IJE TRANSACTIONS B: Applications Vol. 31, No. 8, (August 2018) 1258-1266

International Journal of Engineering

Journal Homepage: www.ije.ir

New Approaches in Metaheuristics to Solve the Truck Scheduling Problem in a Cross-docking Center

H. Mohammadzadeh[a], N. Sahebjamnia[b], A. M. Fathollahi-Fard[b], M. Hahiaghaei-Keshteli[b]

[a] *Department of Mathematics and Computer Science, University of Science and Technology of Mazandaran, Behshahr, Iran*
[b] *Department of Industrial Engineering,University of Science and Technology of Mazandaran, Behshahr, Iran*

PAPER INFO

Paper history:
Received 21 January 2018
Received in revised form 03 April 2018
Accepted 18 June 2018

Keywords:
Cross-docking
Truck Scheduling
Red Deer Algorithm
Virus Colony Search
Water Wave Optimization

ABSTRACT

Nowadays, cross-docking is one of the main concepts in supply chain management in which products received to a distribution center by inbound trucks which are directly to lead into outbound trucks with a minimum handling and storage costs as the main cost of a cross-docking system. According to the literature, several metaheuristics and heuristics are attempted to solve this optimization model. In this regard, this study utilizes three recent nature-inspired metaheuristics among the first studies in this area. Red Deer Algorithm (RDA), Virus Colony Search (VCS) and Water Wave Optimization (WWO) are three novel nature-inspired algorithms proposed recently to employ their applications in engineering problems. The used algorithm's parameters were selected by Taguchi method to enhance the efficiency of algorithms. The outputs of the proposed algorithms are assessed with each other in different criteria along with statistical analyses and the results yielded by prior works. The results demonstrate that RDA showed a competitive performance compared with mixed other existing algorithms.

***doi**: 10.5829/ije.2018.31.08b.14*

1. INTRODUCTION[1]

Supply Chain Management (SCM) considers different theories to uniform the flow of products between levels of supply chain from suppliers to customers [1]. According to SCM, several goods and services are received in a suitable time with the lowest cost [2]. This issue is the more efficient issue to improve the quality of supply chain system [3]. In this regard, this study proposes a cross-docking system, which helps the supply chain managers to find the best strategy in a competitive environment [4]. Therefore, supply chain systems are starting to cross-docking which leads significant benefits to reduce the total cost of system including a little or no inventory (just in time), low shipment costs, low space requirement, low transportation costs [5, 6].

The cross-docking is defined as an important topic in management science to make a trade-off between the store-keeping and ware-house management activities [7]. In the literature of cross-docking systems, most of related papers to cross-docking truck scheduling problem were varied form single objective to multi-objective. In this issue, the determining inventory level or minimizing the makespan as the objectives were considered and reported in literature [2-9]. Among the mentioned them, Yu and Egbelu [8] presented one of important articles concerning truck scheduling in cross-docking systems. A mixed integer linear programming was utilized by their work. In order to solve the model, they offered nine algorithms to reach the optimal solutions. Chen and Song [5] also proposed a two-stage heuristic algorithm and compared with Yu and Egbelu [8].

In our work, it should be mentioned that there is not existed enough information about receiving and loading the goods in real industries. In this regard, Larbi *et al.* [10] proposed a special offer in this issue. They added different types of information levels into truck scheduling problem. Like most of related studies [8], they also considered only one door for both loading and

*Corresponding Author Email: n.sahebjamnia@mazust.ac.ir (N. Sahebjamnia)

Please cite this article as: H. Mohammadzadeha, N. Sahebjamniab, A. M. Fathollahi-Farda, M. Hahiaghaei-Keshteli, New Approaches in Metaheuristics to Solve the Truck Scheduling Problem in a Cross-docking, International Journal of Engineering (IJE), IJE TRANSACTIONS B: Applications Vol. 31, No. 8, (August 2018) 1258-1266

unloading process. They introduced two approaches to solve the model.

Regarding the application of truck scheduling problem in a real industry, a few studies have been introduced. Amini and Tavakkoli-Moghaddam [11] developed a mathematical model for truck scheduling problem in a cross-docking system. In their bi-objective model, the trucks confront breakdowns during the service times. Moreover, a numerous researches are to combine the vehicle routing operation with cross-docking as VRPCD which is studied repeatedly in the recent years as a noteworthy problem in truck scheduling problems [12-16]. For instance, Dondo and Cerdá [13] developed a mixed integer linear programming model to determine a mix of routing and scheduling for vegicle fleet, the dock door assignment, the truck docking sequence and the travel time required to move the goods, the assigned stack door all at once.

While, in past two decades, extensive research have focussed on truck scheduling problem; limited literature have focussed on the future lines of this issue [17-20] For instance, Boysen and Fliedner [20] considered a review article for truck scheduling problems, specifically, in which they classified the published papers by considering three aspects: door environments, operational aspect and type of objective functions. In addition, as a recent study, Ladier and Alpan [21] provided a complete review in the issue of cross-docking scheduling problems.

Since the truck scheduling problem was known as an NP-*hard* one, different types of metaheuristics and heuristics were also proposed in this area. Madani-Isfahani, Tavakkoli-Moghaddam and Naderi [22] by developing a mixed integer programming model proposed Simulated Annealing (SA) and Firefly Algorithm (FA) to solve their model. Furthermore, Cota, Gimenez, Araújo, Nogueira, de Souza and Ravetti [23] explored an efficient heuristic to solve the truck scheduling and compared their results with Chen and Song [5]. Finally, in another study, Golshahi-Roudbaneh *et al.* [7] uses different types of metaheuristic such as GA, SA, Keshtel Algorithm (KA) and Stochastic Fractal Search (SFS) and a hybrid of SA and PSO for their model. Also, they proposed two strong heuristics and compared with the related studies. Moreover, they also developed a lower bound for the cross-docking problem for the first time.

According to the illustrated literature and to the best of our knowledge, this study firstly uses three recent nature-inspired metaheuristics named as Red Deer Algorithm (RDA), Virus Colony Search (VCS) and Water Wave Optimization (WWO). The goal of objective function is to optimize the makespan and determining a sequence for both receiving trucks and shipping ones. In this way, the core contributions of this study can be outlined as follows:

- Solving the truck scheduling in cross-docking systems with three novel metaheuristics: RDA, VCS and WWO proposed firstly in this area.
- Discussing about the advantages and disadvantages of employed approaches.
- Comparing the benefits of used metaheuristics results among each other and also with a recent related study.

Since the used mathematical model has been only adopted from literature [2, 7, 8] and due to page limitation, the mathematical model was not provided. Accordingly, the following sections can be organized as follows. Section 2 considers the solution approaches and our presented metaheuristics with their steps in detail. Section 3 performs the experimental computation for comparison among approaches by different criteria. Finally, the last section proposes the conclusion and future remarks for this study.

2. SOLUTION APPROACH

Notably, the main contribution of this study is utilizing three recent metaheuristics and compared the results with the used metaheuristics in the literature. As mentioned earlier, several studies investigated the truck scheduling problem using heuristics and metaheuristics. According to No Free Lunch theory [24], there is no algorithm to solve all optimization problems [25]. This means that there is always possibility that a new metaheuristic shows a better results in a current or new problem [26]. Accordingly, although heuristics give a quick solution, metaheuristics shows a better quality for solutions. In this paper, three recent nature-inspired metaheuristics including Red Deer Algorithm (RDA), Virus Colony Search (VCS) and Water Wave Optimization (WWO) were employed to tackle the considered problem. Regarding the literature, these metaheuristics have not tried yet in this area. The description and details for approaches are provided as following sub-sections.

2. 1. Encoding Scheme Regarding the representation of a solution for performing a metaheuristic solution planning, a procedure is always needed to encode and decode the mathematical formulation in search space of metaheuristics [6, 19, 20]. Here, the encoding plan is reffered to literature [7].

2. 2. Red Deer Algorithm (RDA) The RDA proposed by Fathollahi Fard and Hajiaghaei-Kehsteli [27] is one of the recent metaheuristic inspired by Red Deer's mating. This method simulates the three main characteristics of this animal during the breading season. Roaring, fighting and mating are the three operators in this algorithm to set the search engine [28]. The Scottish Red Deers (RDs) live natively in British

Isles. An amazing behavior of this animal during the beading season motivates a numerous of people to follow and study their behavior [28]. In this special time of year, male RDs roar loudly to attract the females called as hinds. In this case, fighting between two males is unavoidable. Some of the winners, namely, commanders form the harems (a group of hinds). Moreover, the mating behavior of this animal is the basis of the proposed evolutionary algorithm [25].

The steps of RDA can be summarized as follows. Like other metaheuristics, an initial population is generated randomly named as Red Deers (RDs). These solutions are divided into two groups: males RD and hinds. Males are selected as a number of the best solutions. Then, males are divided into two types: commanders and stags. These two groups are to fight each other. Then, commanders form the harems and their territory. Finally, the mating behavior is performed by commanders between hinds in harems and stags with a nearest hinds without considering the territory [26]. In order to see more description, Figure 1 provides a pseudo-code for the employed RDA.

Moreover, in the recent years, novel metaheuristics are designed in which they control the search phases intelligently [27]. In addition, the mating behavior in three ways is done to focus the exploration phase [28].

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 according to this formula: ($male_{new} = \frac{(1+rand)*male+(1-rand)*male}{2}$).
        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: ($new1 = \frac{(com+stag+rand)}{2}, new2 = \frac{(com+stag-rand)}{2}$).
        Update the position of male commander and stag.
    end for
    Form harems: ($V_n = v_n - \max_i\{v_i\}$; $P_n = \left|\frac{V_n}{\sum_{i=1}^{N_{Com}} V_i}\right|$; $N.harem_n = round\{P_n.N_{hind}\}$).
    for each male commander
        Mate male commander with the selected hinds of his harem randomly.
        $new = \frac{com+hind}{2}$;
        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.
        $new = \frac{stag+hind}{2}$;
        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

Figure 1. The pseudo-code of RDA

2. 3. Virus Colony Search (VCS) The VCS is another recent nature-inspired algorithm presented by Dung Lee *et al.* [29]. VCS considers the diffusion and infection strategies in the host cells. Viruses are adopted the host cells in a cell environment. VCS starts with an initial random population like other metaheuristics. In order to start the main steps of algorithm, the initial agents of population are divided into two types. Viruses as the best agents and host cells are the two kinds of population. Then, the diffusion process is started with creating a new random agent. Also, it should be mentioned that each virus infects only one host cell. Moreover, the infection process is ordered by the reproduction of each virus formulated on the destroying its host cell to employ its nutrients. Another characteristic of this event of nature is the response of immune system for the host cell. In this case, only a percentage of the best viruses remain in each generation and the rest of them are evolved so as to survive. The steps of algorithm are organized by a pseudo-code as seen in Figure 2. As mentioned earlier, authors of novel metaheuristics employ the two search phases in an intelligence way. The immune system of host cells by a probabilistic mechanism helps the algorithm to escape from local optimum.

2. 4. Water Wave Optimization (WWO) Zheng [30] offered the Water Wave Optimization (WWO) as a powerful nature - inspired metaheuristic. WWO simulates the shallow water wave models. In WWO, the population maintains a set of solutions, each of which is analogous to a "*wave*" with a height of 'h' and a wavelength of 'γ'. This nature-inspired algorithm utilizes three search operators to find the optimal solutions. According to water wave models, the operators are wave propagation, breaking and refraction process [31]. Propagation operator controls the waves to propagate with a random position exactly only one in around the nearest of the waves. In this regard, if a wave attains a lower depth (the best fitness), it breaks into solitary waves which are formed in the breaking operation. Then, the refraction operation is done to search the other potential areas for the selected waves. In order to explain the steps of algorithm, clearly, a pseudo-code is provided as shown in Figure 3.

```
Initialize random population and set parameters.
Calculate the fitness and sort them in two types: V_pop and H_pop
X*= the best solution.
while (t<maximum number of iteration)
    for each virus
        Do the diffusion process.
        V_pop' = Guassion(X*, δ) + (rand × X* − rand × V_pop);
        Check the boundary.
    end for
    Update V_pop with V_pop'.
    for each host cell
        Do the infection process.
        Create the new virus (V_pop'').
        V_pop'' = X* + H_pop × N(0, C);
        Check the boundary.
        Response of immune system.
        if p<rand
            V_pop'' = V_pop − rand × (H_pop − V_pop);
        else
            V_pop'' = V_pop;
        end if
    end for
    Update V_pop and H_pop.
    Update the X* if there is better solution.
    t=t+1;
endwhile
```

Figure 2. The pseudo-code of VCS

```
Set the parameters.
Initialize a random population P of n waves.
X*=the best solution.
while (t< maximum number of iteration)
    for each x ∈ P
        /*propagate x to a new x'*/
        x'(d) = x(d) + U(−1, 1) × γL(d) ;
        if f(x') is better than f(x)
            if f(x') is better than f(X*)
                /*break x'*/
                x'(d) = x(d) + N(0,1) × βL(d);
                Update X* with x'.
            end if
            Replace x with x'.
        else
            Decrease x.h by one;
            if x.h==0
                /*refract x to a new x'*/
                x'(d) = N((X*(d)+x(d))/2, |X*(d)−x(d)|/2);
                γ' = γ × f(x)/f(x');
            end if
        end if
    end for
    /*update the wavelengths*/
    γ = γ × ∝^(−((f(x)−fmin+ε)/(fmax−fmin+ε))
    t=t+1;
end while
return X*
```

Figure 3. The pseudo-code of WWO

In the WWO, propagation operator is a local search while the breaking operation increases the exploitation properties by generating random solitary waves around the current best solution. Furthermore, refraction operator motivates the algorithm to investigate the search space intelligently for finding any other best solutions to escape from the local optima.

3. COMPUTATIONAL EXPERIMENTS

In this section, firstly, the instances are considered by benchmarked test problems. Then, presented metaheuristics are tuned by Taguchi method. Finally, the algorithms were compared with each other and the best results existing in the literature. It should be noted, the codes of algorithms were written by Microsoft Visual Studio 2014 in C^{++} language. Also, a computer with 1.7GB CPU and 6.0GB RAM was used in our study.

3. 1. Instances First of all, 20 small test problems (1 to 20) generated by Yu [8] have been utilized to check the validation of metaheuristics. Furthermore, to evaluate the metaheuristics in the large sizes, the benchmarked test problems benchmarked by Golshahi-Roudbaneh *et al.* [7] in 15 instances [8, 21-34] were utilized. Due to page limitation of journal, the details of test problems have not been reported. Interested readers are referred to see Golshahi-Roudbaneh *et al.* [7] and Yu [8] for more information.

3. 2. Parameter Setting In this section, the algorithm's parameters are tuned by Taguchi method. If the parameters are not chosen properly, the behavior of algorithm would be inefficient [26]. Genichi Taguchi [34] designed some orthogonal arrays to reduce the number of experiments. This approach was also used in several recent papers to set the algorithm's parameters [30-33]. In addition, the used parameters of presented metaheuristics are given as listed in Table 1. For each factor (parameter), maximum four levels were considered in the provided algorithms' parameters tuning. As a result, according to the mentioned experiments, Table 2 shows the best-selected level for the parameters.

3. 3. Comparison of Metaheuristics In this section, first of all, metaheuristics have been validated by 20 benchmarked instances. From Table 3, the average outputs of algorithms during thirty run times and discrepancy from the optimal value of exact solver found by Yu and Egbelu [8] is given in the table. The presented metaheuristics can reach an optimal value in a reasonable time. Note that there is a little difference between computational times of algorithms. According to the gap of metaheuristics, in small sizes, RDA is slightly better than other algorithms. Moreover, the three presented metaheuristics are employed to solve the 15 benchmark test problems in the large sizes. Note that the exact algorithm cannot find a solution for the high dimensional of problem. So, only metaheuristics are recommended to reach an optimal solution.

TABLE 1. The metaheuristics algorithms parameters and their levels

Algorithm	Parameter	Levels			
		1	2	3	4
RDA	A: Maximum iteration (*Maxit*)	100	200	300	-
	B: Population size (n_{pop})	100	200	300	-
	C: Number of males (*Nmale*)	15	25	40	-
	D: Percentage of commanders (*c*)	0.6	0.8	0.9	-
	E: Percentage of inside the harems (*a*)	0.5	0.7	0.9	-
	F: Percentage of outside the harems (*b*)	0.4	0.6	0.7	-
VCS	A: Maximum iteration (*Maxit*)	200	400	600	800
	B: Population size (n_{pop})	100	150	200	250
	C: input variable of search function (*l*)	0.02	0.05	0.08	0.1
WWO	A: Maximum iteration (*Maxit*)	300	500	600	-
	B: Population size (n_{pop})	100	200	250	-
	C: Wavelength reduction coefficient (*a*)	1.001	1.01	1.1	-
	D: Breaking coefficient (*b*)	0.001	0.01	0.1	-
	E: Maximum wave height (h_{max})	5	6	-	-

TABLE 2. The best level of parameters for each algorithm

Algorithm	Parameter	Best level
RDA	Maximum iteration (*Maxit*)	300
	Population size (n_{pop})	300
	Number of males (*Nmale*)	25
	Percentage of commanders (*c*)	0.9
	Percentage of inside the harems (*a*)	0.9
	Percentage of outside the harems (*b*)	0.4
VNS	Maximum iteration (*Maxit*)	800
	Population size (n_{pop})	250
	Input variable of search function (*l*)	0.02
WWO	Maximum iteration (*Maxit*)	600
	Population size (n_{pop})	200
	Wavelength reduction coefficient (*a*)	1.01
	Breaking coefficient (*b*)	0.01
	Maximum wave height (h_{max})	6

In order to be reliable, each algorithm is run for thirty times. The best, the worst and the average of outputs of algorithms were saved. In addition, the standard deviation among thirty outputs is computed. In addition, the average of computational time for algorithms is also noted. Furthermore, the hitting time, the first time that the best solution ever found is also considered to compare the metaheuristics. All mentioned outputs are given in Table 4. In addition, the behavior of algorithms in term of solution time and hitting time is illustrated in Figures 4 and 5, respectively. As can be seen, RDA needs more time in most of test problems. However, in the term of hitting time in some test problems RDA shows a less value in this term. Furthermore, the result of this study in large-sizes is compared with Golshahi-Roudbaneh et al. [7] as resulted in Table 5. Regarding the results, our presented metaheuristics reach a better bound for the most of benchmark test problems. Additionally, in most of them, RDA finds the better best cost value and shows a strong behaviour between two other ones.

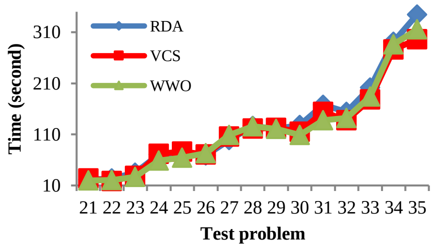

Figure 4. The behavior of algorithms in term of solution time

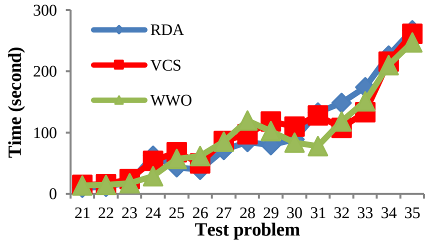

Figure. 5. The behavior of algorithms in term of hitting time

TABLE 3. The results of metaheuristics in small sizes (CP=computational time (second), M=the average of solutions, Gap=the deviation from the best solution=$(Z_{Alg}-Z_{best})/Z_{best}$)

Instances	Optimal value found by Yu and Egbelu [8]	RDA			VCS			WWO		
		M	CP	Gap	M	CP	Gap	M	CP	Gap
1	1557	1650.4	15.78	0.06	1634.8	13.68	0.05	1681.5	16.29	0.08
2	1577	1687	14.25	0.07	1671.6	12.45	0.06	1687.3	15.77	0.07
3	1372	1440.6	12.64	0.05	1454.3	12.01	0.06	1454.3	13.65	0.06
4	1749	1888.9	18.33	0.08	1871.4	16.79	0.07	1923.9	17.25	0.1
5	1579	1657.9	14.82	0.05	1673.7	11.55	0.06	1736.9	15.13	0.1
6	1546	1592.3	16.25	0.03	1685.1	16.09	0.09	1654.2	16.94	0.07
7	1535	1611.7	15.27	0.05	1673.1	14.91	0.09	1657.8	15.52	0.08
8	1525	1601.2	15.66	0.05	1616.5	14.78	0.06	1600	16.34	0.05
9	1473	1605.5	14.37	0.09	1605.5	15.53	0.09	1561.3	16.88	0.06
10	1452	1597.2	15.83	0.1	1568.1	14.89	0.08	1568	15.92	0.08
11	2232	2410.5	18.02	0.08	2455.2	17.32	0.1	2388.2	19.06	0.07
12	2833	3087.9	19.21	0.09	3087.9	17.89	0.09	3059.6	18.86	0.08
13	2386	2481.4	18.58	0.04	2576.8	18.12	0.08	2505.3	19.53	0.05
14	2385	2528.1	18.73	0.06	2599.6	16.97	0.09	2551.9	19.22	0.07
15	2745	2937.1	19.59	0.07	2964.6	19.64	0.08	2964.6	20.86	0.08
16	2407	2527.3	17.81	0.05	2623.6	16.52	0.09	2551.4	17.97	0.06
17	1867	2035	16.44	0.09	1997.6	17.33	0.07	2035	18.44	0.09
18	2502	2577	19.53	0.03	2727.1	18.92	0.09	2677.1	19.29	0.07
19	2553	2706.1	20.75	0.06	2757.2	19.36	0.08	2782.7	19.88	0.09
20	2732	2868.6	18.89	0.05	2950.5	17.41	0.08	2977.8	19.37	0.09
Average				0.0625			0.078			0.075

TABLE 4. Comparison of presented metaheuristics with different criteria in large instances (B=the best solution, W=the worst solution, SD=standard deviation, HT=hitting time (second))

Set		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
RDA	B	3012	3505	5026	3922	5076	7734	7014	7246	6131	5516	8182	7952	9146	10644	9513
	M	3179.6	3652	5231.8	4086	5207	8129.8	7220.6	7418	6399.47	5770.476	8565.57	8087.81	9595.85	11033.05	9773.952
	W	3490	4104	5836	4608	5605	9145	7954	8054	7253	6519	9486	8723	10539	11829	10463
	SD	184.97	191.6	281.84	206.12	169.4	574.46	292.15	252.01	376.71	370.902	475.98	223.04	460.34	431.40	350.7109
	CP	21.26	23.86	34.79	70.58	72.46	68.53	98.67	126.47	120.82	128.49	167.53	154.18	201.34	291.23	344.59
	HT	9.85	11.53	18.67	62.45	43.71	39.03	71.82	84.91	79.74	89.15	132.72	148.29	173.83	225.91	267.18
VCS	B	3046	3525	5082	4062	5104	7688	7014	7348	6246	5508	8224	8024	9253	10467	9429
	M	3268.6	3743	5263	4239.2	5341	8203.3	7336.8	7673.9	6502.23	5913.76	8587.429	8325.9	9630.667	10857.57	9678.143
	W	3562	4086	5674	4708	5688	9243	7834	8054	7067	6423	9145	9245	10483	11538	10574
	SD	212.93	184.1	203.21	210.65	214	546	322.82	232.67	317.59	372.96	326.7396	413.63	465.7958	373.8717	335.9331
	CP	23.76	18.93	28.75	71.62	76.54	70.85	105.48	121.65	122.75	114.93	153.81	137.86	178.65	275.89	296.05
	HT	14.65	15.73	24.16	53.85	68.13	49.76	86.35	97.14	118.23	109.78	128.05	107.54	133.29	215.54	261.28

WWO	B	3012	3466	5048	3968	5076	7709	6950	7314	6218	5516	8209	8016	9204	10587	9429
	M	3077.4	3720	5249.2	4682	5292	8282.6	7241.8	7673.0	6633.28	5878.61	8583.76	8220.71	9597.76	10893.38	9729.381
	W	3428	4145	5736	4166	5739	9145	7834	8164	7134	6423	9516	8859	10386	11296	10487
	SD	125.08	268.1	220.81	222.45	234.5	622.35	327.39	232.67	320.37	386.97	446.98	311.89	464.87	280.943	344.2513
	CP	19.76	20.6	26.54	58.65	64.2	72.96	108.69	125.74	120.7	108.68	137.64	142.35	184.27	286.19	315.67
	HT	13.76	14.8	17.49	28.92	57.1	61.83	85.32	119.75	102.51	83.67	77.98	118.26	150.83	209.83	247.13

TABLE 5. Metaheuristics compound solutions in our study compared with a related study (the minimum output found by algorithms)

Set	**Golshahi-Roudbaneh *et al.* [7]**		**This study**	
21	3046	Found by SA-PSO	3012	Found by RDA & WWO
22	3505	Found by SA & SFS	3466	Found by WWO
23	5026	Found by SFS	5026	Found by RDA
24	3826	Found by SFS	3922	Found by RDA
25	5161	Found by SA-PSO	5076	Found by RDA & WWO
26	7799	Found by KA	7688	Found by VCS
27	6950	Found by KA	6950	Found by WWO
28	7484	Found by SFS	7246	Found by RDA
29	6131	Found by SFS	6131	Found by RDA
30	5472	Found by SA-PSO	5508	Found by VCS
31	8327	Found by SA	8182	Found by RDA
32	8166	Found by SA-PSO	7952	Found by RDA
33	9300	Found by SA	9146	Found by RDA
34	10758	Found by SFS	10467	Found by VCS
35	9338	Found by SA & SA-PSO	9429	Found by VCS & WWO

Eventually, to confirm statistically the algorithms' results, an analysis of variance (ANOVA) is employed to compare the performance of algorithms. The results are formulated on the RPD among thirty run times. Figure 6 shows the interval plot at 95% confidence level for three presented metaheuristics. According to the figure, it is evident that results of VCS are more reliable than other algorithms and it has a lower tolerance among thirty run times.

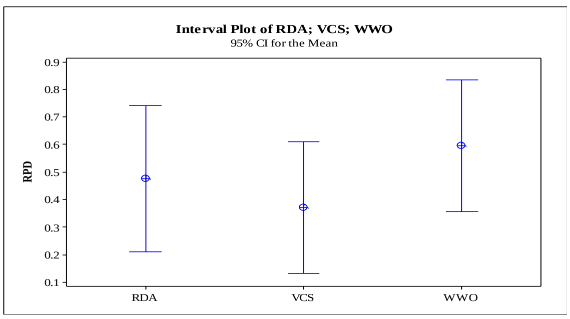

Figure 6. Means plot and LSD intervals for the presented algorithms

4. CONCLUSION

In this study, a famous truck scheduling problem has been solved by three recent metaheuristics which had not being used in previous works, namely, RDA, VCS and WWO. First of all, metaheuristics are validated by conventional optimization method in 20 small instances benchmarked by the literature. In addition, 15 benchmark instances in large sizes were examined by proposed methodologies and compared with the best existing results in this research area. The algorithms were evaluated in different criteria e.g. solution time and quality, hitting time, convergence and statistical analyses. Finally, results confirm RDA is more successful than others. However, it needs more time by increasing the size of problem. To achieve the managerial implication of this study, it is inclined to believe that managers usually need an efficient and quick solution approach to decide a proper decision. So, this study by proposing three novel metaheuristics showed that RDA is one of most effective and efficiency solution methods in the literature to provide a quick and performance answer for the decision-makers of cross-docking centers.

For future works, more analyses on the results of this paper can be suggested to be investigated. Additionally, the presented metaheuristics can be applied to the other real scale optimization problem in cross-docking systems with supposition assumptions such as multi-door or multi-objective truck scheduling problems. The last but not the least is suggesting new heuristics and metaheuristics to solve the model which can be ordered to continue the line of this study.

5. REFERENCES

1. Sadeghi-Moghaddam, S., Hajiaghaei-Keshteli, M. and Mahmoodjanloo, M., "New approaches in metaheuristics to solve the fixed charge transportation problem in a fuzzy environment", ***Neural Computing and Applications***, (2017), 1-21.
2. Nourmohamadi Shalke, P., Paydar, M.M. and Hajiaghaei-Keshteli, M., "Sustainable supplier selection and order allocation through quantity discounts", ***International Journal of Management Science and Engineering Management***, Vol. 13, No. 1, (2018), 20-32.
3. Fard, A.M.F., Gholian-Jouybari, F., Paydar, M.M. and Hajiaghaei-Keshteli, M., "A bi-objective stochastic closed-loop supply chain network design problem considering downside risk", ***Industrial Engineering & Management Systems***, Vol. 16, No. 3, (2017), 342-362.
4. Fard, A.M.F. and Hajiaghaei-Keshteli, M., "A bi-objective partial interdiction problem considering different defensive systems with capacity expansion of facilities under imminent attacks", ***Applied Soft Computing***, Vol. 68, (2018), 343-359.
5. Chen, F. and Song, K., "Minimizing makespan in two-stage hybrid cross docking scheduling problem", ***Computers & Operations Research***, Vol. 36, No. 6, (2009), 2066-2073.
6. Wisittipanich, W. and Hengmeechai, P., "Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization", ***Computers & Industrial Engineering***, Vol. 113, (2017), 793-802.
7. Golshahi-Roudbaneh, A., Hajiaghaei-Keshteli, M. and Paydar, M.M., "Developing a lower bound and strong heuristics for a truck scheduling problem in a cross-docking center", ***Knowledge-Based Systems***, Vol. 129, (2017), 17-38.
8. Yu, W. and Egbelu, P.J., "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage", ***European Journal of Operational Research***, Vol. 184, No. 1, (2008), 377-396.
9. Bellanger, A., Hanafi, S. and Wilbaut, C., "Three-stage hybrid-flowshop model for cross-docking", ***Computers & Operations Research***, Vol. 40, No. 4, (2013), 1109-1121.
10. Larbi, R., Alpan, G., Baptiste, P. and Penz, B., "Scheduling cross docking operations under full, partial and no information on inbound arrivals", ***Computers & Operations Research***, Vol. 38, No. 6, (2011), 889-900.
11. Amini, A. and Tavakkoli-Moghaddam, R., "A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks", ***Computers & Industrial Engineering***, Vol. 96, (2016), 180-191.
12. Dondo, R. and Cerdá, J., "A sweep-heuristic based formulation for the vehicle routing problem with cross-docking", ***Computers & Chemical Engineering***, Vol. 48, (2013), 293-311.
13. Dondo, R. and Cerdá, J., "The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system", ***Computers & Chemical Engineering***, Vol. 76, (2015), 42-62.
14. Morais, V.W., Mateus, G.R. and Noronha, T.F., "Iterated local search heuristics for the vehicle routing problem with cross-docking", ***Expert Systems with Applications***, Vol. 41, No. 16, (2014), 7495-7506.
15. Yin, P.-Y., Lyu, S.-R. and Chuang, Y.-L., "Cooperative coevolutionary approach for integrated vehicle routing and scheduling using cross-dock buffering", ***Engineering Applications of Artificial Intelligence***, Vol. 52, (2016), 40-53.
16. Grangier, P., Gendreau, M., Lehuédé, F. and Rousseau, L.-M., "A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking", ***Computers & Operations Research***, Vol. 84, (2017), 116-126.
17. Stephan, K., Boysen, N. and Cross-docking, J., "Manage", ***Control***, Vol. 22, (2011), 129-137.
18. Van Belle, J., Valckenaers, P. and Cattrysse, D., "Cross-docking: State of the art", ***Omega***, Vol. 40, No. 6, (2012), 827-846.
19. Zuluaga, J.P.S., Thiell, M. and Perales, R.C., "Reverse cross-docking", ***Omega***, Vol. 66, (2017), 48-57.
20. Boysen, N. and Fliedner, M., "Cross dock scheduling: Classification, literature review and research agenda", ***Omega***, Vol. 38, No. 6, (2010), 413-422.
21. Ladier, A.-L. and Alpan, G., "Cross-docking operations: Current research versus industry practice", ***Omega***, Vol. 62, (2016), 145-162.
22. Madani-Isfahani, M., Tavakkoli-Moghaddam, R. and Naderi, B., "Multiple cross-docks scheduling using two meta-heuristic algorithms", ***Computers & Industrial Engineering***, Vol. 74, (2014), 129-138.
23. Cota, P.M., Gimenez, B.M., Araújo, D.P., Nogueira, T.H., de Souza, M.C. and Ravetti, M.G., "Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre", ***Computers & Industrial Engineering***, Vol. 95, (2016), 135-143.
24. Hajiaghaei-Keshteli, M. and Sajadifar, S.M., "Deriving the cost function for a class of three-echelon inventory system with n-retailers and one-for-one ordering policy", ***The International Journal of Advanced Manufacturing Technology***, Vol. 50, No. 1-4, (2010), 343-351.
25. Hajiaghaei-Keshteli, M. and Fard, A.M.F., "Sustainable closed-loop supply chain network design with discount supposition", ***Neural Computing and Applications***, (2018), 1-35.
26. Samadi, A., Mehranfar, N., Fathollahi Fard, A. and Hajiaghaei-Keshteli, M., "Heuristic-based metaheuristics to address a sustainable supply chain network design problem", ***Journal of Industrial and Production Engineering***, Vol. 35, No. 2, (2018), 102-117.
27. Fard, A.F. and Hajiaghaei-Keshteli, M., "Red deer algorithm (rda); a new optimization algorithm inspired by red deers' mating", in International Conference on Industrial Engineering, IEEE., (2016), 33-34.
28. Golmohamadi, S., Tavakkoli-Moghaddam, R. and Hajiaghaei-Keshteli, M., "Solving a fuzzy fixed charge solid transportation problem using batch transferring by new approaches in meta-heuristic", ***Electronic Notes in Discrete Mathematics***, Vol. 58, (2017), 143-150.
29. Li, M.D., Zhao, H., Weng, X.W. and Han, T., "A novel nature-inspired algorithm for optimization: Virus colony search", ***Advances in Engineering Software***, Vol. 92, (2016), 65-88.

30. Zheng, Y.-J., "Water wave optimization: A new nature-inspired metaheuristic", ***Computers & Operations Research***, Vol. 55, (2015), 1-11.
31. Fathollahi-Fard, A.M. and Hajiaghaei-Keshteli, M., "A stochastic multi-objective model for a closed-loop supply chain with environmental considerations", ***Applied Soft Computing***, Vol. 69, (2018), 232-249.
32. Fard, A.M.F. and Hajaghaei-Keshteli, M., "A tri-level location-allocation model for forward/reverse supply chain", ***Applied Soft Computing***, Vol. 62, (2018), 328-346.
33. Nasiri, E., Afshari, A. and Hajiaghaei-Keshteli, M., "Addressing the freight consolidation and containerization problem by recent and hybridized meta-heuristic algorithms", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 30, No. 3, (2017), 403-410.
34. Taguchi, G. and Clausing, D., "Robust quality", ***Harvard Business Review***, Vol. 68, No. 1, (1990), 65-75.

New Approaches in Metaheuristics to Solve the Truck Scheduling Problem in a Cross-docking Center

H. Mohammadzadeh[a], N. Sahebjamnia[b], A. M. Fathollahi-Fard[b], M. Hahiaghaei-Keshteli[b]

[a] *Department of Mathematics and Computer Science, University of Science and Technology of Mazandaran, Behshahr, Iran*
[b] *Department of Industrial Engineering,University of Science and Technology of Mazandaran, Behshahr, Iran*

PAPER INFO

Paper history:
Received 21 January 2017
Received in revised form 03 April 2018
Accepted 18 June 2018

Keywords:
Cross-docking
Truck Scheduling
Red Deer Algorithm
Virus Colony Search
Water Wave Optimization

چکیده

امروزه، مراکز اتصال متقابل یکی از موضوعات اصلی در مدیریت زنجیره تامین است که به دریافت محصولات در مراکز توزیع با استفاده از فرابارانداز کامیون ها که مستقیما به فرابارانداز کامیون دیگر با حداقل هزینه حمل و نگهداری که جزء هزینه های اصلی یک مرکز اتصال متقابل است. با توجه به مرور ادبیات این حوزه، الگوریتم های فراابتکاری و ابتکاری متعددی در حل این مساله بهینه سازی به کار گرفته شده است. در این رابطه، این مطالعه سه الگوریتم جدید الهام گرفته از طبیعت را در میان اولین مطالعات بهره برداری میکند. الگوریتم گوزن سرخ، الگوریتم جست و جوی جمعی ویروس و بهینه سازی موج آب، سه الگوریتم جدید الهام گرفته از طبیعت هستند که به تازگی برای استفاده از کاربرد آنها در علوم مهندسی بهره برداری شده است. پارامترهای استفاده شده در الگوریتم ها با استفاده از روش تاگوچی انتخاب شده اند تا اثربخشی الگوریتم ها افزایش یابد. خروجی های الگوریتم ها با یکدیگر در معیارهای مختلف و آنالیزهای آماری ارزیابی شده و نتایج به دست آمده با کارهای گذشته نیز مقایسه شده است. نتایج بیانگر آن است که الگوریتم گوزن سرخ یک کارایی مختلط را نشان می دهد و از دیگر روش های پیشنهاد شده در این نوع مساله بهینه سازی قوی تر بوده است.

doi*: 10.5829/ije.2018.31.08b.14*