4):1109–21.
15. Buijs P, Vis IF, Carlo HJ. Synchronization in cross-docking networks: A research classification and framework. European Journal of Operational Research. 2014; 239(3):593–608.
16. Miao Z, Cai S, Xu D. Applying an adaptive tabu search algorithm to optimize truck-dock assignment in the cross-dock management system. Expert Systems with Applications. 2014; 41(1):16–22.
17. Eshghi K, Kariminasab M. Combinatorial Optimization & Metaheuristics. Tehran: MehrAzinPress; 2012. p. 125–38.
18. Lobo FG, Goldberg DE. Decision making in a hybrid genetic algorithm. IEEE International Conference on Evolutionary Computation. 1997; 8:121–5.