Published in "Computers & Industrial Engineering", 2019, vol. 137, article no 106103, pp. 1-15 which should be cited to refer to this work.
DOI: 10.1016/j.cie.2019.106103

Novel modifications of social engineering optimizer to solve a truck scheduling problem in a cross-docking system

Amir Mohammad Fathollahi-Fard[a,*], Mehdi Ranjbar-Bourani[b], Naoufel Cheikhrouhou[c], Mostafa Hajiaghaei-Keshteli[b]

[a]*Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran, Iran*
[b]*Department of Industrial Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran*
[c]*Geneva School of Business Administration, University of Applied Sciences Western Switzerland (HES-SO), 1227 Carouge, Switzerland*

Abstract

The truck scheduling problem is one of the most challenging and important types of scheduling with a large number of real-world applications in the area of logistics and cross-docking systems. This problem is formulated to find an optimal condition for both receiving and shipping trucks sequences. Due to the difficulty of the practicality of the truck scheduling problem for large-scale cases, the literature has shown that there is a chance, even with low possibility, for a new optimizer to outperform existing algorithms for this optimization problem. The paper introduces modified versions of the Social Engineering Optimizer (SEO), an algorithm inspired by social engineering phenomena to solve the truck scheduling problem. To validate these optimizers, they are evaluated by solving a set of standard benchmark functions. All the algorithms have been calibrated by the Taguchi experimental design approach to further enhance their optimization performance. In addition to some truck scheduling benchmarks , a real case study is addressed to show the high-efficiency of the developed optimizers in a real situation. The results indicate that the proposed modifications of SEO considerably outperform the state of the art algorithms for the truck scheduling problem.

Keywords: truck scheduling problem, cross-docking system, Social Engineering Optimizer (SEO), benchmark functions.

* Corresponding author: amirfard@aut.ac.ir (A. M. Fathollahi-Fard)

Notations and nomenclature:

Indices

i,j	Index of trucks
k	Index of products

Parameters

D	The truck changeover time
r_{ik}	The number of units of product type k that was initially loaded in receiving truck i
s_{jk}	The number of units of product type k that was initially needed for shipping truck j
V	The moving time of products from the receiving dock to the shipping dock
M	The big number
R	The number of the receiving trucks
S	The number of the shipping trucks
N	The number of product types

Variables

T	Makespan
d_j	Time at which shipping truck j enters the shipping dock
L_j	Time at which shipping truck j leaves the shipping dock
c_i	Time at which receiving truck i enters the receiving dock
F_i	Time at which receiving truck i leaves the receiving dock
X_{ijk}	The number of units of product type k that is transferred from receiving truck i to shipping truck j

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck i to shipping truck j;} \\ 0, & \text{Otherwise;} \end{cases}$$

$$p_{ij} = \begin{cases} 1, & \text{If receving truck i preceeds receiving truck j in the receiving truck sequence;} \\ 0, & \text{Otherwise;} \end{cases}$$

$$q_{ij} = \begin{cases} 1, & \text{If shipping truck i preceeds shipping truck j in the shipping truck sequence;} \\ 0, & \text{Otherwise.} \end{cases}$$

Abbreviations

SA	Simulated Annealing
GA	Genetic Algorithm
KA	Keshtel Algorithm
RDA	Red Deer Algorithm
SEO	Social Engineering Optimizer
SFS	Stochastic Fractal Search
WWO	Water Wave Optimization
VCS	Virus Colony Search
PSO	Particle Swarm Optimization
NFL	No Free Lunch
GWO	Grey Wolf Optimizer
ABC	Artificial Bee Colony
ICA	Imperialist Competitive Algorithm
FA	Firefly Algorithm
L-SHADE	Linear-Success-History based on Adaptive of Differential Evolution

1-Introduction and literature review

Nowadays, quick changes in today's competitive markets highlight that the satisfaction of customers has become crucially important for companies in logistics and cross-docking systems (Mohtashami et al., 2015). The significance of being responsive has led many companies to modify their logistics and cross-docking systems by taking logistics principles into application (Zuluaga et al., 2017). To improve the accessibility of logistics facilities, truck scheduling plays a key role to preserve the supply chain objectives aiming at making a better trade-off between the total cost and customers' expectations (Golshahi-Roudbaneh et al., 2017). Therefore, to address this challenging and important type of scheduling problem, new optimizers need to be developed (Golshahi-Roudbaneh et al., 2017). This motivates our attempt to introduce a set of novel modifications for a recently-developed optimizer to better solve the truck scheduling problem for a cross-docking system.

Generally, cross-docking aims to manage the flow of products so that no storage of inventory occurs for more than twenty-four hours (Madani-Isfahani et al., 2014; Fathollahi Fard and Hajiaghaei-Keshteli, 2018). The goods are directly transferred to the outbound dock in order to be loaded into shipping trucks. In such systems, long-term storage is not allowed. Hence, the cross-docking system is beneficial to improve the physical flow of products through the supply chain in an efficient way (Hajiaghaei-Keshteli and Fathollahi-Fard, 2018). More recent reviews and advances about the supply chain systems can be referred to Sayyadi and Awasthi (2018a) and (2018b); Giri and Bardhan (2014); Giri and Masanta (2018); Sarkar and Giri (2018); Hao et al., (2018); Rabbani et al., (2018) and (2019); Gharaei et al., (2019), (2019a), (2019b), (2019c); Awasthi and Omrani (2019); Tsao, (2015); Hoseini-Shekarabi et al., (2019); Dubey et al., (2015); Duan et al., (2018); Kazemi et al., (2018); Yin et al., (2016); Shah et al., (2018).

The literature of the cross-docking can be divided into several scopes such as location of cross-docks, layout design, vehicle routing, truck scheduling, dock door assignment and supply chain networks among others (Hajiaghaei-Keshteli and Fathollahi-Fard, 2018; Samadi et al., 2018; Zuluaga et al., 2017). A survey done by Ladier and Alpan (2016) reviewed cross-docking operations and categorized them into five groups including truck to door sequencing, truck to door assignment, truck to door scheduling and truck sequencing and scheduling. The authors also indicated that most of studies mainly focus on minimization of the makespan (total operation time) and traveled distance. Subsequently, they suggested that developing more heuristics and metaheuristics are still needed to be explored due to the complexity of the mathematical models and the need for adopting quick decisions by managers.

The literature of truck scheduling is very rich from both aspects of modeling and solution methodologies. From the modeling perspective, one of the seminal research works was conducted by Yu (2002) who studied the truck scheduling problem with the aim of determining the selected sequence for both shipping and receiving trucks and optimizing the makespan. From the solution methodology standpoint, Yu and Egbelu (2008) presented a mathematical model for a truck scheduling problem in which they consider a receiving door, a shipping door and a temporary storage in front of the shipping door. They suggested nine heuristic methods to solve their proposed model, then compared them with the exact results obtained from the complete enumeration method. Chen and Lee (2009) modeled the truck scheduling problem as a flow-shop machine scheduling problem. They solved the problem using the branch-and-bound algorithm. According to the authors, this algorithm can find the optimal solution of the problems up to 60 jobs in a reasonable amount of time. Boysen (2010) studied a cross dock scheduling problem in a storage ban mode. The objectives of the model were to minimize the processing time, the flow time and the tardiness of outbound trucks. He used dynamic programming and Simulated Annealing (SA) to solve the proposed mathematical model.

In particular, there is an increased interest in truck scheduling problem in cross-docking systems. The focus is to develop new solution approaches to find an efficient result by using heuristics and metaheuristics. Konur et al., (2013) considered the cross docking scheduling problem as a two-phase parallel machine problem with earliness and tardiness. In their research, a new metaheuristic based on Genetic Algorithm (GA) was utilized. Subsequently, Amini et al. (2014) developed a mathematical model for truck scheduling problem considering arrival times for inbound trucks and adding the learning effect for unloading and loading process for the first time. Due to the complexity of the model in large-scale instances, a Particle Swarm Optimization (PSO) was applied to solve their problem. Amini and Tavakkoli-Moghaddam (2016) formulated a problem in which truck availability faces reductions during the times of services. They also considered a due date for each shipping truck and used three multi-objective meta-heuristic algorithms to solve the problem. Golshahi-Roudbaneh *et al.* (2017) proposed efficient heuristics and metaheuristics to reach the optimal value for both receiving and shipping trucks sequences, based on Yu (2002). They applied Stochastic Fractal Search (SFS) and Keshtel Algorithm (KA). Mohammadzadeh et al., (2018) proposed three novel metaheuristics including Red Deer Algorithm (RDA), Water Wave Optimization (WWO) and Virus Colony Search (VCS) to solve the truck scheduling problem based on Golshahi-Roudbaneh et al. (2017). Ye et al. (2018) proposed an improved PSO to solve a truck scheduling problem considering the products loading and unloading constraints. They showed the efficiency of the proposed metaheuristic against Genetic Algorithm (GA), Particle Swarm Optimisation (PSO) and another variation of PSO.

Peng and Zhou (2019) proposed a hybrid Grey Wolf Optimizer (GWO) to address a bi-objective truck scheduling problem in automotive industry environment. Tadumadze et al., (2019) developed an integrated truck and workforce scheduling problem for unloading trucks and proposed a set of heuristics to solve it. Yi et al., (2019) proposed a scheduling appointment system for container truck arrivals considering their effects on congestion and solved this complex problem by PSO.

The exist many formulations for the scheduling problem with different assumptions and scopes depending on the application under investigation. Due to the intricate nature of these problems, most of the relevant studies applied different optimizers including heuristics and metaheuristics to find robust solutions (Hlal et al., 2019; Pourdaryaei et al., 2019; Abbasi et al., 2019). This fact is based on the No Free Lunch (NFL) theory which states that there is no optimizer to solve all optimization problems (Wolpert and Macready, 1997). According to the literature, for a modified optimizer there will always be a chance, even with low probability, that it can outperform existing algorithms for a particular problem at hand (Kaboli et al., 2016, 2016a, 2017a; Kaboli and Alqallaf, 2019). To the best of our knowledge, the SEO proposed by Fathollahi-Fard *et al.,* (2018) has not been employed in the field of truck scheduling problems. Although the authors of this research have previously worked on developing different novel hybridizations and modifications of other recently-developed algorithms e.g., WWO (Fathollahi Fard and Hajiaghaei-Keshteli, 2018), RDA (Samadi et al., 2018; Fathollahi-Fard et al., 2018a), KA (Hajiaghaei-Keshteli and Fathollahi-Fard, 2018; Fathollahi-Fard et al., 2018a and 2018c), VCS (Fathollahi-Fard and Hajiaghaei-Keshteli, 2018), Salp Swarm Algorithm (Fathollahi-Fard et al., 2018b), SFS (Fathollahi-Fard et al., 2018c) etc., this is the first attempt to offer novel versions of SEO in comparison with the previous works. In this study, the proposed modifications are compared with the original version of SEO and to existing efficient algorithms in the literature. Overall, the core innovations of this paper are as follows:

- This is the first attempt to apply SEO to solve the truck scheduling problem;
- Some novel modifications of SEO are proposed;
- A set of standard benchmark functions for the assessment of novel optimizers are employed for the evaluations of SEO;
- The modified algorithms are evaluated using the benchmark problems for the truck scheduling problem;
- A real case study to approve the proposed truck scheduling problem is conducted and the results confirm the effectiveness and efficiency of the proposed modifications.

The rest of this paper is organized as follows. Section 2 explains the problem description and its mathematical formulation for the truck scheduling problem. In Section 3, the proposed solution approach is explained in detail. An extensive comparison and evaluation of the proposed modifications are provided in Section 4. Finally, in the last section, the results are discussed and suggestions along with future directions for further research are elaborated.

2-Problem description and mathematical formulation

The section describes a formulation for the truck scheduling problem in a cross-docking system. To define the general idea of problem in the real domain, consider a cross dock with I-shaped structure as represented in Figure 1. All activities through this system is automated. As a one-touch cross-docking system, temporary storage is not allowed. The inbound trucks sent from suppliers are unloaded and the outbound trucks are loaded to submit the products based on their orders. The products can be directly transferred from strip doors to stack ones. Therefore, the doors of the considered cross-dock are assumed to be in an exclusive mode, meaning that they are not considered as decision variables. Another main characteristic of this system is that the interruption for unloading and loading of trucks is not allowed. The arrival pattern of trucks is concentrated so that all trucks are available at time zero. Consequently, the departure pattern of the trucks is planned with no restriction and no penalty for the postponement of trucks. The cross-dock can be classified as a pre-distribution center. Accordingly, the interchangeability of products is not allowed. The proposed truck scheduling problem aims to determine the optimal sequence of receiving and shipping trucks with minimizing the makespan.

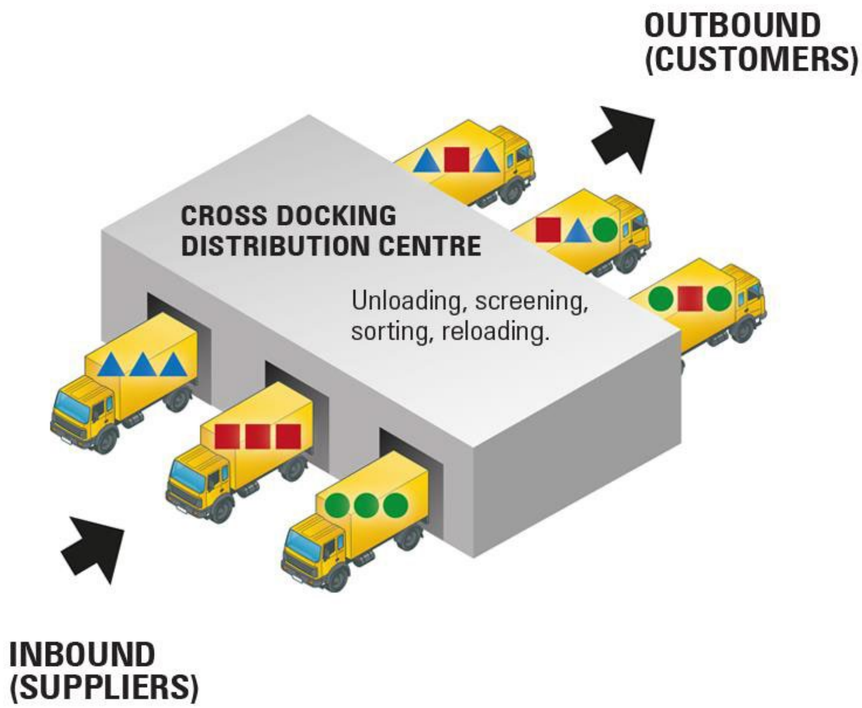


Figure 1. Graphical illustration of proposed cross-dock Yu and Egbelu (2008)

A mathematical model is formulated based on the previous studies e.g., Yu (2002), Yu and Egbelu (2008), Golshahi-Roudbaneh et al., (2017) and Mohammadzadeh et al., (2018). The major model assumptions of the problem under study are listed as follows:

- The time for loading and unloading of products for all trucks is the same.
- This time is the same for one unit of time for each product.
- At time zero, all shipping and receiving trucks are available.
- For all trucks, the changeover time is the same.
- The location of temporary storage is in front of the shipping dock and its capacity is infinite.

The mathematical model investigated in this study is based on the following notations:

Indices

i,j	Index of trucks
k	Index of products

Parameters

D	The truck changeover time
r_{ik}	The number of units of product type k that was initially loaded in receiving truck i
s_{jk}	The number of units of product type k that was initially needed for shipping truck j
V	The moving time of products from the receiving dock to the shipping dock
M	The big number
R	The number of the receiving trucks
S	The number of the shipping trucks
N	The number of product types

Variables

T	Makespan
d_j	Time at which shipping truck j enters the shipping dock
L_j	Time at which shipping truck j leaves the shipping dock
c_i	Time at which receiving truck i enters the receiving dock
F_i	Time at which receiving truck i leaves the receiving dock
X_{ijk}	The number of units of product type k that is transferred from receiving truck i to shipping truck j

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck i to shipping truck j;} \\ 0, & \text{Otherwise;} \end{cases}$$

$$p_{ij} = \begin{cases} 1, & \text{If receving truck i preceeds receiving truck j in the receiving truck sequence;} \\ 0, & \text{Otherwise;} \end{cases}$$

$$q_{ij} = \begin{cases} 1, & \text{If shipping truck i preceeds shipping truck j in the shipping truck sequence;} \\ 0, & \text{Otherwise.} \end{cases}$$

The applied mathematical formulation for the truck scheduling problem in a cross-docking system is based on the following model:

Min T
s.t.

$$T \geq L_j, \qquad \forall j, \qquad (1)$$

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \qquad \forall i,k \qquad (2)$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \qquad \forall j, k \qquad (3)$$

$$x_{ijk} \leq M v_{ij}, \qquad \forall i, j, k \qquad (4)$$

$$F_i \geq c_i + \sum_{k=1}^{N} r_{ik}, \qquad \forall i \qquad (5)$$

$$c_j \geq F_i + D - M(1 - p_{ij}), \qquad \forall i, j \text{ and where } i \neq j \qquad (6)$$

$$c_i \geq F_j + D - M p_{ij}, \qquad \forall i, j \text{ and where } i \neq j \qquad (7)$$

$$p_{ii} = 0, \qquad \forall i \qquad (8)$$

$$L_j \geq d_j + \sum_{k=1}^{N} s_{jk}, \qquad \forall j \qquad (9)$$

$$d_j \geq L_i + D - M(1 - q_{ij}), \qquad \forall i, j \text{ and where } i \neq j \qquad (10)$$

$$d_i \geq L_j + D - M q_{ij}, \qquad \forall i, j \text{ and where } i \neq j \qquad (11)$$

$$q_{ii} = 0, \qquad \forall i \qquad (12)$$

$$L_j \geq c_i + V + \sum_{k=1}^{N} x_{ijk} - M(1 - v_{ij}), \qquad \forall i, j \qquad (13)$$

$$T, c_j, F_j, d_j, L_j, X_{ijk} \geq 0; v_{ij}, q_{ij}, p_{ij} \in \{0,1\} \qquad \forall i, j, k \qquad (14)$$

The objective function (T) aims to minimize the total operational time (makespan) of the cross-docking process. Eq. (1) ensures that the departure time of shipping trucks is lower than the total operational time. The latter equals to the departure time of the last shipping truck. Similarly, Eq. (2) guarantees that initially, the total number of products arriving by each receiving truck is equal to the total number of products loaded. Eq. (3) ensures that the total number of products loaded by each shipping truck is equal to its demand rate. Eq. (4) confirms that the x_{ijk} variables and the v_{ij} variables have the correct relationship. Eq. (5) reveals that the arrival and departure times of receiving truck i have a relationship as shown in the equation. Similarly, Eqs. (6) and (7) confirm that the arrival and departure times of the receiving trucks are similar to each other. Eq. (8) specifies that there is no received truck which may not be in sequence by preceding itself. The indications behind Eq. (9) to (12) are the same as to Eq. (5) to (8) explained earlier. The main difference of these constraints is their relation to the sequence of shipping trucks. As such, Eq. (13) illustrates that the departure time of a shipping truck and the arrival time of a receiving truck have a specific relationship with each other. Finally, all variables are guaranteed to be bounded as shown in Eq. (14).

3-Solution approach

The main innovation of this study is to develop three new versions of SEO to solve the truck scheduling problem. Based on our experiments, the authors found that these versions are the best improvements of the original of SEO. We have combined these ideas to generate the best improved versions of SEO. These improvements can help us to achieve better results to get the global optimum instead of local ones in comparison with other well-known and successful algorithms. In this section, first an appropriate solution representation is designed considering the continuous search space of SEO. Then, SEO is illustrated in details. Subsequently, different variations of SEO are designed to improve its original version performance and reduce the computational runtime by adding a set of adaptive procedures to the proposed modified SEOs.

3-1-Solution representation

The first step for solving a mathematical model by using an optimizer such as SEO is designing an appropriate solution representation to show that how the constraints of the model would be handled by metaheuristics (Fathollahi-Fard et al., 2018a; 2018b; 2018c). As shown by Golshahi-Roudbaneh *et al.,* (2017), Fig. 2 depicts the encoding scheme of the problem. For clarification purposes, the figure shows four trucks. The size of this matrix is equal to the summation of the number of both shipping and receiving trucks. Referring to the figure, Part I shows a specific sequence regarding the receiving trucks and Part II represents the sequence for the shipping trucks.

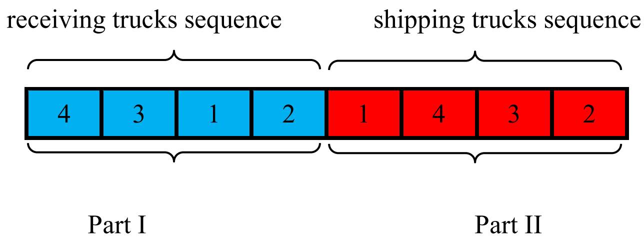


Fig. 2. An example of encoding plan (Golshahi-Roudbaneh et al., 2017)

Since the search space of SEO is continuous, a procedure is needed to perform the proposed encoding scheme of the problem (Fathollahi-Fard et al., 2018). For this purpose, a two-stage random technique named Random-Key (RK) is utilized (Sebtahmadi et al., 2017; Abdi et al., 2019; Fathollahi-Fard and Hajiaghaei-Keshteli, 2018). Although there are different techniques to encode the metaheuristics over the last decades, many papers confirm that the RK is the best feasible alternative to run the algorithms and saving the run time (Fathollahi-Fard et al., 2018a; 2018b). The main advantage of this technique is that there is no repair step. This technique has only two simple steps: In the first step, random numbers between zero and one are drawn by the proposed algorithm from a uniform distribution, shown as $U(0,1)$. In the second step, this solution is converted to a feasible representation solution as shown in Fig. 2. This modification procedure is performed by sorting the vector of this array to consider the sequence of allocation. Fig. 3 shows an example of the proposed representation method by the RK technique. The first row is generated by metaheuristics and the numbers in the second row are determined by the RK procedure.

0.82	0.73	0.24	0.64	0.45	0.91	0.89	0.51
4	3	1	2	1	4	3	2

Fig. 3. An example of the proposed representation method by the RK technique

3-2-Social Engineering Optimizer (SEO)

The promising performance of recent metaheuristics to solve complex problems has motivated several researchers to apply them in real-world and well-known engineering issues (Shakeri et al., 2012; Modiri-Delshad et al., 2016; Schwerdfeger et al., 2018; Abdi et al., 2019). For instance, Golshahi-Roudbaneh et al., (2017), as a novel research work, employed KA to tackle their truck scheduling problem. Similarly, in the area of supply chain network design, Samadi et al., (2018) for the first time introduced RDA to solve their proposed mathematical model. This study is the first attempt to apply SEO, developed by Fathollahi-Fard et al., (2018), to solve a truck scheduling problem. The advent of SEO has been inspired by social engineering theory. One attribute of social engineering systems can be the act of some indirect attacks by specific techniques to employ individuals who are willing to disclose their important data and information. To better understand the concept, assume there are a defender and attacker in the following explanations. The first step refers to the attacker aiming to the training and retraining activities from a defender registered in an online system. The attacker targets to gather a set of valuable information from the defender. This data may cover different topics and issues. For instance, by logging into the website of an attacker, a number of questions about famous special video clips, sports, public events and music, which may happen in the community or other dimensions of special family systems, can be requested. The next step illustrates how the attacker spots a social engineering attack. Clearly, to increase the robustness of an attack, one position, which has a higher probability of success should be identified. Generally, the attacker controls the defender in a position which is desirable for the attacker. The underlying assumption of this game is that the defender can think and understand like the attacker. Regarding the memory of learning, the attacker can choose different types of social engineering attacks including pretext placement, obtaining, diversion theft and phishing, which may be dependent on each other. This assumption increases the probability of success for the attacker. In each technique, there are a set of merits and demerits for each position with different profits and variables. How to respond to a social engineering attack is one of the main steps in a social engineering cycle. In this step, addressing the questions of how much information the striker wishes to gather as well as the reactions of the defender are very important and challenging tasks. Finally, the attacker seeks to steal data which might be useful to eliminate the defender and tries to conduct such attacks in another way or person. Based on the close analogy between the social engineering phenomena and metaheuristic optimization processes, an efficient optimizer called SEO is employed to solve the truck scheduling problem.

The SEO starts with two solutions, an attacker as the best solution and a defender. Regarding the training and retraining phase, a set of random tests for determining the defender's traits are considered. Subsequently, the attacker aims to assess the defender by a set of traits. The corresponding item of this action in the search space is to copy a trait from the attacker to the same trait to the defender and to calculate the rate of retraining for the attacker from the defender, accordingly. The next step is how to spot an attack of the attacker on the defender. The corresponding element of this action is changing the position of the defender by an approach in the feasible search space. In the following equations, def_{old} and att are the current positions of the defender and the attacker, respectively. Furthermore, def_{new} reveals the defender's new position. The algorithm uses four different techniques as follows:

- Obtaining (technique 1):

$$def_{new} = def_{old} \times (1 - \sin\beta \times U(0,1)) + \frac{(def_{old} + att)}{2} \times \sin\beta \times U(0,1) \tag{15}$$

- Phishing (technique 2):

$$def_{new}^{1} = att \times (1 - \sin\beta \times U(0,1)) + \frac{(def_{old} + att)}{2} \times \sin\beta \times U(0,1) \tag{16}$$

$$def_{new}^{2} = def_{old} \times \left(1 - \sin\left(\frac{\pi}{2} - \beta\right) \times U(0,1)\right) + \frac{(def_{old} + att)}{2} \times \sin\left(\frac{\pi}{2} - \beta\right) \times U(0,1) \tag{17}$$

- Diversion theft (technique 3):

$$def_{new} = def_{old} \times (1 - \sin\beta \times U(0,1)) + \frac{(def_{old} + att \times U(0,1) \times \sin(\frac{\pi}{2} - \beta))}{2} \times \sin\beta \times U(0,1) \tag{18}$$

- Pretext (technique 4):

$$def_{new} = \left(def_{old} \times U\left(0,1\right) \times \sin\left(\frac{\pi}{2} - \beta\right)\right) \times (1 - \sin\beta \times U(0,1)) + \frac{\left((def_{old} \times U\left(0,1\right) \times \sin\left(\frac{\pi}{2} - \beta\right)\right) + att)}{2} \times \sin\beta \times U(0,1) \tag{19}$$

For more information, the details of above techniques can be found in Fathollahi-Fard et al., (2018) and interested readers are referred to this paper. Regarding the response to the attack, the new position of the defender is computed again and the defender's old and current positions are compared with each other. Then, the best position is chosen to improve the solution of the algorithm. If the cost of the defender is better than the attacker, their positions are exchanged. Finally, the defender is eliminated and a new random solution in the search space is generated to form the new defender. To better understand the main steps of SEO, it can be shown in a graphical view. Fig. 4 shows the flowchart of the applied SEO and the pseudo-code of this optimizer is provided in Fig. 5.

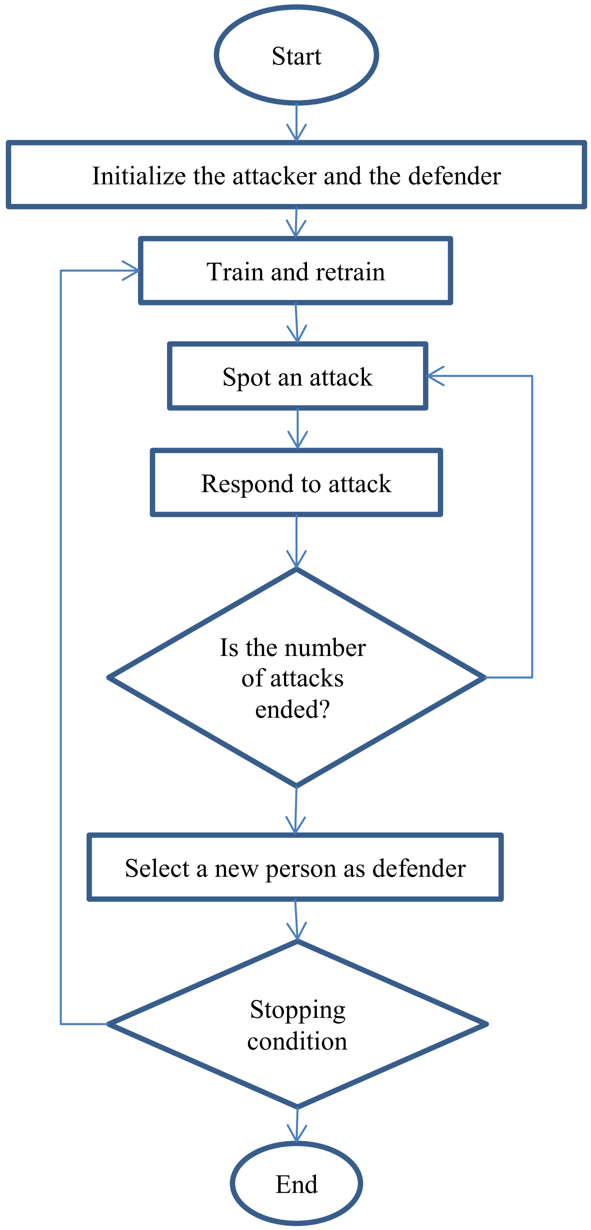


Fig. 4. The flowchart of the proposed SEO (Fathollahi-Fard et al., 2018)

```
Initialize an attacker and defender
It=0;
while It < Maxit
        Do training and retraining;
        Num_attack=0;
        while Num_attack < Max_attack
                  Spot an attack;
                  Check the boundary;
                  Respond to attack;
                  if the Objective Function (OF) of defender is lower
                  than attacker
```

```
            Exchange the defender and attacker position;
        endif
        Num_attack= Num_attack+1;
    endwhile
    Create a new solution as a defender;
    It=It+1;
endwhile
Return the attacker
```

Fig. 5. The pseudo-code of the applied SEO (Fathollahi-Fard et al., 2018)

3-3-Proposed modifications of SEO

This study is the first attempt to further develop the SEO performance by proposing three new modifications. Each modified version of SEO is built by putting different weights on the SEO features by creating different search strategies for striking a balance between exploration and intensification. To the best of our knowledge, these proposed variants of SEO have not yet been introduced. In the following sections, the details of each optimizer are provided in greater details.

3-3-1-MSEO_1

The training and retraining phase of SEO is one of the main steps for this algorithm. This step helps the algorithm to improve the exploitation properties. In the original version of SEO, the attacker aims to assess the traits of defender randomly to select an efficient one. Accordingly, α percent of traits are arbitrarily chosen. Then, the trait of the attacker will be copied to the trait of the defender. Here, an improved version of the training and retraining phase is presented. To further mimic the behaviors existing in real-world social engineering phenomena, an adoptive memory for the attacker has been added into the general training and retraining phase of SEO. Over the course of different iterations as algorithm runs, if a trait shows a successful effect on the fitness of the defender, it has more chance to be selected again in the next iteration of the algorithm. A roulette wheel strategy, a well-known evolutionary mechanism (Fu et al., 2019; Safaeian et al., 2019), is considered to select an appropriate trait from the defender. In this case, there are some other feasible alternatives such as the tournament selection. From our treatments to design this procedure, it is revealed that the roulette wheel strategy shows better impacts on the performance of the algorithm. For further clarification of an adopted roulette wheel, let us assume that there are four traits and the rate of α equals to 0.25. Therefore, one trait should be selected from the trait of attacker and be copied to the same trait of defender. Assume that the algorithm is running and it is in on its 10^{th} iteration and the rates of the success for these traits are {5, 1, 3, 1}, respectively.

Accordingly, the selection probabilities of these traits are {0.5, 0.1, 0.3, 0.1}, respectively. Apparently, the chance of the first trait is more than other traits. Based on this idea, the first modification of SEO called as MSEO_1 is proposed. In this modification, the other parts of the algorithm are similar to the original version of SEO as explained earlier.

3-3-2-MSEO_2

Another modification of SEO focuses on proposing a new spot for the defender inspired by a recent real technique called *reverse social engineering* (Krombholz et al., 2015). Due to the novelty of social engineering, the recent years have seen a great deal of interest in adopting different techniques to reveal the information of people by attackers. Reverse social engineering is a recent trend and more interesting for attackers in comparison with other feasible alternatives. This motivates our attempt to formulate this technique within the SEO. In this technique, instead of directly contacting the defender, the attacker tries to make the defender believe that they are a trustworthy individual. The goal of this technique is to make a potential victim if the defender asks for help. Generally, the attacker generates a problem for the defender. After that, the defender requests help. Finally, the attacker fixes the problem to get their desirable goals. To formulate this technique, two steps have been considered. The first step is the movement of the defender to a random position. The second step is the movement of this new defender in the neighborhood of the attacker. If the new solution of the defender has a better fit as compared to the one obtained prior to adopting the two-stage attack, this new solution will be replaced. Otherwise, the current defender will be used for the next attack.

$$def_{new}^{1} = \frac{(def_{old} + def_{old} \times (1 - \sin\left(\frac{\pi}{2} - \beta\right) \times U(0,1)))}{2} \times \sin\beta \times U(0,1) \tag{20}$$

$$def_{new}^{2} = \frac{(def_{new}^{1} + att)}{2} \times \sin(\frac{\pi}{2} - \beta) \times U(0,1) \tag{21}$$

The notations used above are adopted from the general version of SEO as illustrated earlier. This new technique with two separate steps is clearly different from the four techniques in the original version of SEO. The other features of this modification is similar to the general version of SEO.

3-3-3-MSEO_3

The main innovation of the third modification of SEO called as MSEO_3 is to introduce a dynamic parameter for the number of attacks. In this version, the number of attacks is not fixed for all iterations. It will be updated in each iteration based on the number of successful attacks and the number of iterations. Here, a successful attack means an attack in which the defender has been improved based on its fit during the attack. From Eq. (22), the initial number of attacks is shown as $Natt_0$ while the number of successful attacks is shown as $Acatt_{it}$ and all attacks are shown as $Natt_{it}$. As can be seen, *it* parameter shows the current iteration and *Maxit* is the maximum number of iteration. The number of attacks in the next iteration (i.e. $Natt_{it+1}$) will be updated using the formula

$$Natt_{it+1} = Natt_0 \times (1 - \frac{it}{Maxit} \times \left(1 - \frac{Acatt_{it}}{Natt_{it}}\right)) \tag{22}$$

This feature helps the algorithm to appropriately improve its both intensification and diversification properties. Similar to the modifications explained earlier, the rest of the MSEO_3 features are similar to the original version of SEO.

4-Experimental results

To evaluate the performance of the different algorithms, a set of comprehensive experiments is conducted to solve the truck scheduling problem. Since each modification has a particular contribution as a variant of SEO, all these three modifications can be merged together to generate several new approaches. Among possible combinations, we selected the hybrid of MSEO_1 and MSEO_3 called as MSEO_13, the hybrid of MSEO_1 and MSEO_2 named as MSEO_12 and the hybrid of all three suggested modifications called as MSEO_123. There are several other feasible alternatives for combinations however, in this study, the best modifications among all possible ones are chosen based on the results obtained during the computer experiments. Based on those experiments, these selected combinations are the most successful ones in comparison with other possible cases.

4-1-Data generation

To generate different truck scheduling problems, 20 small instances introduced by Yu (2002) along with 15 large instances applied by Golshahi-Roudbaneh et al. (2017) are solved and the results are compared with the relevant results in the literature (Yu, 2002; Yu and Egbelu, 2008; Mohammadzadeh et al., 2018). The main reason for adopting these benchmarks is that these papers treat problems similar to the proposed truck scheduling model in this study. All these instances are available in the appendix as Supplementary Materials F1.

Regarding the standard benchmark functions, this paper utilizes a set of standard functions to evaluate the proposed novel optimizers. The literature reports that there are more than 50 assessment functions (Ghorbani and Babaei, 2014; Fard and Hajiaghaei-Keshteli, 2016; Kaboli et al., 2017; Mortazavi et al., 2018; Schwarzrock et al., 2018; Etminaniesfahani et al., 2018; Fathollahi-Fard et al., 2019). In this work, 12 standard functions among all feasible alternatives were adopted from (Ghorbani and Babaei, 2014) provided originally from (Fard and Hajiaghaei-Keshteli, 2016) and Fathollahi-Fard et al., (2018). The main reason to choose these standard functions is that each of them has a particular feature to better evaluate the proposed algorithms. It means that the other test problems cannot make a difference and affect a significant result on the performance of algorithms. Therefore, we have selected these standard functions and numbered as P1 to P12. It should be noted that the original idea of SEO is also compared with these tests. All of them are minimization problems and their global optimum value is zero. The details of these functions are provided in the appendix as Supplementary Materials F2.

4-2- Tuning of optimizers

Since the presented optimizers have some controlling parameters, it is necessary to appropriately calibrate them to improve their performance (Bartz-Beielstein et al., 2010). If these metaheuristics are not tuned very well, their functions would be inefficient (Safaeian et al., 2019; Abdi et al., 2019). There are many elaborated methods for calibration such as response surface nested designs, F-Race methods and Taguchi experimental design method and so on. Among these feasible alternatives, this study applies Taguchi method to set the algorithms' parameters (Taguchi, 1986). There are several similar research studies that employed this methodology to fine-tune their optimizers (Fathollahi-Fard et al., 2018b and 2018c; Fu et al., 2019). The main advantage of this method is to reduce the tuning time for optimizers by decreasing the number of experiments. Their results also declared that the Taguchi can be more suitable for the tuning of optimizers in the case of a combinatorial optimization problem such as the proposed truck scheduling problem. To apply this, Taguchi method utilizes two well-known performance metrics called Signal-to-Noise (S/N) and Relative Percentage Deviation (RPD) to select the best parameters among all the candidate parameters (Fathollahi-Fard et al., 2019a). To the best of our knowledge, all the papers using Taguchi method have employed these two important metrics for calibration and there is no other feasible alternatives. For a minimization optimization model, the higher the value of S/N is, the better the quality of the algorithm. In contrast, the lower the value of RPD is, the better the capability of the optimizer. In this regard, the following two equations are presented to formulate the S/N and RPD metrics, respectively.

$$S/N = -10\log_{10}(Z)^2 \tag{23}$$

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \tag{24}$$

From Eq. (23), Z brings the value of the objective function whereas in Eq. (24), Min_{sol} is the best solution among all solutions during all runs and Alg_{sol} is the output of the algorithm. Basically, due to the stochastic nature of optimizers, such algorithms are run for 30 times and their results are utilized to calculate these two metrics in the Taguchi methodology. Accordingly, each parameter of the optimizers is a considered as a factor with a certain level, meaning each particular parameter has a different value. Table 1 shows the candidate values for each level as well as each parameter. The employed optimizers have the same factors. Four levels are considered for each factor.

Table 1. The list of factors and their levels of optimizers

Algorithms	Factors		Levels			
	Notation	Description	1	2	3	4
SEO_2, MSEO_13, MSEO_12 and MSEO_123	*Maxit*	The maximum number of iteration	1000	1500	2000	3000
	α	The rate of training	0.1	0.2	0.3	0.4
	β	The rate of spotting an attack	0.05	0.1	0.15	0.25
	Natt	The number of attacks	30	50	70	100

Since each optimizer has four factors with four levels (4^4), the total number of required experiments is 256×30 = 7680. It means that for each case of experiment, the optimizer needs to be run 30 times. Accordingly, by using an orthogonal array Taguchi reduces the total number of cases. According to the Table 1, Taguchi offers L_{16} to do the experiments. This orthogonal array means that the total number of testes equals to 16×30. By calculating the S/N and RPD, the best level for each factor is found. Regarding the truck scheduling problem and the standard benchmark functions, each optimizer is separately tuned. Due to page limitation, the results of S/N and RPD showing the rational on the choice of parameters are reported in the appendix as Supplementary Materials F3. Having fine-tuned all the algorithms, the best parameter value of the optimizers are presented in Table 2.

Table 2. The tuned values of algorithms

Algorithms	Tuned values of truck scheduling problem	Tuned values of standard benchmarked functions
SEO_2	*Maxit=2000; α=0.2; β=0.05; Natt=70;*	*Maxit=3000; α=0.1; β=0.05; Natt=70;*
MSEO_13	*Maxit=2000; α=0.2; β=0.15; Natt=100;*	*Maxit=3000; α=0.3; β=0.25; Natt=100;*
MSEO_12	*Maxit=3000; α=0.3; β=0.25; Natt=70;*	*Maxit=3000; α=0.3; β=0.15; Natt=70;*
MSEO_123	*Maxit=3000; α=0.1; β=0.05; Natt=70;*	*Maxit=3000; α=0.3; β=0.15; Natt=100;*

4-3-Comparison of the effectiveness and efficiency of the developed algorithms

In this section, a comparative work related to the truck scheduling problem is presented in which the developed algorithm is compared to the state of art methods in the literature (Yu, 2002; Yu and Egbelu, 2008; Mohammadzadeh et al., 2018). Firstly, the performance of the proposed optimizers is not only compared to each other but also benchmarked against the best solution found in the relevant literature (Yu and Egbelu, 2008). The results are shown in Table 3 in which the average outputs of optimizers along with their computational runtime for 30 times are calculated and provided. The best optimal value from the literature is also provided. It is observed that all the optimizers reached the best optimal solution found in the literature for the 30 times. Accordingly, the deviation of the algorithms, that is the average of solutions from the best solution, called the gap of the optimizers, is computed. The behavior of the algorithms' gap is presented in Fig. 6 by using an interval plot. In addition, the performance of the algorithms in term of computational time is given in Fig. 7.

From a general point of view, the results of optimizers shown in Table 3 demonstrate competitive outputs. The average results of developed optimizers such as MSEO_13, MSEO_12 and MSEO_123 are clearly better than the results obtained from the best original version of SEO and the average results of algorithms are close to the best optimal values found in the literature

(Yu and Egbelu, 2008; Golshahi-Roudbaneh et al., 2017). Regarding the metric of the gap of optimizers, the developed optimizers show better results. On the other hand, the averages of the gaps for MSEO_13 and MSEO_12 are close to each other. Overall, MSEO_123 with an average value of 0.03846 is the best algorithm in the table.

Fig. 6 shows the interval plot of the behavior of the gaps with a 95% confidence interval. It can be inferred from this figure is that the proposed MSEO_123 shows a robust behavior in comparison with other methods. Conversely, SEO_2 underperforms compared to the other methods.

Fig. 7 shows the computational time of optimizers and their comparisons. It is observed that the behaviors of SEO_2 and MSEO_13 share a set of similarities. Both of them show the best performance in this regard. Conversely, MSEO_12 shows the weakest performance compared to the majority of the instances.

Overall, MSEO_123 is the best existing optimizer in the category of small instances. However, it needs more time in comparison with the original SEO.

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

Instances	The optimal value found by (Yu and Egbelu, 2008)	The optimal value found by (Golshahi-Roudbaneh et al., 2017)	SEO_2			MSEO_13			MSEO_12			MSEO_123		
			M	CP	Gap	M	CP	Gap	M	CP	Gap	M	CP	Gap
1	1557	1562	1670.994	14.83819	0.073214	1693.259	14.3906	0.087514	1616.692	16.98439	0.038338	1643.082	17.22743	0.055287
2	1577	1577	1726.583	13.81573	0.094853	1620.306	14.63593	0.027461	1689.342	16.01982	0.071238	1657.795	17.96764	0.051233
3	1372	1372	1403.606	12.03937	0.023036	1459.99	12.47931	0.064133	1423.153	15.59835	0.037284	1444.479	15.50999	0.052827
4	1749	1789	1909.074	18.15976	0.091523	1795.27	17.82431	0.026455	1795.016	21.37343	0.02631	1789.538	22.09042	0.023178
5	1579	1579	1678.371	14.70704	0.062933	1623.131	14.72914	0.027949	1640.032	18.38499	0.038652	1618.305	18.3887	0.024892
6	1546	1546	1597.767	15.3648	0.033484	1595.462	15.89054	0.031994	1594.535	19.8224	0.031394	1594.292	19.45305	0.031237
7	1535	1535	1630.268	14.80739	0.062064	1621.916	13.8449	0.056623	1613.181	17.60528	0.050932	1614.064	18.65532	0.051507

8	1525	1525	1679.163	15.07109	0.10109	1610.751	15.49956	0.05623	1602.253	19.69359	0.050658	1561.836	17.79577	0.024155
9	1473	1473	1623.103	14.16271	0.101903	1602.01	13.58545	0.087583	1590.318	16.8786	0.079646	1507.187	18.24444	0.023209
10	1452	1452	1599.308	15.05054	0.101452	1607.504	15.45916	0.107096	1623.543	17.64709	0.118143	1499.53	17.54227	0.032734
11	2232	2232	2422.562	17.32537	0.085377	2314.477	17.8532	0.036952	2316.758	19.1117	0.037974	2293.387	20.06849	0.027503
12	2833	2833	3108.629	19.0486	0.097292	3098.468	18.76164	0.093706	3090.769	19.97785	0.090988	3088.331	22.2303	0.090127
13	2386	2403	2489.749	17.9462	0.043482	2492.213	17.79881	0.044515	2482.527	20.79313	0.040456	2482.158	21.42129	0.040301
14	2385	2413	2542.129	18.48669	0.065882	2538.319	18.73348	0.064285	2532.37	21.79393	0.06179	2531.243	22.28858	0.061318
15	2745	2762	2944.822	18.61363	0.072795	2946.375	19.0217	0.073361	2939.437	21.8436	0.070833	2940.773	20.77299	0.07132
16	2407	2407	2583.64	17.3832	0.073386	2525.836	17.53476	0.049371	2530.971	19.25178	0.051504	2427.977	20.4583	0.008715
17	1867	1885	2054.522	16.39642	0.10044	2042.578	16.69978	0.094043	2038.908	18.90215	0.092077	1936.516	20.16693	0.037234
18	2502	2642	2685.85	19.48674	0.073481	2583.308	19.48544	0.032497	2578.652	21.89823	0.030636	2579.823	22.17041	0.031104
19	2553	2553	2788.07	20.71672	0.092076	2683.442	21.26279	0.051094	2660.708	23.70438	0.042189	2606.449	22.65798	0.020936
20	2732	2926	2873.471	18.27144	0.051783	2776.399	18.7373	0.016251	2771.867	23.05516	0.014593	2760.806	21.23761	0.010544
Average					0.075077			0.056456			0.053782			0.03846

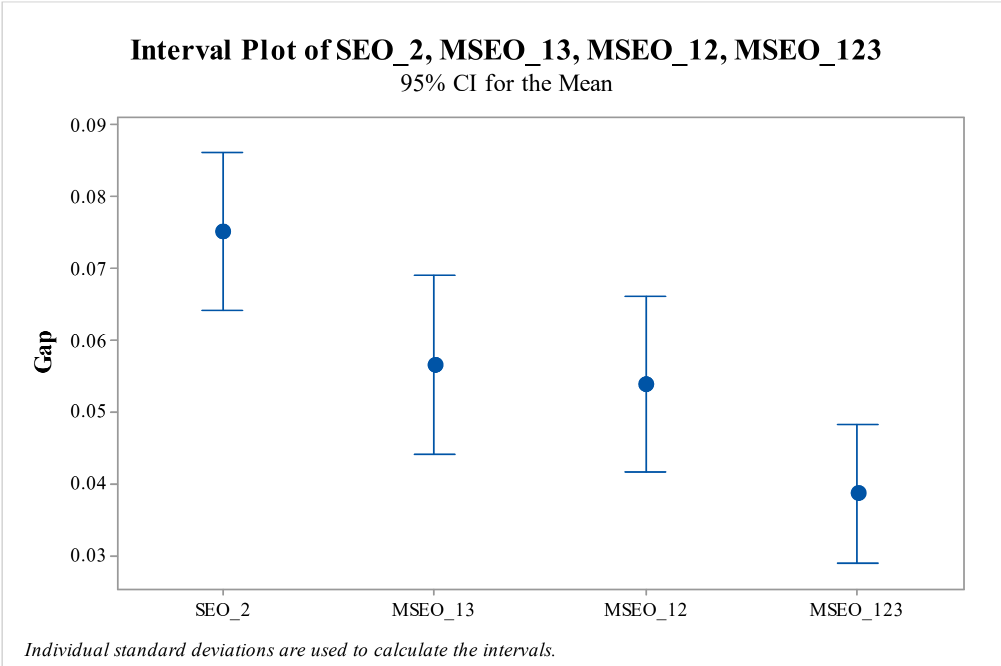


Fig. 6. Interval plot for Gap behavior of algorithms for small size instances

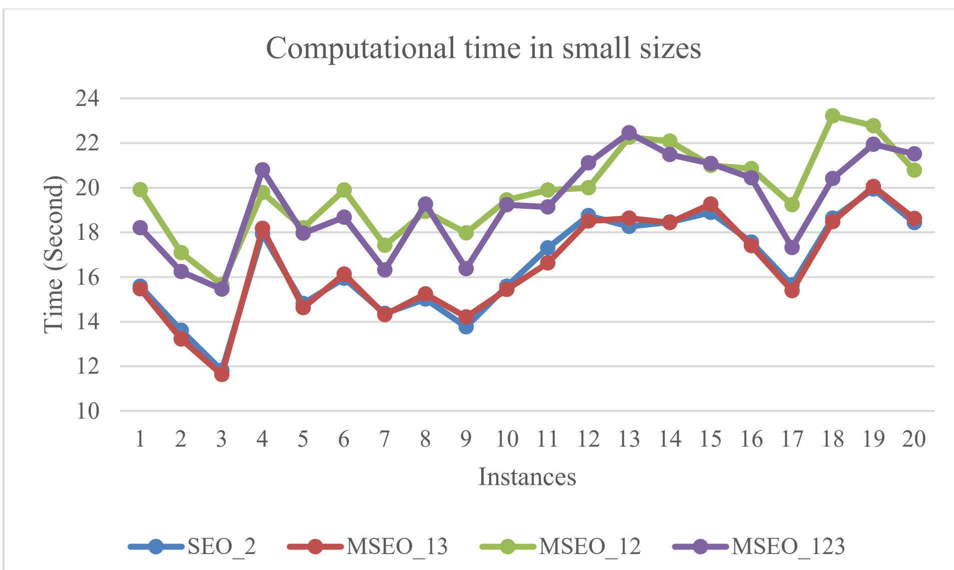


Fig. 7. Behavior of the optimizers in terms of computational time for small size instances

Table 4 shows the evaluations of the developed optimizers based on the best, the average, the worst, the standard deviation, computational time and the hitting time of the optimizers. The behavior of the optimizers in terms of computational time is illustrated in Fig. 8. Similarly, the behavior of the employed metaheuristics in terms of hitting time to determine and to compare the convergence rate of the optimizers is illustrated in Fig. 9. To evaluate the robustness of the optimizers, the interval plot is provided in Fig. 10 showing the RPD for the standard deviation of the algorithms during 30 run times. The best solutions of the truck scheduling problem coming from this study and the related works are given in Table 5 demonstrating the contribution of this research work and its results to the state of art in this field of research.

According to Table 4, the developed optimizers MSEO_13, MSEO_12 and MSEO_123 are clearly better than SEO_2 which is derived from the original version of SEO. The results confirm that there is no optimizer that shows the best performance in all case studies. Overall, MSEO_123 shows the best outputs in the majority of the instances. Further analyses are performed as shown in the following figures. As can be seen in Fig. 8, there is a set of similarities in terms of computational time of the algorithms. The results guarantee that by increasing the size of the problem, different performance of the optimizers dealing with different problem sizes can be easily observed. The results indicate that MSEO_123 needs more time in the majority of cases. Conversely, MSEO_13 is the best optimizer subject to the computational time.

As mentioned earlier, the better the hitting time is, the better the convergence rate therefore a lower value is preferable. Fig. 9 shows that based on the hitting time, MSEO_123 is the worst optimizer in the majority of case studies except for some large instances. The performance of MSEO_12 is also weak. Conversely, MSEO_13 shows better rates of hitting time in most of the instances. In conclusion, comparisons of the results show that the proposed MSEO_13 not only requires less computational time but also offers better capability in terms of hitting time.

Since the optimization mechanism of the proposed optimizers is stochastic in nature, a set of statistical analyses is needed to identify the best optimizers. Fig. 10 indicates that there is a clear difference between the effectiveness and the efficiency of the proposed optimizers. It is shown that the robustness of SEO_2 and MSEO_13 are similar to each other. In the same way, the performance of MSEO_12 and MSEO_123 are rather the same. Generally, based on RPD for the standard deviation of algorithms during 30 run times, all the developed optimizers perform better than their original versions. The proposed MSEO_123 is the best optimizer subject to these criteria and hence its results are more reliable.

Finally, the results of the developed metaheuristics are compared with the related works in the literature. The results of the proposed optimizers are compared with SA, PSO, the hybrid of PSO and SA (PSO-SA), KA and SFS adopted from Golshahi-Roudbaneh et al., (2017) as well as

VCS, RDA and WWO adopted from Mohammadzadeh et al., (2018). The proposed optimizers can improve the best state of the art results in six out of fifteen studies. Among the best results, MSEO_123 shows the best performance for 7 instances. Moreover, the proposed MSEO_13 and MSEO_12 are better than others only in 4 instances. In conclusion, the proposed MSEO_123 shows the best performance in which its results are more reliable than the others. However, it needs more computational and hitting times than the rest of the reported algorithms.

Table 4. Comparison of the metaheuristics with different criteria for 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
SEO_2	B	3034	3511	5036	3945	5089	7764	7041	7259	6141	5539	8212	7967	9170	10665	9522
	M	3241.5	3758	5381.6	4214.6	5307	8353.9	7410.5	7593.6	6604.82	5956.159	8763.857	8266.27	9771.617	11182.68	9925.317
	W	3511	4113	5877	4613	5627	9168	7970	8104	7274	6559	9514	8744	10549	11850	10480
	SD	188.97	200.6	290.84	214.12	174.4	583.46	300.15	262.01	379.71	376.902	479.98	229.04	463.34	439.4	354.7109
	CP	16.375	19.20	31.409	65.778	67.81	64.821	95.073	123.74	116.953	124.2591	166.1481	152.777	198.4523	286.6644	341.4428
	HT	5.0834	17.32	22.727	35.404	41.56	55.333	54.428	82.803	76.9093	104.1165	65.96373	135.254	174.1591	164.5124	325.3163
MSEO_13	B	3023	3488	5036	3934	5069	7751	7043	7255	6128	5529	8187	7950	9155	10663	9507
	M	3258.1	3780	5429.5	4251.2	5328	8428.3	7474.8	7653.8	6667.27	6012.72	8820.619	8315.42	9825.206	11229.56	9971.439
	W	3510	4096	5871	4605	5608	9180	7971	8113	7269	6553	9511	8730	10549	11843	10482
	SD	191.97	197.6	290.84	216.12	170.4	582.46	299.15	257.01	375.71	372.902	478.98	231.04	459.34	437.4	351.7109
	CP	16.406	19.56	29.385	63.860	66.33	69.075	90.687	116.61	125.791	117.93	166.50	148.537	181.188	272.401	354.383
	HT	10.75	12.64	14.28	39.61	42.86	47.51	63.12	79.32	87.23	99.52	126.77	109.42	178.32	206.85	275.134
MSEO_12	B	3027.8	3512	5020.5	3945.0	5061	7763	7022	7238.9	6125.92	5537.75	8189	7945.67	9150.11	10661.02	9511
	M	3348.9	3931	5607.1	4407.5	5448	8733.1	7674.7	7829.3	6914.36	6245.425	9106.15	8499.42	10120.03	11479.13	10172.66
	W	3486.5	4110	5858.6	4605.7	5614	9148.8	7954.4	8082.3	7252.26	6548.715	9499.122	8736.73	10535.7	11829.74	10455.96
	SD	191.75	175.4	266.67	208.48	158.3	563.94	286.13	246.08	352.008	374.3145	476.1126	207.373	450.1578	415.3733	350.4821
	CP	19.22	23.45	34.158	87.249	74.60	88.559	110.43	166.77	170.947	139.6532	174.1296	211.606	216.2008	319.7597	417.6699
	HT	15.67	16.31	19.84	48.15	56.33	49.24	66.73	108.26	96.35	109.33	145.32	186.72	169.32	221.64	208.13
MSEO_123	B	3036	3507	4996	3908.4	5070	7742	7035	7255.5	6110	5507	8206	7966.53	9140	10653	9511
	M	3352.4	3924	5608.8	4399.2	5462	8743.5	7689.4	7837.2	6927.89	6229.162	9110.806	8501.98	10118.18	11488.44	10179.95
	W	3488.1	4103	5871.1	4609.5	5629.	9172.7	7969.8	8086.5	7278.23	6538.288	9498.38	8731.46	10537.19	11846.29	10466.49
	SD	189.05	184.4	274.36	210.25	162.7	575.07	270.13	258.37	350.884	361.5926	455.1681	211.827	428.2455	428.4061	347.4784
	CP	22.346	22.38	47.396	95.905	102.6	88.687	96.045	127.80	142.176	132.3567	246.6578	241.295	188.9462	332.2008	383.073

HT	12.63	18.62	28.61	69.22	75.89	74.28	81.55	109.37	119.8	89.73	189.75	199.46	158.32	246.87	266.89

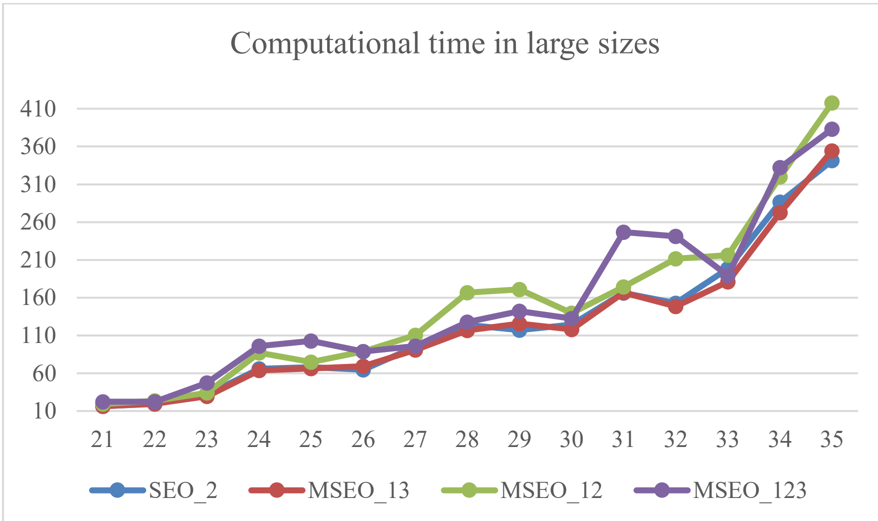


Fig. 8. Behavior of optimizers in term of computational time in large size instances

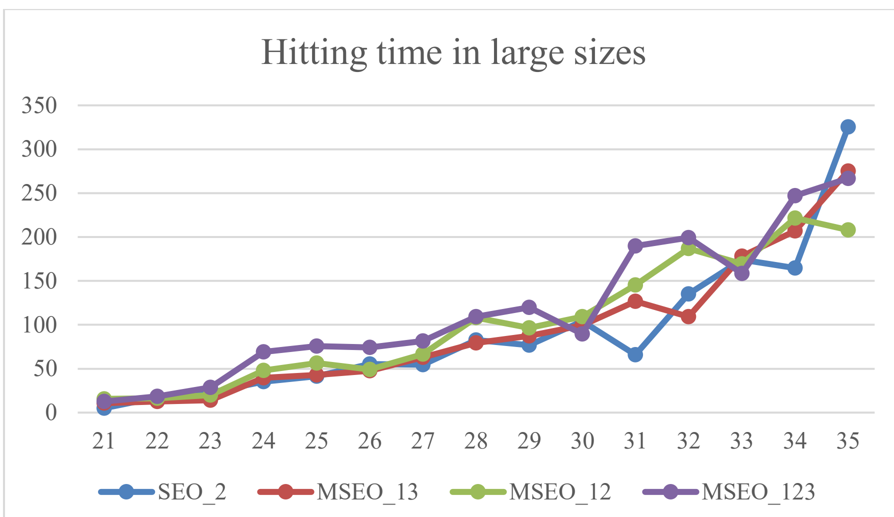


Fig. 9. Behavior of optimizers in term of hitting time in large size instances

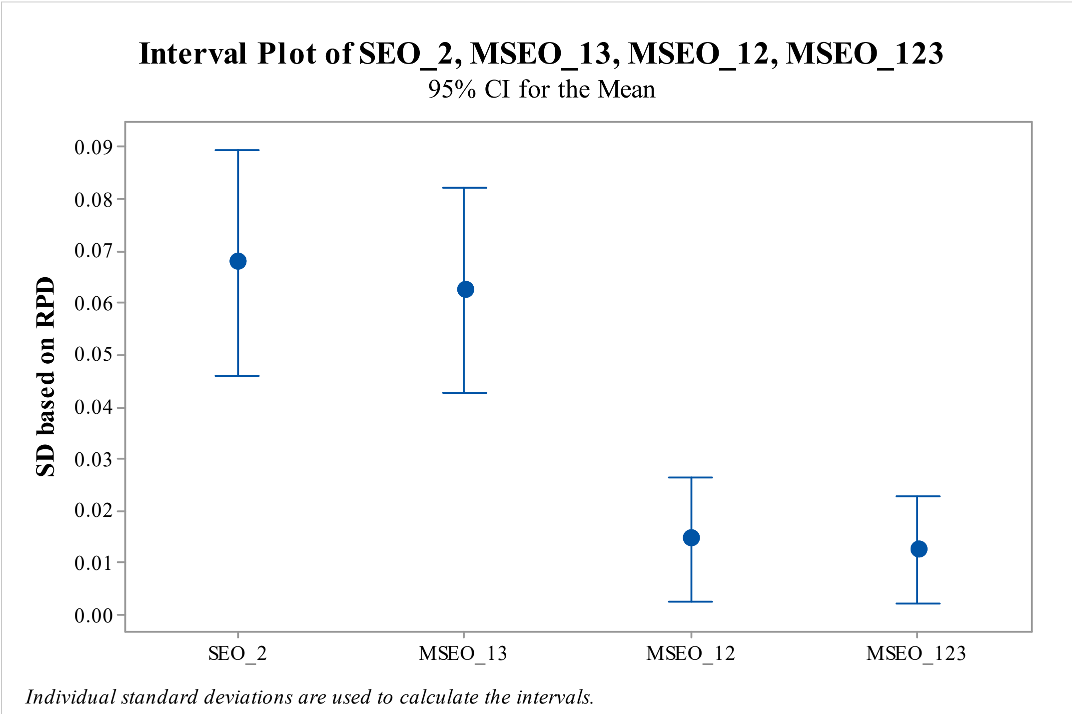


Fig. 10. Interval plot of standard deviation of optimizers in term of RPD in large size instances

Table 5

Proposed optimizers compared with related works (the minimum output found by algorithms)

Set	Golshahi-Roudbaneh et al., (2017)		Mohammadzadeh et al., (2018)		This work	
21	3046	Found by SA-PSO	**3012**	Found by RDA & WWO	3023	Found by MSEO_13
22	3505	Found by SA & SFS	**3466**	Found by WWO	3488	Found by MSEO_13
23	5026	Found by SFS	5026	Found by RDA	**4996**	Found by MSEO_123
24	**3826**	Found by SFS	3922	Found by RDA	3908	Found by MSEO_123
25	5161	Found by SA-PSO	5076	Found by RDA & WWO	**5061**	Found by MSEO_12
26	7799	Found by KA	**7688**	Found by VCS	7742	Found by MSEO_123
27	**6950**	Found by KA	**6950**	Found by WWO	7022	Found by MSEO_12
28	7484	Found by SFS	7246	Found by RDA	**7238**	Found by MSEO_12
29	6131	Found by SFS	6131	Found by RDA	**6110**	Found by MSEO_123
30	**5472**	Found by SA-PSO	5508	Found by VCS	5507	Found by MSEO_123
31	8327	Found by SA	**8182**	Found by RDA	8187	Found by MSEO_13
32	8166	Found by SA-PSO	7952	Found by RDA	**7945**	Found by MSEO_12
33	9300	Found by SA	9146	Found by RDA	**9140**	Found by MSEO_123
34	10758	Found by SFS	**10467**	Found by VCS	10653	Found by MSEO_123
35	**9338**	Found by SA & SA-PSO	9429	Found by VCS & WWO	9507	Found by MSEO_13

*The best values are shown in bold.

4-4-Evaluation of developed optimizers based on benchmark functions

To evaluate the performance of developed optimizers, a set of benchmark functions in the literature was employed. The details and formulations of these functions have been reported in the appendix as Supplementary Materials F2. To analyze the behavior of the developed optimizers MSEO_13, MSEO_12 and MSEO_123, the low (i.e. 30 variables) and high (i.e. 100 variables) dimensional of these functions were considered. Accordingly, these algorithms were not only compared with each other but also with the best results obtained from the original versions of SEO as well as a number of recent and optimizers from the state of art including Artificial Bee Colony (ABC), Imperialist Competitive Algorithm (ICA), Firefly Algorithm (FA), RDA and Linear-Success-History based on Adaptive of Differential Evolution (L-SHADE) (Fathollahi-Fard et al., 2018). We have selected these algorithms based on the original idea of SEO compared with them. The final results were given in Table 6. During 30 run times, the best, the worst, the medium and the standard deviation of algorithms have been provided. The rank of the algorithms in each standard function and dimension has been determined. Finally, a set of statistical analyses were performed to reveal the performance of the developed optimizers. Least Significant Differences (LSD) by using an interval plot for the applied optimizers in both low and high dimensional functions was conducted. Accordingly, the range of standard deviation based on Relative Deviation Index (RDI) was also computed. Since most of related papers (Ghorbani and Babaei, 2014; Fard and Hajiaghaei-Keshteli, 2016; Mortazavi et al., 2018) have utilized this metric to do the statistical analyses, this paper is also considered this metric to evaluate the range of standard deviation as can be seen in Fig. 11. The other feasible alternative is RPD which was utilized for calibration. As far as we know, there is no limit for the upper bound of RPD during the analyses. However, the outputs' ranges in RDI are between zero and one. This advantage motivates us to use this metric which can help to better do the comparison among algorithms.

Regarding Table 6, it can be observed that the developed optimizers in this study outperform the existing algorithms in the literature. The results confirm that the proposed optimizers have considerably contributed to the state of art of outputs for other applied algorithms. The results show that the average rank of these optimizers is better than other algorithms particularly for MSEO_123 with a rank of 1.675. Overall, the developed optimizers outperform the other existing algorithms in the majority of the case functions.

From Fig. 11, it is clear that there is a little difference between the performances of the algorithms subject to the dimension of the functions. In the low dimensional functions, the proposed optimizers are clearly better than other existing algorithms. It can also be observed that MSEO_123 depicts the best performance. However, the results show that the differences between the performances of the optimizers have been reduced in high dimensional functions. Therefore, the proposed optimizers are generally efficient in high dimensional assessment. As can be observed, the general versions of SEO are still better than other algorithms. These results confirm that it is possible for other new optimizers to outperform the proposed algorithms. As resulted from the experiments, the developed MSEO_123 performs better than the other proposed algorithms to solve the standard benchmark functions.

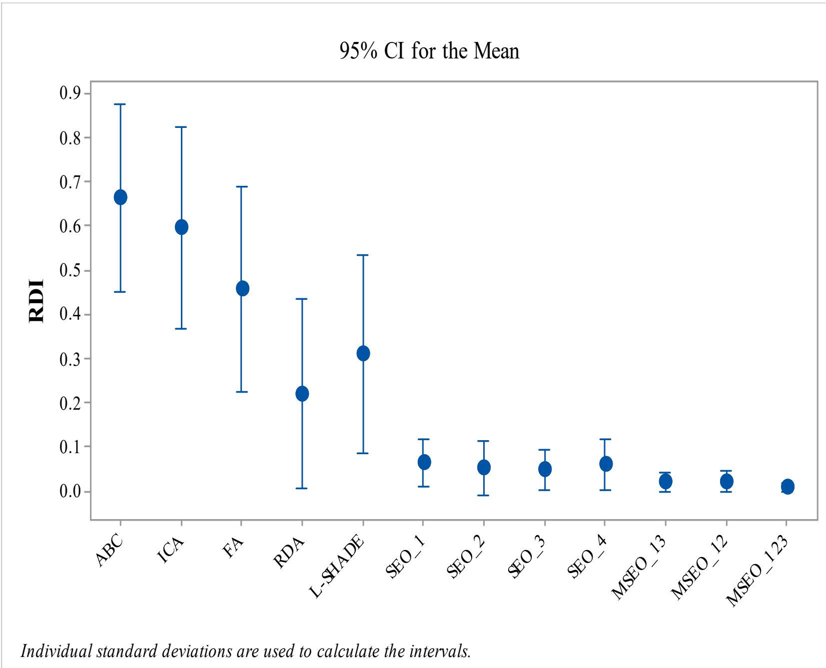
Table 6
The final outputs of algorithms in benchmark functions during thirty run times (B=best, W=worst, M=mean, SD=standard deviation, D=dimension, R=rank).

Function	D		Fathollahi-Fard et al., (2018)									This work		
			ABC	ICA	FA	RDA	L-SHADE	SEO_1	SEO_2	SEO_3	SEO_4	MSEO_13	MSEO_12	MSEO_123
P1	30	W	8.37E-01	1.37E-02	4.93E-04	3.52E-04	1.92E-02	4.18E-07	6.48E-09	3.51E-8	1.58E-09	8.16E-09	6.43E-09	4.19E-10
		M	2.51E-02	5.36E-03	6.91E-05	1.73E-06	1.17E-07	2.59E-08	2.18E-13	4.26E-11	5.37E-15	2.86E-14	3.85E-13	2.74E-15
		B	3.16E-04	2.51E-04	3.78E-08	2.86E-11	5.48E-09	1.23E-12	0	7.15E-16	0	0	0	0
		SD	0.46176	0.393635	0.000534	0.000136	1.20E-03	0.000047	0.000002	0.000007	0.000001	4.87E-06	8.41E-06	2.15E-07
		R	11	12	10	8	9	7	4	6	5	3	2	1
	100	W	2.42	5.84	2.76E-01	4.28E-03	2.39E+00	6.81E-06	8.54E-07	2.76E-05	5.38E-04	6.85E-08	2.58E-09	5.81E-09
		M	0.7534	4.32	5.47E-04	5.95E-04	1.22E-01	4.26E-07	2.16E-09	1.85E-06	1.85E-06	5.82E-10	3.19E-11	2.85E-10
		B	0.4627	2.17	2.81E-05	1.53E-05	8.37E-03	5.87E-09	5.27E-10	2.51E-07	7.29E-07	3.29E-12	0	6.82E-11
		SD	1.0325	3.2817	0.05894	0.07634	2.18E-01	0.000007	0.000005	0.000056	0.000084	5.38E-04	8.25E-05	6.39E-05
		R	11	12	8	9	10	5	4	7	6	2	1	3
P2	30	W	1.25E-04	1.87E-04	2.35E-04	2.64E-04	6.45E-04	3.19E-05	2.63E-06	3.12E-04	5.84E-05	5.91E-06	3.82E-07	2.81E-06
		M	2.68E-05	2.17E-06	3.18E-07	3.85E-08	8.87E-06	2.87E-09	3.27E-10	6.27E-09	1.58E-10	2.84E-10	6.18E-11	2.57E-11
		B	3.19E-06	3.12E-07	2.57E-08	1.25E-11	4.44E-07	1.25E-11	2.75E-14	2.86E-12	1.07E-12	6.82E-13	8.64E-15	1.34E-15
		SD	2.64E-03	8.53E-03	4.81E-03	5.73E-04	1.43E-02	3.17E-04	5.28E-05	2.18E-05	3.84E-05	4.81E-05	3.18E-06	1.66E-06
		R	12	11	9	7	10	8	3	5	6	4	1	2
	100	W	3.18E-02	5.15E-02	1.75E-02	3.27E-01	1.73E-02	2.55E-02	3.18E-02	1.56E-03	2.55E-03	2.58E-04	5.18E-05	7.28E-04
		M	7.29E-03	3.59E-04	3.21E-04	4.38E-06	2.49E-03	5.48E-05	2.15E-06	3.28E-06	7.13E-05	4.82E-07	6.21E-07	6.83E-08
		B	5.42E-05	2.88E-05	2.72E-06	5.74E-08	2.77E-05	5.17E-07	2.19E-08	3.65E-07	8.12E-07	5.82E-09	1.74E-08	5.29E-11
		SD	4.18E-01	3.15E-02	3.22E-03	3.16E-03	1.31E-01	8.41E-03	1.52E-03	2.59E-04	2.64E-03	5.12E-05	7.42E-04	6.83E-05
		R	10	11	9	3	12	7	5	8	6	2	4	1
P3	30	W	3.17E-04	2.16E-03	3.15E-02	5.93E-03	7.93E-05	2.61E-05	1.48E-07	2.74E-05	2.18E-05	6.27E-07	1.92E-05	1.85E-08
		M	2.62E-05	3.71E-06	4.71E-06	8.15E-07	4.37E-06	4.61E-10	2.12E-14	3.65E-12	6.72E-12	7.28E-15	6.31E-14	7.41E-16
		B	4.83E-08	4.85E-08	1.85E-09	3.15E-12	9.66E-09	2.54E-25	0	1.93E-22	1.82E-21	0	0	0
		SD	5.77E-03	3.15E-03	2.71E-02	8.52E-03	1.44E-03	4.27E-04	6.81E-06	1.77E-05	8.52E-04	2.81E-08	5.24E-07	8.13E-09
		R	12	11	10	8	9	5	4	6	7	2	3	1
	100	W	6.43E-02	4.36E-02	2.58E-01	4.26E-02	9.19E-03	4.63E-03	2.18E-05	2.67E-03	1.65E-04	2.51E-05	1.77E-04	6.31E-05
		M	3.17E-04	1.27E-04	2.83E-05	4.27E-06	1.06E-04	8.53E-08	3.19E-09	2.17E-9	5.14E-09	5.82E-10	4.78E-10	4.72E-12
		B	1.68E-06	3.19E-06	7.12E-07	1.29E-09	2.40E-07	7.28E-14	2.17E-15	2.65E-12	5.86E-13	7.82E-16	7.55E-18	6.81E-20
		SD	1.25E-01	1.78E-01	2.16E-01	2.81E-02	1.56E-02	2.18E-03	1.28E-04	3.81E-02	5.87E-03	3.82E-05	7.32E-05	7.41E-06
		R	12	11	9	8	10	5	4	7	6	3	2	1

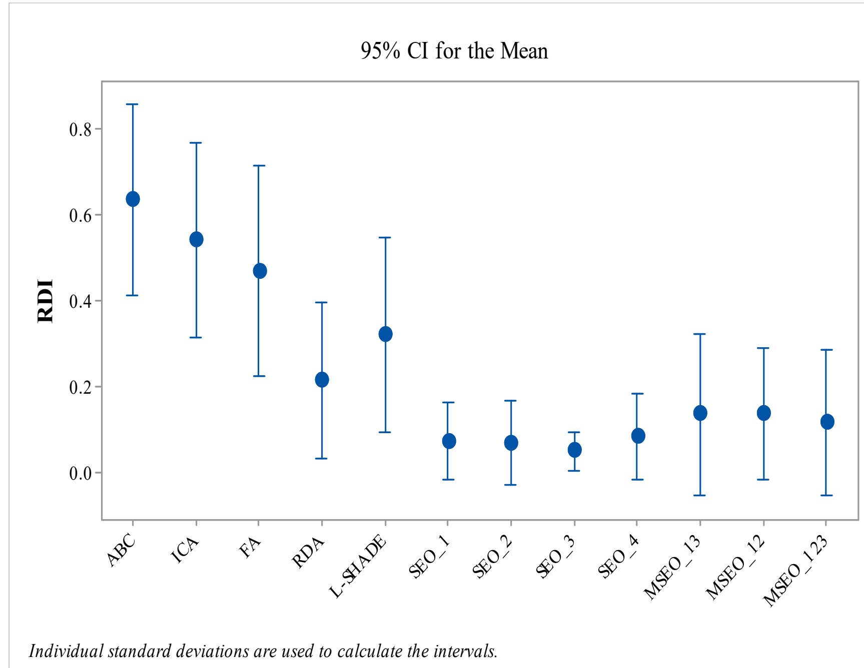
P4	30	W	9.54	6.19	7.52	5.01	1.19	3.28	2.81	5.92	1.42	1.085	1.37	0.94
		M	0.8659	0.8134	0.5894	4.82E-04	1.08E-01	3.91E-03	5.71E-03	5.18E-05	2.75E-06	6.82E-07	2.81E-06	5.16E-09
		B	1.29E-02	5.28E-03	2.71E-05	2.77E-06	3.23E-03	5.18E-08	2.18E-04	3.84E-09	1.25E-08	7.93E-09	6.82E-10	0
		SD	4.28	3.994	2.71E-00	3.82E-01	1.43E+00	0.7843	0.8754	0.2785	0.5645	4.82E-01	1.74E-01	2.81E-02
		R	12	10	8	7	11	5	9	4	6	3	2	1
	100	W	15.85	12.95	11.37	7.83	3.19E+01	6.83	8.93	5.81	4.97	2.75	4.91	1.58
		M	3.1854	5.8728	0.8623	3.61E-02	5.92	3.81E-02	5.12E-01	4.73E-02	1.86E-04	5.28E-04	3.68E-04	7.83E-06
		B	1.6809	0.7854	3.72E-03	5.87E-05	1.14	1.25E-06	2.61E-03	3.46E-03	8.25E-07	8.37E-06	1.68E-06	1.48E-08
		SD	1.7905	2.6847	3.78	1.9923	8.27	0.8645	0.9387	0.3623	0.3783	1.86E-02	6.27E-01	3.92E-03
		R	12	10	7	6	11	5	9	8	2	3	4	1
P5	30	W	114.46	119.76	102.54	87.54	39.9	75.18	68.52	82.17	84.94	49.82	50.21	40.12
		M	98.72	91.45	82.18	29.58	45.7	17.9328	16.475	19.58	18.6271	18.58	19.38	16.27
		B	30.99	31.58	30.85	24.61	15.8	15.29	14.28	15.4763	15.4763	7.41	14.81	9.84
		SD	1845.62	6254.91	1392.57	196.74	2.08E+03	28.97	48.73	38.761	40.632	29.86	52.8	27.13
		R	11	12	10	9	8	7	3	5	6	1	4	2
	100	W	156.75	142.68	116.75	95.47	57.5	89.25	91.27	88.43	91.27	78.9	81.3	64.7
		M	115.63	99.74	86.53	31.47	17.6	29.88	28.65	32.47	30.27	26.9	32.8	24.87
		B	32.16	34.82	33.81	26.91	15.1	24.86	25.651	27.91	25.48	18.68	20.5	18.93
		SD	1974.25	754.38	1473.89	215.48	215.3	30.81	58.93	39.78	41.58	54.8	91.6	56.3
		R	10	12	11	8	1	5	7	9	6	2	4	3
P6	30	W	3	7	0	0	0	0	0	0	0	0	0	0
		M	2.35E-09	2.57E-01	0	0	0	0	0	0	0	0	0	0
		B	0	0	0	0	0	0	0	0	0	0	0	0
		SD	2.7283	1.54724	0.95219	0.06066	0	0	0.353553	0.353553	0	0	0	0
		R	12	11	10	9	1	1	7	7	1	1	1	1
	100	W	87	23	13	21	1.90E+01	9	13	11	9	7	7	3
		M	2.19E-05	2.16E-00	1.75E-15	3.72E-10	8.80E-05	2.85E-16	3.72E-19	5.13E-20	5.82E-19	2.85E-21	6.82E-20	7.31E-24
		B	1	3	0	1	0	0	0	0	0	0	0	0
		SD	3.28E-02	1.57E-01	3.82E-02	3.16E-02	6.23E-02	5.28E-04	3.19E-04	5.14E-03	5.19E-04	6.83E-08	5.81E-07	6.71E-09
		R	11	12	8	9	10	7	6	4	5	2	3	1
P7	30	W	13.69	12.54	9.76	9.81	11.75	8.76	9.15	9.25	9.54	9.85	10.54	9.23
		M	10.86	10.25	8.46	8.26	9.86	7.56	7.15	6.82	6.91	6.97	8.14	6.25
		B	6.47	8.92	7.15	6.38	7.65	6.84	5.16	4.92	5.84	5.01	5.81	4.82
		SD	1.5843	1.3672	1.91	2.685	1.36	0.98	0.47	0.86	0.75	0.61	0.74	0.58
		R	8	12	10	7	11	9	6	2	5	3	4	1

	100	W	15.86	14.74	12.85	10.86	12.85	11.54	10.32	10.58	10.39	11.36	12.64	10.85
		M	11.63	10.54	9.86	9.24	11.76	9.54	8.99	8.59	9.64	8.52	9.57	7.85
		B	8.25	8.18	8.82	8.10	10.22	7.86	7.36	7.12	7.84	6.74	7.02	6.81
		SD	1.79	1.64	2.56	2.89	3.11	1.067	0.58	0.88	0.89	1.023	0.96	0.89
		R	10	9	11	8	12	7	5	4	6	1	3	2
P8	30	W	5.42E-02	1.67E-02	2.72E-04	3.81E-03	4.90E-03	2.84E-05	3.71E-06	4.16E-05	2.85E-05	4.82E-06	5.86E-07	1.86E-06
		M	1.48E-04	5.78E-04	6.24E-07	5.87E-05	7.30E-05	2.67E-09	1.36E-13	1.72E-12	2.84E-11	5.86E-13	6.97E-12	8.26E-12
		B	3.27E-09	3.11E-08	6.82E-11	5.46E-09	2.91E-06	2.81E-16	0	3.82E-14	2.85E-12	0	0	0
		SD	2.64E-02	4.18E-02	1.82E-03	7.53E-04	4.13E-02	5.86E-04	7.86E-05	2.45E-04	2.88E-04	5.92E-05	7.94E-05	6.15E-05
		R	9	11	8	10	12	5	2	6	7	1	4	3
	100	W	4.17E-01	5.83E-01	8.42E-03	5.17E-02	2.92E-01	3.76E-03	5.66E-04	2.94E-04	3.18E-04	5.93E-04	6.81E-05	3.81E-05
		M	1.54E-02	4.83E-02	1.28E-04	4.37E-07	2.42E-02	1.84E-07	1.28E-10	4.64E-08	3.95E-08	6.81E-09	3.17E-09	5.81E-10
		B	5.38E-05	5.92E-04	6.19E-05	8.29E-08	2.96E-04	8.29E-9	8.29E-14	8.29E-11	8.29E-10	4.18E-13	6.18E-12	6.83E-14
		SD	0.284115	0.088928	0.036815	0.000181	2.96E-02	8.7E-05	4.32E-06	3.04E-05	0.000129	7.82E-03	3.86E-02	2.85E-03
		R	10	11	9	8	12	7	1	5	6	3	4	2
P9	30	W	38.29	43.81	35.92	32.18	19.1	24.81	22.67	25.74	24.81	18.45	24.71	18.92
		M	28.11	17.85	29.32	16.931	14.1	10.44	3.96	6.29	7.59	2.61	2.85	1.587
		B	22.5918	16.3917	16.3917	8.5474	7.51	4.47	1.53	2.933	2.5933	1.86	1.51	0
		SD	10.67775	0.182485	25.73	20.563	3.56	4.55	1.55	5.022	7.978	0.98	1.05	0.99
		R	12	10	11	9	8	7	3	6	5	4	2	1
	100	W	51.23	52.19	43.64	42.58	29.6	31.86	32.86	33.81	32.54	28.71	34.51	29.18
		M	30.82	24.71	32.81	19.76	17.4	18.65	9.86	14.56	12.84	7.25	8.91	7.98
		B	18.16	12.76	14.83	11.56	10.4	10.52	9.23	8.97	8.75	6.18	6.72	5.62
		SD	6.32	8.182	7.93	6.15	3.05	5.47	5.82	4.03	5.83	4.96	5.13	4.61
		R	12	10	11	9	7	8	6	5	4	2	3	1
P10	30	W	2.71	1.28	3.88E-03	8.53E-04	9.03E-01	3.71E-02	6.22E-02	8.16E-02	4.72E-02	5.81E-04	5.13E-05	2.64E-06
		M	7.81E-02	4.82E-02	5.82E-05	6.26E-07	3.91E-02	8.52E-04	8.22E-02	1.73E-02	4.21E-02	5.76E-06	6.73E-07	4.62E-08
		B	2.97E-02	8.31E-02	5.88E-08	1.43E-09	9.90E-03	2.75E-06	1.86E-06	2.32E-06	5.96E-06	5.88E-09	3.82E-08	5.73E-12
		SD	4.37E-01	2.36E-01	6.82E-02	3.82E-03	1.46E-01	2.81E-03	1.84E-03	4.72E-03	8.21E-03	9.51E-04	2.74E-06	6.27E-04
		R	12	11	4	3	10	7	9	8	6	2	5	1
	100	W	6.34	3.91	0.684	0.523	2.61E+00	0.9634	0.8734	0.2716	0.7532	5.83E-02	4.81E-02	7.25E-03
		M	4.71E-02	5.81E-02	1.57E-04	5.81E-06	1.91E-01	3.16-02	4.92E-03	1.58E-02	1.57E-03	5.27E-03	5.38E-04	6.28E-04
		B	3.66E-04	2.55E-05	9.51E-06	8.94E-07	9.70E-03	3.82E-04	1.65E-05	4.61E-04	3.92E-05	2.48E-05	6.28E-05	1.75E-06
		SD	3.81E-01	2.48E-02	1.753-03	2.51E-04	2.07E-01	3.81E-03	1.94E-03	3.72E-03	2.85E-03	2.18E-03	6.27E-03	8.29E-03
		R	10	7	2	1	12	11	8	9	5	6	4	3

P11	30	W	0.43	0.875	0.36	0.89	4.05E-01	0.52	0.26	0.75	0.87	3.92E-02	6.81E-02	1.85E-02
		M	6.81E-03	1.92E-04	6.42E-04	8.92E-03	2.86E-03	1.93E-05	5.92E-05	1.84E-04	8.52E-04	5.73E-05	4.22E-06	4.81E-06
		B	7.22E-05	7.61E-06	4.91E-06	7.61E-06	1.94E-04	6.58E-07	5.81E-08	8.94E-07	3.41E-06	6.14E-07	3.29E-08	6.84E-08
		SD	0.0619	0.08662	0.071474	0.027377	7.06E-04	0.00969	0.00405	0.009652	0.002751	2.88E-03	5.27E-03	4.86E-04
		R	11	8	9	7	12	5	2	4	10	6	3	1
	100	W	4.92	5.37	2.81	3.97	2.69	2.55	1.64	2.17	1.98	0.98	1.25	1.08
		M	5.76E-01	6.89E-01	4.76E-02	1.85E-02	3.45E-01	5.47E-04	6.13E-02	6.84E-03	5.94E-03	4.81E-03	6.28E-04	1.85E-04
		B	5.92E-03	3.88E-04	5.82E-04	6.11E-04	1.29E-04	5.92E-05	3.11E-05	6.73E-04	5.93E-05	4.81E-05	6.73E-06	8.28E-06
		SD	0.2119	0.366	0.274	0.377	1.83E-01	0.0969	0.0805	0.0092	0.20751	1.07	0.86	0.976
		R	12	10	9	8	11	4	6	7	3	5	2	1
P12	30	W	1.25	1.45	1.25	5.82E-02	7.25E-01	5.28E-02	6.31E-02	8.11E-02	2.93E-02	7.83E-03	2.58E-03	1.85E-04
		M	3.82E-03	7.82E-05	5.83E-02	6.84E-09	3.91E-05	3.92E-05	8.15E-06	8.16E-07	8.10E-03	4.86E-05	2.77E-06	5.71E-06
		B	5.18E-04	4.29E-04	6.23E-04	2.84E-11	2.15E-04	7.83E-07	2.33E-08	8.42E-10	2.85E-07	8.15E-08	5.14E-08	6.84E-09
		SD	52.6121	21.88676	26.9302	0.899861	7.30E+00	0.390465	15.82594	0.147952	2.50135	5.83E-02	6.71E-03	7.15E-02
		R	10	11	9	1	12	7	6	2	8	4	5	3
	100	W	9.12	10.96	2.85	0.9372	2.98E+00	1.672	0.38	0.987	0.854	5.83E-02	5.71E-02	8.93E-02
		M	1.54E-02	5.66E-01	9.79E-01	1.26E-07	1.72E-01	1.45E-03	1.71E-04	9.01E-04	8.10E-03	5.28E-04	8.15E-04	5.27E-05
		B	6.38E-04	4.29E-04	6.19E-03	6.39E-08	2.77E-03	5.38E-04	6.31E-06	1.47E-07	2.85E-07	3.81E-06	6.14E-06	6.84E-07
		SD	59.21	45.86	34.02	0.9861	6.15E+00	0.0465	5.4	2.72	4.59	2.2	1.46	0.95
		R	9	10	11	1	12	8	5	4	3	7	6	2
Average of Rank			10.875	10.625	8.875	6.7916	9.7083	6.3333	5.1666	5.75	5.4166	3	3.1666	**1.625**



(a)



(b)

Fig. 11. The LSD intervals of the optimizers for low (a) and high (b) dimensional comparison

4-5-A real case study

This section presents a real case study to assess the performance of the developed algorithms in a real application. The aforementioned experiments confirm that the proposed novel approaches can better solve a set of benchmarks compared to other recent and state of art optimizers. In this regard, the proposed combinations are utilized to solve a real case study in Shahid Rejaee port as one of the biggest international cross-docking system in Iran[1]. The Shahid Rajaee Port plays a very important and vital role in Iran's economy and trade as the biggest container port of Iran that is in charge of handling the highest volume of container operations. Therefore, efficient optimizers that achieve robust answers are highly important for daily decisions of this cross-docking system.

The Shahid Rajaee port, as one of the main parts of cross-docking system in the field of import and export of cargoes in or out of the country, committed to support new investments through establishing logistics and distribution centers and providing value added services centers. It is estimated that over 1000 tons of products have been loaded and unloaded from trucks every day. About twenty types of products have been considered in this study. When importing, the products are loaded into trucks after the unloading from vessels. When exporting, the products are loaded into trucks after transferring the products from vessels. It is assumed that over 100 trucks are daily utilized in this cross-docking system. Less than 30 trucks are receiving the products and more than 60 trucks are shipping the products to wholesalers and or the retailers. Some details about this case study are available online in Supplementary Materials F4.

The proposed truck scheduling problem given in Section 2 is considered for this case study. All characteristics of the problem have been covered by the proposed cross-dock. Note that from the utilized benchmarks given in instance 35, the maximum number of trucks and products are 20 and 12, respectively. Therefore, due to the large size of presented case study is not able to be solved using exact methods. In this case, the performance of developed combinations are compared to the best algorithms from previous analyses. The RDA, SFS, SA-PSO and SEO_2 are selected for the comparison. The algorithms are run 30 times. The best, the worst, the average, the computational time and the standard deviation though the run times are reported. All results based on these criteria are given in Table 7 and the best value for each criterion is highlighted. It is shown that the developed MSEO_123 is the best algorithm in the majority of metrics considered for the assessment of the quality of the algorithms. Besides the computational time and the worst solution, the MSEO_123 shows a robust performance in this comparison.

Table 7. Results of case study (W=worst, M=average, B=best, SD=standard deviation, CPU=computational time in seconds).

Criteria	RDA	SFS	SA-PSO	SEO_2	MSEO_13	MSEO_12	MSEO_123
W	60771	61148	61032	60598	60853	**60546**	60751
M	60644.5	60984	60870.5	60420	60359	60365	**60258.5**
B	60265	60492	60386	60292	60265	60184	**60166**
SD	2106.646	2731.146	2689.512	1229.435	2211.04	2267	**1886.803**
CPU	2819.52	2694.26	2738.52	**2455.91**	2468.16	2602.57	2688.54

5-Conclusion and future works

The truck scheduling decision-making seeks to optimize both receiving and shipping trucks sequences and the allocation of them using a simplified objective function to formulate the cross-docking system. In many contexts however, and perhaps most especially in developing countries such as Iran where the management of cross-docking system is of particular concern, such a simplified approach to truck scheduling is failing to deliver satisfactory all outcomes under the recent advances of the supply chain and logistics management. To this end, a practical truck scheduling from the literature based on a real case study in Iran was introduced by this study. More practicality and efficiency need capable algorithms for this complicated optimization problem, which are robust and computationally manageable. Hence, some novel modifications of SEO as a successful recently-developed algorithm are proposed to identify the most efficient one.

In this paper, three novel strategies regarding the three steps of SEO are developed. Benefiting from each modification, three hybrid versions of SEO are also proposed among all possible combinations: MSEO_13, MSEO_12 and MSEO_123, and applied to solve a truck scheduling problem based on a set of standard benchmark functions as well as a real case study. Through an extensive comparison, the proposed optimizers provide competitive results compared to the existing metaheuristics in the literature.

A mathematical model formulating the proposed truck scheduling problem in a cross-docking center is explained followed by an encoding scheme demonstrating how to handle the constraint of the proposed model and the developed solution approaches. The proposed optimizers were tuned by the Taguchi experimental design methodology. According to a comparative study using benchmarked problems and a real case study, the proposed combinations better solve the

[1] https://shahidrajaeeport.pmo.ir/

problem, specifically the proposed MSEO_123 which has become a scientific contribution to this field of study.

Additionally, a set of benchmark functions is applied to show the high-efficiency of the proposed optimizers compared to other recent and state of the art algorithms. It can be observed that the proposed optimizers reveal the best performance especially in low dimensional evaluations based on a comprehensive analysis on both low and high dimensional evaluations. Although the comparison for high dimensional evaluation shows that there is a little difference between the performances of the algorithms, the proposed MSEO_123 is far better than the other compared methods. Finally, a real case study in Shahid Rajaei port as an international cross-docking system in Iran is considered to evaluate the algorithms in a real-world application. Similar to other analyses, the high-efficiency of developed MSEO_123 is confirmed by this case study.

There still are some limitations to this research, which open several new avenues for future works. Analyses such as convergence analysis on the proposed optimizers and the sensitivity analyses for the key parameters of the proposed algorithms may still need to be explored. The main future recommendations for this study are to develop more adaptive strategies to improve the proposed SEO. Accordingly, employing some other evolutionary mechanisms such as levy flight, crossover and mutation operators for the SEO would be an interesting extension of the current work.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version.

References:

Abdi, A., Abdi, A., Fathollahi-Fard, A. M., & Hajiaghaei-Keshteli, M. (2019). A set of calibrated metaheuristics to address a closed-loop supply chain network design problem under uncertainty. *International Journal of Systems Science: Operations & Logistics*, 1-18.

Abbasi, M. R., Shamiri, A., Hussain, M. A., & Aghay Kaboli, S. H. (2019). Dynamic process modeling and hybrid intelligent control of ethylene copolymerization in gas phase catalytic fluidized bed reactors. *Journal of Chemical Technology & Biotechnology*, *94*(8), 2433-2451.

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.

Amini, A., Tavakkoli-Moghaddam, R., & Omidvar, A. (2014). Cross-docking truck scheduling with the arrival times for inbound trucks and the learning effect for unloading/loading processes. *Production & Manufacturing Research*, 2(1), 784-804.

Awasthi, A., & Omrani, H. (2019). A goal-oriented approach based on fuzzy axiomatic design for sustainable mobility project selection. *International Journal of Systems Science: Operations & Logistics*, *6*(1), 86-98.

Bartz-Beielstein, T., Chiarandini, M., Paquete, L., & Preuss, M. (Eds.). (2010). *Experimental methods for the analysis of optimization algorithms* (pp. 978-3642025372). Berlin: Springer.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37(1), 32-41.

Chen, F., & Lee, C.-Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193(1), 59-72.

Duan, C., Deng, C., Gharaei, A., Wu, J., & Wang, B. (2018). Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions. *International Journal of Production Research*, *56*(23), 7160-7178.

Dubey, R., Gunasekaran, A., Sushil, & Singh, T. (2015). Building theory of sustainable manufacturing using total interpretive structural modelling. *International Journal of Systems Science: Operations & Logistics*, *2*(4), 231-247.

Etminaniesfahani, A., Ghanbarzadeh, A., & Marashi, Z. (2018). Fibonacci indicator algorithm: A novel tool for complex optimization problems. *Engineering Applications of Artificial Intelligence*, *74*, 1-9.

Fard, A. M. F., & Hajiaghaei-Keshteli, M. (2016). Red Deer Algorithm (RDA); a new optimization algorithm inspired by Red Deers' mating. In *International Conference on Industrial Engineering, IEEE.,vol. 12, (2016 e)* (pp. 331-342).

Fathollahi Fard, A. M., & Hajiaghaei-Keshteli, M., (2018). A bi-objective partial interdiction problem considering different defensive systems with capacity expansion of facilities under imminent attacks. *Applied Soft Computing*, 68, 343-359.

Fathollahi-Fard, A. M., & Hajiaghaei-Keshteli, M., (2018). A stochastic multi-objective model for a closed-loop supply chain with environmental considerations, *Applied Soft Computing, 69, 232-249.*

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Mirjalili S., (2018a), Multi-objective stochastic closed-loop supply chain network design with social considerations. *Applied Soft Computing*, 71, 505-525.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Tavakkoli-Moghaddam, R., (2018). The Social Engineering Optimizer (SEO). *Engineering Applications of Artificial Intelligence*, 72, 267-293.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Mirjalili, S. (2019). A set of efficient heuristics for a home healthcare problem. *Neural Computing and Applications*, 1-21.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Mirjalili, S., (2018c), Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem. *Applied Soft Computing*. 70, 701-722.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2018b). A bi-objective green home health care routing problem. *Journal of Cleaner Production*, 200, 423-443.

Fathollahi-Fard, A. M., Govindan, K., Hajiaghaei-Keshteli, M., & Ahmadi, A. (2019a). A green home health care supply chain: New modified simulated annealing algorithms. *Journal of Cleaner Production*, 118200.

Fu, Y., Tian, G., Fathollahi-Fard, A. M., Ahmadi, A., & Zhang, C. (2019). Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. *Journal of Cleaner Production*, 226, 515-525.

Ghorbani, N., & Babaei, E. (2014). Exchange market algorithm. *Applied Soft Computing*, *19*, 177-187.

Gharaei, A., Hoseini Shekarabi, S. A., & Karimi, M. (2019). Modelling and optimal lot-sizing of the replenishments in constrained, multi-product and bi-objective EPQ models with defective products: Generalised Cross Decomposition. *International Journal of Systems Science: Operations & Logistics*, 1-13.

Gharaei, A., Karimi, M., & Shekarabi, S. A. H. (2019a). An integrated multi-product, multi-buyer supply chain under penalty, green, and quality control polices and a vendor managed inventory with consignment stock agreement: The outer approximation with equality relaxation and augmented penalty algorithm. *Applied Mathematical Modelling*, *69*, 223-254.

Gharaei, A., Karimi, M., & Hoseini Shekarabi, S. A. (2019b). Joint economic lot-sizing in multi-product multi-level integrated supply chains: generalized benders decomposition. *International Journal of Systems Science: Operations & Logistics*, 1-17.

Gharaei, A., Hoseini Shekarabi, S. A., Karimi, M., Pourjavad, E., & Amjadian, A. (2019c). An integrated stochastic EPQ model under quality and green policies: generalised cross decomposition under the separability approach. *International Journal of Systems Science: Operations & Logistics*, 1-13.

Giri, B. C., & Bardhan, S. (2014). Coordinating a supply chain with backup supplier through buyback contract under supply disruption and uncertain demand. *International Journal of Systems Science: Operations & Logistics*, *1*(4), 193-204.

Giri, B. C., & Masanta, M. (2018). Developing a closed-loop supply chain model with price and quality dependent demand and learning in production in a stochastic environment. *International Journal of Systems Science: Operations & Logistics*, 1-17.

Golshahi-Roudbaneh, A., Hajiaghaei-Keshteli, M., & Paydar, M. M. (2017). Developing a lower bound and strong heuristics for a truck scheduling problem in a cross-docking center. *Knowledge-Based Systems*, *129*, 17-38.

Hajiaghaei-Keshteli, M., & Fathollahi-Fard, A. M., (2018), A set of efficient heuristics and metaheuristics to solve the two-stage stochastic bi-level decision-making model for the distribution network problem. *Computers & Industrial Engineering*. 123, 378-395.

Hao, Y., Helo, P., & Shamsuzzoha, A. (2018). Virtual factory system design and implementation: integrated sustainable manufacturing. *International Journal of Systems Science: Operations & Logistics*, *5*(2), 116-132.

Hlal, M. I., Ramachandaramurthya, V. K., Padmanaban, S., Kaboli, H. R., Pouryekta, A., Abdullah, T., & Ab Rashid, T. (2019). NSGA-II and MOPSO based optimization for sizing of hybrid PV/wind/battery energy storage system. *International Journal of Power Electronics and Drive Systems*, *1*(1), 463-478.

Hoseini-Shekarabi, S. A., Gharaei, A., & Karimi, M. (2019). Modelling and optimal lot-sizing of integrated multi-level multi-wholesaler supply chains under the shortage and limited warehouse space: generalised outer approximation. *International Journal of Systems Science: Operations & Logistics*, *6*(3), 237-257.

Kaboli, S. H. A., Selvaraj, J., & Rahim, N. A. (2016). Long-term electric energy consumption forecasting via artificial cooperative search algorithm. *Energy*, *115*, 857-871.

Kaboli, S. H. A., Fallahpour, A., Kazemi, N., Selvaraj, J., & Rahim, N. A. (2016a). An expression-driven approach for long-term electric power consumption forecasting. *American Journal of Data Mining and Knowledge Discovery*, *1*(1), 16-28.

Kaboli, S. H. A., Selvaraj, J., & Rahim, N. A. (2017). Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science*, *19*, 31-42.

Kaboli, S. H. A., Fallahpour, A., Selvaraj, J., & Rahim, N. A. (2017a). Long-term electrical energy consumption formulating and forecasting via optimized gene expression programming. *Energy*, *126*, 144-164.

Kaboli, S. H. A., & Alqallaf, A. K. (2019). Solving non-convex economic load dispatch problem via artificial cooperative search algorithm. *Expert Systems with Applications*, *128*, 14-27.

Kazemi, N., Abdul-Rashid, S. H., Ghazilla, R. A. R., Shekarian, E., & Zanoni, S. (2018). Economic order quantity models for items with imperfect quality and emission considerations. *International Journal of Systems Science: Operations & Logistics*, *5*(2), 99-115.

Konur, D., & Golias, M. M. (2013). Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 49(1), 71-91.

Krombholz, K., Hobel, H., Huber, M., & Weippl, E. (2015). Advanced social engineering attacks. *Journal of Information Security and applications*, *22*, 113-122.

Ladier, A. L., & Alpan, G. (2016). Cross-docking operations: Current research versus industry practice. *Omega*, *62*, 145-162.

Madani-Isfahani, M., Tavakkoli-Moghaddam, R., & Naderi, B. (2014). Multiple cross-docks scheduling using two meta-heuristic algorithms. *Computers & Industrial Engineering*, 74, 129-138.

Modiri-Delshad, M., Kaboli, S. H. A., Taslimi-Renani, E., & Rahim, N. A. (2016). Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options. *Energy*, *116*, 637-649.

Mohammadzadeh, M., Sahebjamnia, S., Fathollahi-Fard, A. M., & Hajiaghaei-Keshteli, M., (2018). New Approaches in Metaheuristics to solve the truck scheduling problem in a cross-docking center, *International Journal of Engineering, Transaction B: Applications*, 31(8), 1258-1266.

Mohtashami, A., Tavana, M., Santos-Arteaga, F. J., & Fallahian-Najafabadi, A. (2015). A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems. *Applied Soft Computing*, 31, 30-47.

Mortazavi, A., Toğan, V., & Nuhoğlu, A. (2018). Interactive search algorithm: A new hybrid metaheuristic optimization algorithm. *Engineering Applications of Artificial Intelligence*, *71*, 275-292.

Rabbani, M., Foroozesh, N., Mousavi, S. M., & Farrokhi-Asl, H. (2019). Sustainable supplier selection by a new decision model based on interval-valued fuzzy sets and possibilistic statistical reference point systems under uncertainty. *International Journal of Systems Science: Operations & Logistics*, *6*(2), 162-178.

Rabbani, M., Hosseini-Mokhallesun, S. A. A., Ordibazar, A. H., & Farrokhi-Asl, H. (2018). A hybrid robust possibilistic approach for a sustainable supply chain location-allocation network design. *International Journal of Systems Science: Operations & Logistics*, 1-16.

Safaeian, M., Fathollahi-Fard, A. M., Tian, G., Li, Z., & Ke, H. (2019). A multi-objective supplier selection and order allocation through incremental discount in a fuzzy environment. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-21.

Samadi, A., Mehranfar, N., Fathollahi Fard, A. M., & Hajiaghaei-Keshteli, M. (2018). Heuristic-based metaheuristic to address a sustainable supply chain network design problem. *Journal of Industrial and Production Engineering.* 35 (2), 102-117.

Sayyadi, R., & Awasthi, A. (2018a). A simulation-based optimisation approach for identifying key determinants for sustainable transportation planning. *International Journal of Systems Science: Operations & Logistics*, *5*(2), 161-174.

Sayyadi, R., & Awasthi, A. (2018b). An integrated approach based on system dynamics and ANP for evaluating sustainable transportation policies. *International Journal of Systems Science: Operations & Logistics*, 1-10.

Sarkar, S., & Giri, B. C. (2018). Stochastic supply chain model with imperfect production and controllable defective rate. *International Journal of Systems Science: Operations & Logistics*, 1-14.

Sebtahmadi, S. S., Azad, H. B., Kaboli, S. H. A., Islam, M. D., & Mekhilef, S. (2017). A PSO-DQ current control scheme for performance enhancement of Z-source matrix converter to drive IM fed by abnormal voltage. *IEEE Transactions on Power Electronics*, *33*(2), 1666-1681.

Schwarzrock, J., Zacarias, I., Bazzan, A. L., de Araujo Fernandes, R. Q., Moreira, L. H., & de Freitas, E. P. (2018). Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence. *Engineering Applications of Artificial Intelligence*, *72*, 10-20.

Schwerdfeger, S., Boysen, N., & Briskorn, D. (2018). Just-in-time logistics for far-distant suppliers: scheduling truck departures from an intermediate cross-docking terminal. *OR Spectrum*, *40*(1), 1-21.

Shakeri, M., Low, M. Y. H., Turner, S. J., & Lee, E. W. (2012). A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers & Operations Research*, 39(11), 2564-2577.

Shah, N. H., Chaudhari, U., & Cárdenas-Barrón, L. E. (2018). Integrating credit and replenishment policies for deteriorating items under quadratic demand in a three echelon supply chain. *International Journal of Systems Science: Operations & Logistics*, 1-12.

Peng, T., & Zhou, B. (2019). Hybrid bi-objective gray wolf optimization algorithm for a truck scheduling problem in the automotive industry. *Applied Soft Computing*, *81*, 105513.

Pourdaryaei, A., Mokhlis, H., Illias, H. A., Kaboli, S. H. A., & Ahmad, S. (2019). Short-Term Electricity Price Forecasting via Hybrid Backtracking Search Algorithm and ANFIS Approach. *IEEE Access*, *7*, 77674-77691.

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes*.

Tadumadze, G., Boysen, N., Emde, S., & Weidinger, F. (2019). Integrated truck and workforce scheduling to accelerate the unloading of trucks. *European Journal of Operational Research*, *278*(1), 343-362.

Tsao, Y. C. (2015). Design of a carbon-efficient supply-chain network under trade credits. *International Journal of Systems Science: Operations & Logistics*, *2*(3), 177-186.

Wolpert, D.H., Macready, W.G., (1997). No free lunch theorems for optimization. *IEEE Transaction on Evolutionary Computation*, 1 (1), 67–82.

Ye, Y., Li, J., Li, K., & Fu, H. (2018). Cross-docking truck scheduling with product unloading/loading constraints based on an improved particle swarm optimisation algorithm. *International Journal of Production Research*, *56*(16), 5365-5385.

Yi, S., Scholz-Reiter, B., Kim, T., & Kim, K. H. (2019). Scheduling appointments for container truck arrivals considering their effects on congestion. *Flexible Services and Manufacturing Journal*, 1-33.

Yin, S., Nishi, T., & Zhang, G. (2016). A game theoretic model for coordination of single manufacturer and multiple suppliers with quality variations under uncertain demands. *International Journal of Systems Science: Operations & Logistics*, *3*(2), 79-91.

Yu, W. (2002). Operational strategies for cross docking systems.

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.

Zuluaga, J. P. S., Thiell, M., & Perales, R. C. (2017). Reverse cross-docking. *Omega*, *66*, 48-57.