Expert Systems With Applications 245 (2024) 123115

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new evolutionary optimization algorithm with hybrid guidance mechanism for truck-multi drone delivery system

Cemal Yılmaz [a,d], Enes Cengiz [b], Hamdi Tolga Kahraman [c,*,1]

[a] Electrical and Electronic Engineering Department, Faculty of Technology, Gazi University, Ankara, Turkey
[b] Electronics and Automation Department, Vocational School of Ayancık, Sinop University, Sinop, Turkey
[c] Software Engineering Department, Faculty of Technology, Karadeniz Technical University, Trabzon, Turkey
[d] Department of Energetics, Faculty of Engineering, Mingachevir State University, Mingachevir, Azerbaijan

ARTICLE INFO

Keywords:
Fitness-Distance Balance-based Evolutionary Algorithm (FDB-EA)
TSP-D problem
Guide selection method
Stability analysis
Routing optimization

ABSTRACT

Synchronization of the Traveling Salesman Problem with Drone (TSP-D) is one of the most complex NP-hard combinatorial routing problems in the literature. The speeds, capacities and optimization constraints of the truck-drone pair are different from each other. These differences lead to the search space of TSP-D having a high geometric complexity and a large number of local solution traps. Being able to avoid local solution traps in the search space of TSP-D and accurately converge to the global optimal solution is the main challenge for evolutionary search algorithms. The way to overcome this challenge is to dynamically adapt exploitation and exploration behaviors during the search process and maintain these two in a balanced manner depending on the geometric structure of TSP-D's search space. To overcome this challenge, research consisting of three steps was conducted in this article: (i) three different guide selection methods, namely greedy, random and FDB-score based, were used to provide exploitation, exploration and balanced search capabilities, (ii) by hybridizing these three methods at different rates, guide selection strategies with different search capabilities were developed, (iii) by associating these hybrid guide selection strategies with different stages of the search process, the guidance mechanism was given a dynamic behavioral ability. Thus, the Fitness-Distance Balance-based evolutionary search algorithm (FDB-EA) was designed to achieve a sustainable exploitation-exploration balance in the search space of TSP-D and stably avoid local solution traps. To test the performance of the FDB-EA, the number of delivery points was set to 30, 50, 60, 80, and 100 and compared with twenty-seven powerful and current competing algorithms. According to the non-parametric Wilcoxon pairwise comparison results, FDB-EA outperformed all competing algorithms in all five different TSP-D problems. According to the results obtained from the stability analysis, the success rates and calculation times of FDB-EA, EA and AGDE algorithms were 88.00% (6308.79 sec), 58.40% (7377.43 sec) and 13.460% (34664.19 sec) respectively.

1. Introduction

Recently, the logistics and transportation sectors and many real-world problems (Lu & Wu, 2022; Lu et al., 2022) are experiencing rapid growth. There is a significant interest by researchers in the development of transportation optimization techniques for logistics networks since the most significant criteria in the execution of logistics activities are cost and time (Chen et al., 2019; Deng et al., 2019; Liang & Luo, 2022; Meng et al., 2023). Many cargo companies have started developing themselves in this field by preferring drones for package delivery in logistics (Frachtenberg, 2019). Today, large logistics companies such as Amazon, DHL, and UPS are in the process of using drones and land vehicles for package delivery (Hamid et al., 2023). Thus, drones have become very popular in recent years due to their effective potential in logistics. In addition to its unmanned operation, the ability to move without being exposed to traffic jams, an undesirable situation for land vehicles, is one of the important positive aspects of drones (Sajid et al., 2022). Although drones are superior to land vehicles in some aspects of logistics delivery, they have disadvantages regarding movement time and carrying capacity. The land vehicle can carry high-weight packages without time problems. However, the drone has a specific payload and airtime. Because of this situation, it is a more

* Corresponding author.
E-mail addresses: cemal.yilmaz@mdu.edu.az, cemal@gazi.edu.tr (C. Yılmaz), ecengiz@sinop.edu.tr (E. Cengiz), htolgakahraman@ktu.edu.tr (H.T. Kahraman).
[1] https://orcid.org/my-orcid?orcid=0000-0001-9985-6324.

https://doi.org/10.1016/j.eswa.2023.123115
Received 18 July 2023; Received in revised form 17 December 2023; Accepted 29 December 2023
Available online 5 January 2024
0957-4174/

reasonable solution to carry out delivery operations with land vehicles and drones in logistics (Murray & Chu, 2015).

The delivery model, in which the truck and drone deliver the parcel to customers simultaneously, makes the routing problem very difficult as it requires the coordination of both vehicles in terms of time and location. Thus, this situation reveals the need to develop new algorithms to solve this problem. There are specific objectives in formulating routing problems. These objective functions used especially for Traveling Salesman Problem with Drone (TSP-D) and Vehicle Routing Problem with Drone (VRP-D) can be expressed when the relevant works are examined as minimizing the completion time of the job (Boccia et al., 2021; de Freitas & Penna, 2020; Poikonen et al., 2019; Roberti & Ruthmair, 2021; Xu et al., 2022; Yurek & Ozmutlu, 2018), minimizing the cost of completing the job (Bouman et al., 2018; Choi & Schonfeld, 2017; Kuo et al., 2022; Tamke & Buscher, 2023; Wang & Sheu, 2019; Zang et al., 2022; Zhang & Li, 2023) and maximizing the number of people to be served in a day (Dayarian et al., 2020; Ulmer & Thomas, 2018). Murray and Chu (2015) are the first to bring this to the literature by presenting special formulations for the Flying Sidekick Traveling Salesman Problem (FSTSP), which aims to minimize the job's completion time. The truck leaves the depot carrying a drone and parcels. While the truck delivers to customers, the drone is launched to serve nearby customers. After the drone delivers to the customer, they meet with the truck. The drone moves with the truck unless it makes a delivery. The authors aim to devise a route plan to minimize the time the delivery process is completed to customers, and then the truck and drone return to the depot.

A good synchronization is necessary for the land vehicle and drone delivery operation to be carried out appropriately. Optimization algorithms based on metaheuristic techniques inspired by nature have a trendy place in this field. There are current studies on the optimum solution of vehicle routing problems using metaheuristic algorithms (Euchi & Sadok, 2021; Ferrandez et al., 2016; Lei et al., 2022; Li et al., 2016; Moeini & Salewski, 2019; Pehlivanoglu & Pehlivanoglu, 2021; Peng et al., 2018, 2019; Rajmohan & Ramasubramanian, 2023; Shi et al., 2022). Solving NP-Hard combinatorial problems with metaheuristic algorithms is quite challenging. As the problem size increases, it becomes more difficult for the optimization algorithm to solve. For the practical solution to these problems, the exploration and operation phases must be designed effectively. Therefore, the selection mechanism of quality solution candidates should be included in the algorithm (Masmoudi et al., 2022). Thus, the performance of the algorithm is ensured to be better.

Furthermore, algorithms need to be stable to solve the problem. It is critical to determine the success rate of the algorithm in problem-solving and to reveal the computational complexity (Duman et al., 2021). These criteria make it easy to compare the algorithm with its competitors.

The optimization of TSP-D presents challenges in several aspects. One of these challenges is the synchronization of a truck-drone pair with different characteristics. These two types of vehicles have different speeds, different capacities and different ranges. These characteristics lead TSP-D to have a multimodal search space with a large number of local solutions. Another challenge in the optimization of TSP-D is that the search space and the complexity of the problem increase exponentially with the increase in the number of delivery locations. This causes evolutionary search algorithms to get caught in local solution traps, leading to premature convergence problems and failure to find optimal solutions for TSP-D. To overcome these challenges, there is a need to design an evolutionary search algorithm that exhibits exploitation and exploration capabilities in a way that is compatible with the geometric structure of the search space of TSP-D.

In this paper, the Fitness-Distance Balance-based Evolutionary Algorithm (FDB-EA) is introduced to overcome the challenges in optimizing TSP-D. The main difference between FDB-EA and competing algorithms in the literature is the design of the guidance mechanism. The first of these differences is the use of three different selection methods in the design of the guidance mechanism of FDB-EA: greedy, random and FDB-based. FDB (Kahraman et al., 2020) is an efficient and current guide selection method that balances exploitation and exploration in the search process, which allows meta-heuristic search algorithms to avoid local solution traps. The second difference of FDB-EA from competing algorithms in the literature is that the application rates of the three methods used in guide selection are changed and hybridized. Thanks to the hybridization of three different guide selection methods, namely Greedy, random and FDB-based, guidance mechanisms with different levels of exploitation, exploration and balanced search capabilities are designed. Thus, it was possible to investigate and determine the hybridization rate and guide selection method that most effectively meets the requirements of the geometric structure of TSP-D's search space. The third difference of FDB-EA from competing algorithms in the literature is that it has a dynamic guidance mechanism. Thanks to this dynamic structure, different guide selection strategies are used in different phases of the search process life cycle of FDB-EA. These guide selection strategies differ from each other in their hybridization rates. The use of guide selection strategies with different hybridization rates at different stages of the exploration process life cycle is a new method in the literature. It is a candidate to be one of the most effective methods for achieving a sustainable exploitation-exploration balance.

The contributions of this paper can be summarized as follows:

- A new guidance mechanism specifically designed for the optimization of synchronous truck-drone routing problems has been introduced in the literature.
- A new evolutionary search algorithm, FDB-EA, is proposed to optimize truck-drone routing problems.
- The proposed algorithm is able to find the optimal route for all delivery problems in at least 5 % less time compared to its twenty-seven competitors.
- The stability of competitor algorithms for TSP-D optimization problems is analyzed for the first time in the literature. In the research conducted on five different routing problems, the stability of the FDB-EA in finding a feasible route is, on average, 30 % better than its best competitor.

The remainder of this paper is organized as follows. Section 2 explains the related study. In Section 3, the problem formulation is described in depth. Section 4 mentions the method of the paper, and in this section, the FDB selection method and the FDB-EA are given in subsections. Section 5 presents, subject to statistical analysis, the experimental study settings and the results of the experimental study. This section examines and analyzes the hybrid mechanism presented in detail.

Furthermore, the stability analysis of the FDB-EA algorithm and competitor algorithms are examined in this section. Finally, Section 6 is the conclusion. It also outlines possible future research directions on this topic.

2. Related study

Truck-drone routing problems differ according to the type of delivery vehicle, the purpose of the problem, and the cooperation between vehicles (Chung et al., 2020; Liang & Luo, 2022; Moshref-Javadi & Winkenbach, 2021; Otto et al., 2018; Rojas Viloria et al., 2021). The delivery problem with drones was first introduced by (Murray & Chu, 2015) and had two variants: Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP) and FSTSP. The PDSTSP has no synchronization between multi-drones and a single truck. The drones deliver by launching over the truck or from the depot while the truck delivers along the TSP route. FSTSP represents a single truck equipped with a single drone; the drone can perform a delivery operation called a sortie and then return to the truck at a customer or depot node. In this type of problem, there is synchronization between the truck and the drone. Agatz et al. (2018)

Table 1
Overview of the related study on truck-drone routing problems requiring synchronization and proposing solutions.

Reference	*#T*	*#D*	*#H*	*Obj.*	*Algorithm- (Number)*	Size	*Alg.*	*#Stb*	*Termination Criteria/Value*
Meng et al. (2023)	m	*m*	*hm*	*cc*	ISA- (3)	8–80	×	×	Max Iteration/–
Gunay-Sezer et al. (2023)	1	1	*hm*	*ct*	GA-AS- (5)	5–75	√	×	Max Iteration/1000
Kloster et al. (2023)	*m*	*m*	*hm*	*ct*	decomposition-based matheuristic, and iterated local search metaheuristic- (4)	25	√	×	Max Iteration/50
Lei et al. (2022)	m	*m*	*hm*	*cc*	DABC- (4)	6–200	√	×	Runtime limit/900 sec
Arishi et al. (2022)	1	*m*	*hm*	*cc*	DRL- (3)	70–250	×	×	Runtime limit/-
Bruni et al. (2022)	1	*m*	*hm*	*ct*	Benders-decomposition- (1)	10–20	×	×	Runtime limit/3600 sec
Euchi and Sadok (2021)	1	*m*	*hm*	*ct*	HGA- (2)	5–200	√	×	Max Iteration/1500
Baik and Valenzuela (2021)	1	1	*hm*	*ct*	K-means clustering- (2)	34–100	×	×	Runtime limit/1800 sec
Pina-Pardo et al. (2021)	1	*m*	*hm*	*ct*	Decomposition with two decisions- (2)	10–50	×	×	Runtime limit/10,800 sec
Murray and Raj (2020)	1	*m*	*ht*	*ct*	3-phased heuristic- (1)	8–100	√	×	Runtime limit/3600 sec
Salama and Srinivas (2020)	1	*m*	*hm*	*ct*, *cc*	K-means clustering- (1)	20–50	×	×	–
Ha, Deville, Pham, and Hà (2020)	1	1	*hm*	*ct*, *cc*	LS, GA (4)	10–100	√	×	Max Iteration/20,000
Dayarian et al. (2020)	1	1	*hm*	*cs*	Restricted and flexible resupply (heuristic)- (2)	60	×	×	Runtime limit/1800 sec
Rich (2020)	1	1	*hm*	*ct*	EA- (2)	10–100	√	×	Runtime limit/1800 sec
Jeong et al. (2019)	1	1	*hm*	*ct*	TPCSA- (4)	10–50	√	×	Max Iteration/10,000
Yurek and Ozmutlu (2018)	1	1	*hm*	*ct*	Iterative decomposition- (2)	10–20	×	×	Runtime limit/3600 sec
Agatz et al. (2018)	1	1	*hm*	*cc*	RFCS- (3)	10–100	√	×	Runtime limit/2000 sec
This paper	1	*m*	*hm*	*ct*	FDB-EA- (27)	30–100	√	√	*maxFEs*/Values was taken from Kumar et al. (2020).

Notes. m: multiple, hm: homogeneous, ht: heterogeneous, ct: minimization of completion time, cc: minimization of completion cost, cs: maximizing the number of customers served in a day, ISA: improved simulated annealing, GA-AS: the genetic algorithm with ant search-based solution method, DABC: dynamical artificial bee colony, DRL: deep reinforcement learning, HGA: hybrid genetic algorithm, LS: local search, EA: evolutionary algorithm, TPCSA: Two-Phase Construction and Search Algorithm, RFCS: route-first, cluster-second., maxFEs: the maximum number of fitness evaluations.

presented a dynamic programming algorithm. They have shown that their proposed modeling is more efficient than Murray and Chu (2015). The paper using the "truck first, drone second" approach showed effective results in large-sized instances. Murray and Raj (2020) have incorporated a heterogeneous fleet of drones into logistics delivery operations. In this FSTSP, drones have different operating characteristics. The operational delivery process was carried out by considering the energy consumption of drones with different characteristics in flight. A Mixed Integer Linear Program (MILP) has been created for the problem, and it has been tried to optimize FSTSP with heuristic algorithms. Ha et al. (2020) proposed a hybrid GA to solve the TSP-D. This algorithm has adaptive diversity control based on crossover and local search operators. According to computational studies, the proposed approach outperforms Murray and Chu (2015) and de Freitas and Penna (2020) regarding solution quality. Jeong et al. (2019) expanded the FSTSP by including the energy consumption depending on the load weight of the package and the no-fly zone. They solved problems with 50 customers and compared them with Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO). Lei et al. (2022) used a Dynamical Artificial Bee Colony (DABC) algorithm for VRP-D. Euchi and Sadok (2021) provide a MILP algorithm for the VRP-D with up to 200 customers and a hybrid genetic algorithm for its solution. Dayarian et al. (2020) discussed the vehicle routing problem with drone resupply. Before Dayarian et al. (2020), papers did not consider a drone-assisted delivery operation where drones resupply trucks. Pina-Pardo et al. (2021) developed the proposed drone resupply problem and studied instances of up to 50 customers. The waiting times of the truck were also analyzed considering the number of drone flights, capacity and the number of orders resupplied. There are delivery types where more than one customer is served in a drone sortie (Baik & Valenzuela, 2021; Gu et al., 2022; Luo et al., 2021, 2022; Masmoudi et al., 2022; Meng et al., 2023; Salama & Srinivas, 2020; Wang et al., 2022; Wen & Wu, 2022).

Table 1 provides an overview of the related work on truck-drone routing problems requiring synchronization and proposing solution approaches. The columns represent the number of trucks *"#T"*, the number of drones *"#D"*, whether multiple drones have different features from each other *"#H"*, the objectives *"Obj."*, the Algorithm used in the study- (number of competing algorithms compared) "*Algorithm-(Number)*", the lowest and highest values of the problem size on which the algorithm is tested *"Size"*, the approaches proposed, the sizes of the problems solved in the papers presented, and whether the development of solution algorithm "*Alg.*", whether stability analysis of algorithms is done or not "*#Stb*", termination criterion and value of the optimization algorithm "*Termination Criteria/Value*", respectively.

Table 1 shows the related studies on the use of drones and drones in logistics. Considering Table 1, the shortcomings of these studies are listed below.

- The number of competing algorithms used for the optimization of the problem in the literature is quite low.
- In some of the studies in the literature, only small size instances were used in problem optimization while conducting experimental studies

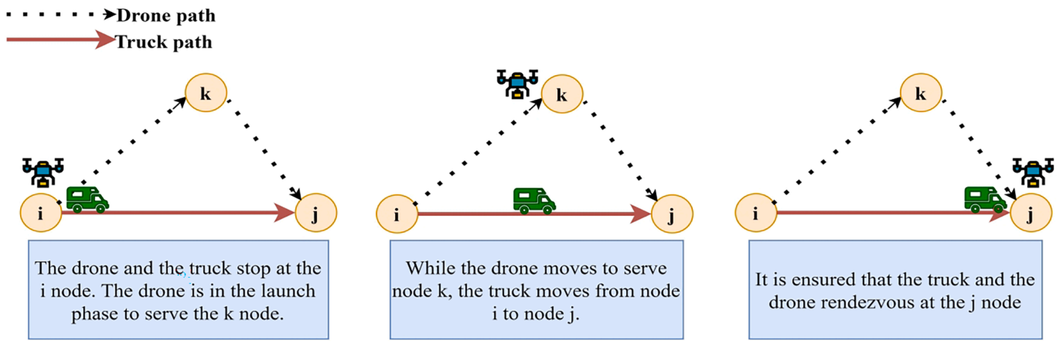

Fig. 1. TSP-D general parcel delivery process.

(Bruni et al., 2022; Jeong et al., 2019; Kloster et al., 2023; Pina-Pardo et al., 2021; Salama & Srinivas, 2020; Yurek & Ozmutlu, 2018). In addition, Kloster et al. (2023) and Dayarian et al. (2020) tested their algorithms with a single instance number.
- Table 1 shows that there are very few studies in which problem-specific optimization algorithms have been developed.
- Stability analysis of optimization algorithms has been performed in none of the studies in the literature.
- When the termination criteria of the algorithms used in the studies in the literature are examined, it is seen that two different termination criteria are used. The first one is the number of iterations, and the second one is the run-time limit. These criteria do not meet the criteria presented in existing studies (Kumar et al., 2020; Liang et al., 2019). This may eliminate fairness.

In order to overcome the shortcomings of the studies given above, the following research is considered: (*i*) Table 1 shows that the number of competing algorithms is in the range of 1–5. This is incomplete in the sense that the performance of the proposed method is not adequately compared. In this study, the performance of the proposed algorithm is effectively tested by using 27 current and well-known competing optimization algorithms. (*ii*) Unlike Kloster et al. (2023) and Dayarian et al. (2020) in Table 1, the problem sizes ($n = 30, 50, 60, 80, 100$) on which the proposed algorithm is tested are created in different sizes. Thus, the performance of the proposed FDB-EA in different sizes is evaluated. (*iii*) As explained in detail in the introduction, it is necessary to design a problem-specific algorithm for solving the TSP-D problem. Few of the literature studies in Table 1 have developed algorithms. In this study, the FDB-EA algorithm is designed as a problem-specific algorithm. (*iv*) Unlike all the studies given in Table 1, stability analysis is performed to perform a detailed performance analysis of the proposed FDB-EA algorithm. Thus, the success of FDB-EA in finding the optimal solution result and the problem complexity were revealed. (*v*) Unlike the studies in the literature, the termination criteria of the algorithms were run according to the maximum fitness evaluation number (*maxFEs*) of the algorithm. The maximum number of iterations given in Table 1 is 20000. In this study, the maxFEs value was determined to vary depending on the problem size with reference to Kumar et al. (2020).

3. Truck – Drone Problem Formulation

In the Traveling Salesman Problem (TSP), completing the job early by reaching the determined positions is aimed. Synchronization in delivery operations between the drone and the land vehicle (truck) comes into question with the inclusion of drones in TSP. While the drone only delivers packages, the truck also serves as a depot for the drone, in addition to delivering packages. Therefore, when the drone meets the truck, it is necessary to carry out a synchronous operation of the drone with the truck. Thus, the waiting time will be reduced. This will contribute significantly to the total completion time of the delivery operation.

This paper examined a variant of TSP-D, which is an extension of the FSTSP hypotheses proposed by (Murray & Chu, 2015). TSP-D shares many common features of FSTSP. The only major difference between the two is that TSP-D allows the drone to be retrieved by the truck from where it was launched. This is not the case in FSTSP (Kitjacharoenchai et al., 2019; Liu et al., 2022; Nguyen et al., 2020; Schermer et al., 2020). Murray and Chu (2015) aim to minimize the delivery completion time, in other words, the time it takes for the drone and truck to return to the depot. It aims to minimize the time needed to complete the work in this paper.

In Fig. 1, the drone is waiting on the truck, ready for delivery at node *i*. When the flight distance and battery life of the drone are suitable for parcel delivery, it is launched from the truck to move to node *k*. Meanwhile, the truck at node *i* moves to node *j* both to serve the customer at point *j* and as a depot for the drone. The drone serving the customer at node *k* needs to land on the truck to serve the new customer. So, the drone moves towards the *j* node, and they meet the truck. The relevant assumptions of this problem are listed as follows:

i. The drone is only allowed to visit one customer per flight.
ii. The drone will remain in flight continuously throughout a sortie, except for delivering the package to the customer. If the drone arrives at the rendezvous point before the truck, it cannot land temporarily to save battery power. It has to wait in the air in working condition.
iii. If the end of a drone sortie meets the truck, this must be done at a customer's location served by the truck.
iv. The drone cannot meet at any location between the truck and the customer locations. The truck cannot revisit the customer to meet with the drone.
v. Only one visit to a customer point is made. Neither can visit or take a break from any location other than the customer and the depot.
vi. The drone is decommissioned if the sortie of it ends at the depot.

3.1. Formulation for the TSP-D

In this section, we explain the formulation of TSP-D and then expand it with side constraints.

Drone sorties (i, j, k) are usually presented with launched node *i*, served customer node *j*, and rendezvous node *k* (Ha et al., 2018; Murray & Chu, 2015; Rich, 2020). Roberti and Ruthmair (2021) used two index variables while creating a mathematical model. This provides convenience due to the smaller number of indexes. Equations with two index variables were also used in this paper. Thus, the paper of Roberti and Ruthmair (2021) was taken as a reference for the TSP-D model. The model graph is defined as $G = (V, A)$. Vertex set $V = \{0, c+1\}$ gives the depots. The set of links between locations, arc set

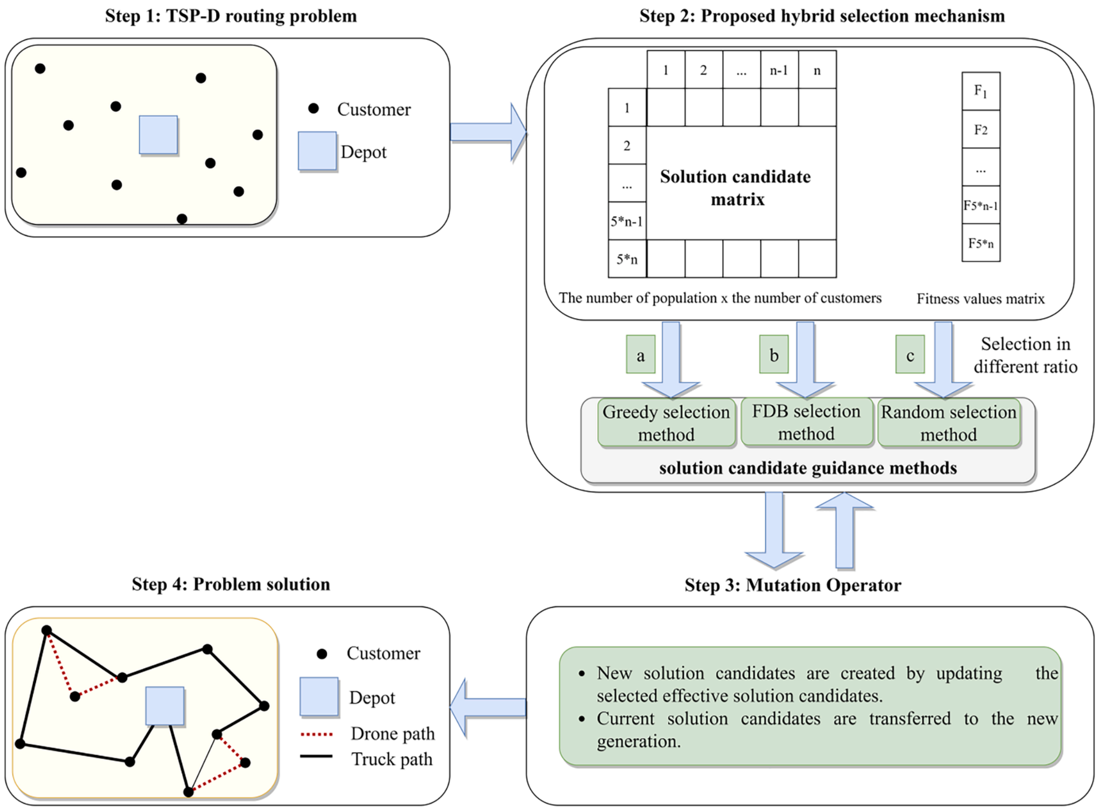

Fig. 2. General flowchart of the proposed FDB-EA-based routing optimization algorithm.

$A = \{(0,j)|j \in n\} \cup \{(i,j)|i,j \in n, i \neq j\} \cup \{(i,c+1)|i \in n\}$ and customer nodes $Nn = \{1, \cdots, c\}$ are represented. The delivery times for trucks and drone are $t^T_{i,j}$ and $t^D_{i,j}$, with $(i,j) \in A$, respectively. Truck traverses or drone traverses for delivery to a customer with $(i,j) \in A$ are defined as $x^T_{i,j} \in \{0,1\}$ and $x^D_{i,j} \in \{0,1\}$, respectively. There are three services to serve a customer: truck, drone, or both. The delivery operation y^T_i with only the truck, the delivery operation y^D_i with the drone only, the delivery operation y^C_i with both. a_i is a continuous variable that records the arrival time at each node. M is defined as a very big number.

The objective function (1) aims to minimize the total duration the drone and truck start at the depot, deliver packages to customers, and then return to the depot. Constraints (2) and (3) indicate the flow conservation constraints at each truck and drone customer delivery.

$$\min a_{c+1}, \tag{1}$$

$$\sum_{(i,j)\in A} x^T_{i,j} = \sum_{(j,i)\in A} x^T_{j,i} i \in n, \tag{2}$$

$$\sum_{i,j\in A} x^D_{i,j} = \sum_{(j,i)\in A} x^D_{j,i} i \in n, \tag{3}$$

Constraints (4) and (5) should start the movement of the truck and drone from the depot, respectively, during the delivery process and return to the last depot.

$$\sum_{0,i\in A} x^T_{0,i} = \sum_{i,j\in A} x^T_{i,c+1} i \in n, \tag{4}$$

$$\sum_{0,i\in A} x^D_{0,i} = \sum_{i,j\in A} x^D_{i,c+1} i \in n, \tag{5}$$

Constraints (6) explain that a node visited by a truck can be a truck or a combination node, while Constraints (7) explain that a node visited by a drone can be a drone or a combined node.

$$\sum_{(i,j)\in A} x^T_{i,j} = y^T_i + y^C_i i \in n, \tag{6}$$

$$\sum_{(i,j)\in A} x^D_{i,j} = y^D_i + y^C_i i \in n, \tag{7}$$

Constraints (8) explain that the drone can only serve one of each sortie during the delivery process to the customer. Constraints (9) mean that a drone or a truck should serve each customer.

$$x^D_{i,j} + x^D_{j,i} \le y^C_i + y^C_j i,j \in n, \tag{8}$$

$$y^T_i + y^D_i + y^C_i = 1 i \in n, \tag{9}$$

The valid constraints for a_i, in which the time reached at each customer node is recorded, are given in Constraints (10), (11). Constraints (10) are for trucks, while Constraints (11) are for drones.

$$a_i + t^T_{i,j} \le a_j + M\left(1 - x^T_{j,i}\right)(i,j) \in A, \tag{10}$$

$$a_i + t^D_{i,j} \le a_j + M\left(1 - x^D_{j,i}\right)(i,j) \in A. \tag{11}$$

Valid inequalities (12) and (13) are given below for the objective function. The total time the drone or truck completes its operations and returns to the depot cannot be greater than a_{c+1}.

$$\sum_{i,j\in A} t^T_{i,j} x^T_{i,j} \le a_{c+1} i \in n \tag{12}$$

$$\sum_{i,j\in A} t^D_{i,j} x^D_{i,j} \le a_{c+1} i \in n \tag{13}$$

Constraints (1)–(11) given above are generally common constraints to fundamental TSP-D problem models. In addition to these, constraints that are modeled based on drone endurance can be included.

Drone flight endurance: Drone endurance is defined as the maximum flight range of the drone, e. If the drone cannot fly more than e units after launching from the specified depot location, then the following constraints are added to formulation (14), (15):

$$\sum_{i,j\in A} t^D_{i,j}x^D_{i,j} + \sum_{j,i\in A} t^D_{j,i}x^D_{j,i} \le e + M\left(1 - y^D_i\right) i,j \in n, \tag{14}$$

$x^D_{i,j} \le x^T_{i,j}$, $(i,j) \in A$: $t^D_{i,j} > e$. (15)

Challenges of the TSP-D problem: The TSP-D problem, mathematically defined above, requires an algorithm that evaluates different customer visit sequences. TSP-D is a combinatorial optimization problem, and its difficulty increases exponentially as the number of locations to visit increases. As you add more locations or drones, the problem quickly becomes computationally intractable. Deciding when and where to use drones instead of trucks and how to switch between them seamlessly is also a challenging task. As given in Eqs. (14)–(15), there are time constraints when the drones must operate at a specific range. Taking these constraints into account increases the difficulty of finding a solution.

4. Method

The selection and search phases are the processes in which solution candidates are selected to guide the search process, and exploration and exploitation are conducted. Exploration identifies potential solution candidates in the search space, while exploitation aims to improve the solution's quality by investigating the solution's environment (Joshi & Bansal, 2020). Achieving productive diversity in the search process is challenging, as many promising solution candidates exist to explore in a high-dimensional and complex search space. It is necessary to improve the weak exploration and balanced search capabilities of the algorithm in order for the optimization process to produce successful solutions. Effective exploration in a large search space and an effective selection of search agents will improve solution performance. This paper uses the FDB selection method to improve the powerful exploration and balanced search capabilities of the FDB-EA algorithm. In this section, the FDB selection method in the first subsection and then the proposed FDB-EA algorithm in the second subsection are explained in detail.

4.1. Fitness distance balance (FDB) selection method

The FDB selection method significantly contributes to selecting effective solution candidates in the population search process. This method was recently introduced by Kahraman et al. (2020). Many optimization algorithms are improved by the FDB selection method. These algorithms showed superior performance compared to competitor algorithms in the paper in which they were introduced.

Although the FDB selection method is similar to the Greedy selection method, there is a significant difference. While selecting a solution candidate from the population, the fitness value is considered with the score value. In the first step of the FDB selection method, the distances of the solutions of all candidates from the solution candidate with the best fit P_{best} are determined. Different metrics, such as Euclidean, Manhattan, and Minkowski, can be used to determine this distance. The distance of each solution candidate from P_{best} is calculated in Eq. (16). In Eq. (17), the distance vector D_P of the solution candidates are given. The parameter X_{1P_i} given in Equation 16 is the first value of solution candidate i in the population matrix. $X_{1P_{best}}$ is the first value of the best solution candidate determined based on the objective function given in Eq. (1). In the second part of the FDB method, the scores of the candidates are calculated. Normalized fitness values *normF* and normalized

Algorithm-1: Steps of FDB selection method

start
Distance metric selection
Creating a random number for weight factor (w)
for $i = 1 : n$
Calculating the distance between P_i and P_{best}
$\forall^m_{i=1}, P_i \ne P_{best}, D_{P_i} = \sqrt{(X_{1P_i} - X_{1P_{best}})^2 + (X_{2P_i} - X_{2P_{best}})^2 + \cdots + (X_{mP_i} - X_{mP_{best}})^2}$ (16)
Creating D_P distance vector
$D_P = [d_1 \ldots d_n]_{nx1}$ (17)
end for
for $i = 1 : n$
Normalization of fitness and distance vectors
Calculating the score for candidates
$\forall^m_{i=1}, P_i, S_{FDB^1 P_{A^\circ}} = w * normF_{P_i} + (1-w) * normD_{P_i}$ (18)
$\forall^m_{i=1}, P_i, S_{FDB^2 P_{A^\circ}} = normF_{P_i} * normD_{P_i}$ (19)
Creating score vector
$S_P = [s_1 \ldots s_n]_{nx1}$ (20)
end for
Selecting candidates from S_P
end start

Table 2
Hybridization strategies based on a, b, and c parameters.

Hybridization strategies (HS)	Guide selection methods		
	a(Greedy based on fitness value)	b(FDB)	c(random)
HS-1	0.5	0.5	0
HS-2	0.5	0.4	0.1
HS-3	0.4	0.4	0.2
HS-4	0.3	0.3	0.4
HS-5	0.2	0.2	0.6

Table 3
Six different versions of FDB-EA's guidance mechanism.

Guidance Mechanism Versions	*Phase-1 (FEs < MaxFEs/4)*	*Phase-2 (MaxFEs/4 < FEs < MaxFEs/2)*	*Phase-3 (MaxFEs/2 < FEs)*
Case-1	HS-1	HS-2	HS-5
Case-2	HS-2	HS-1	HS-4
Case-3	HS-1	HS-3	HS-2
Case-4	HS-2	HS-4	HS-2
Case-5	HS-5	HS-1	HS-5
Case-6	HS-4	HS-1	HS-5

FEs: current number of fitness evaluations, maxFEs: the maximum number of fitness evaluations.

Table 4
Friedman scores of the proposed algorithms for different numbers of customers.

			Versions of FDB-EA					
		EA	Case-1	Case-2	Case-3	Case-4	Case-5	Case-6
Dimension	$n = 30$	5.76	3.93	**3.02**	4.00	3.86	4.05	3.38
	$n = 50$	5.95	**2.76**	2.86	3.52	3.48	3.67	5.76
	$n = 60$	5.86	**2.81**	3.24	4.05	4.00	4.43	3.62
	$n = 80$	6.71	3.86	3.95	**1.81**	3.38	4.00	4.29
	$n = 100$	6.10	**3.24**	4.00	3.67	3.38	4.02	3.60
	Mean	6.08	**3.32**	3.41	3.41	3.62	4.03	4.13

Table 5
Wilcoxon pairwise comparison results between EA and FDB-EA versions (proposed).

	Versions of FDB-EA					
	Case-1	Case-2	Case-3	Case-4	Case-5	Case-6
vs. EA $+/=/-$	5/0/0	5/0/0	5/0/0	5/0/0	5/0/0	5/0/0

distance values $normD_P$ from candidates are used when calculating these scores. Numerical values are used to prevent two parameters from dominating each other. In the calculation of member scores, the weight factor w is used to determine the effect of fitness and distance values. For the P population, w is chosen between 0 and 1. In this study, this weight value was chosen as 0.5 to ensure a balanced effect of $normF$ and $normD_P$ in the score calculation. Eq. (18) or Eq. (19) calculates each candidate's score. The score vector obtained using Eq. (18) or Eq. (19) is given in Eq. (20).

4.2. Proposed FDB-EA algorithm

In evolutionary algorithms, the task that the guidance mechanism has to fulfill changes dynamically throughout the lifecycle of the search process. For example, in the early stages of the search process, the diversity within the population is high because populations are initially created randomly. In the later stages of the search process, due to the "fitness-based survival mechanism", the individuals (solution candidates) with the best fitness values become dominant, and the diversity within the population decreases. Based on this information, it is understood that the exploitation and exploration behaviors required by the population change throughout the life cycle of the search process. This necessitates the use of different methods in the design of the guidance mechanism and the application of dynamic strategies in evolutionary search algorithms. This requirement is taken into account in this paper, and hybrid guidance strategies are developed to dynamize the operation of the guidance mechanism of FDB-EA.

The difference between the FDB-EA and EA (Rich, 2020) algorithms comes from the design of the guidance mechanisms of both algorithms. To facilitate the understanding of this difference, this section describes the design of EA's guidance mechanism and briefly discusses its shortcomings below.

Design of the guidance mechanism in EA: Only the fitness-based greedy selection method was used to design the guidance mechanism of the EA algorithm. In the algorithm used in the solution of the problem handled by Rich (2020), four new solution candidates are created from this solution candidate by considering the solution candidate with the best fitness value of the subpopulations with five solution candidates. The solution candidate with the best fitness value and the four newly formed solution candidates ensure the regeneration of the old subpopulations with five solution candidates. A new population is created by processing the same process in all subpopulations with five solution candidates. The fitness-based greedy selection method fulfills the exploitation task in evolutionary algorithms.

Meanwhile, finding optimal solutions in the search space depends on evolutionary algorithms having both a precise exploitation capability and an effective diversity capability. The exploitation capability allows for a precise search in the neighborhood of good solutions in the search space, while the diversity capability allows the avoidance of local solution traps and converges to global solutions. According to these explanations, taking only the solution candidate with the best fitness value and ignoring the others in the subpopulations with five solution candidates under consideration negatively affects diversity.

Problems in the design of EA's guidance mechanism: as can be seen from the explanations in the paragraph above, using only fitness-based greedy selection in the design of EA's guidance mechanism leads to all solution candidates in the population carrying the genes of the best individuals. This is the main problem that triggers the rapid decrease of genetic diversity in the population. Moreover, as in all other meta-heuristic search algorithms, due to the fitness-based update mechanism, all individuals in the population quickly become similar to the best guides. This leads to the problem of premature convergence in the later stages of the search process, when all solution candidates in the population become similar to each other. This strategic failure in the design of EA's guidance mechanism destroys the diversity in the population and causes this algorithm to get caught in local solution traps.

Methods used in the design of the guidance mechanism in FDB-EA: To overcome the problems in the design of EA's guidance mechanism, it is essential to maintain diversity and a balance between exploitation and exploration throughout the search process. This can be achieved by effectively combining guide selection methods that exhibit different search capabilities. In metaheuristic search algorithms, the most commonly used method to achieve diversity is to direct the search through randomly selected guides from the population. The most challenging task in the search process is to establish the exploitation-discovery balance and to maintain this search capability effectively. One of the most recent and effective methods used in the literature to fulfill this challenge is Fitness-Distance Balance (FDB), which was introduced in the previous section. In the FDB method, the exploitation and exploration abilities of guide candidates are represented by the parameters "fitness" and "distance", respectively. Depending on the

Table 6
Mean and standard deviation values for different customer numbers of EA and FDB-EA versions (proposed).

n	**EA**	**Case-1**	**Case-2**	**Case-3**	**Case-4**	**Case-5**	**Case-6**
30	53.96 (0.78)	53.06 (0.63) +	53.10 (0.46) +	53.29 (0.49) +	53.42 (0.81) +	52.99 (0.55) +	53.65 (1.47) +
50	50.27 (1.94)	47.41 (1.14) +	47.23 (0.99) +	47.59 (1.33) +	49.03 (0.89) +	48.84 (0.81) +	48.37 (1.52) +
60	58.77 (3.65)	53.59 (2.11) +	53.35 (2.01) +	54.13 (0.91) +	54.31 (1.12) +	53.94 (1.72) +	54.13 (1.68) +
80	69.20 (3.51)	63.65 (1.33) +	63.36 (1.40) +	63.91 (2.55) +	62.85 (1.35) +	64.25 (2.09) +	64.77 (2.21) +
100	78.07 (5.28)	72.12 (1.83) +	73.73 (1.76) +	71.16 (1.98) +	72.51 (3.31) +	73.58 (2.10) +	74.34 (3.49) +

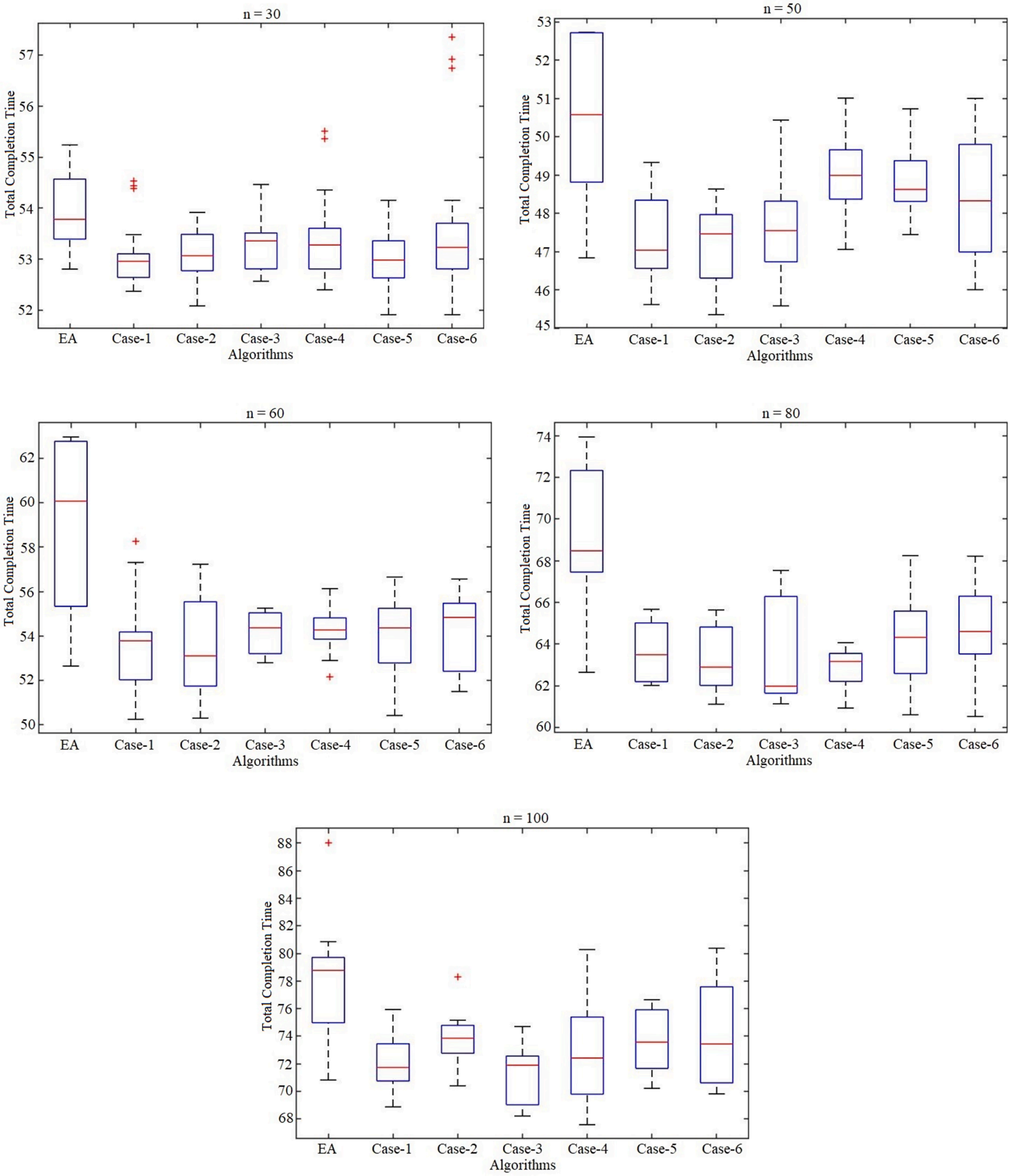

Fig. 3. Box plots of EA and FDB-EA versions for different dimensional routing problems.

Table 7
Friedman scores of algorithms based on the number of customers.

	Algorithms						Mean
1.	FDB-EA	1.76	1.10	1.67	1.00	1.71	1.45
2.	AGDE	3.21	3.83	3.64	3.64	3.26	3.52
3.	BESD	3.60	4.19	3.60	3.48	4.14	3.80
4.	EA	3.52	3.67	4.00	4.43	3.76	3.88
5.	WDE	4.26	3.69	4.24	4.52	4.26	4.20
6.	L-SHADE	4.88	4.52	3.86	3.93	3.86	4.21
7.	ARO	11.00	8.24	7.48	7.33	7.10	8.23
8.	BWO	8.48	7.33	8.81	9.19	9.62	8.69
9.	ABC	8.05	9.00	8.62	8.76	9.10	8.70
10.	SWO	8.62	10.43	9.29	9.14	8.67	9.23
11.	RIME	10.24	12.00	12.24	11.95	11.62	11.61
12.	FDA	14.38	11.86	12.14	11.05	10.95	12.08
13.	CS	11.81	11.43	11.90	13.00	14.90	12.61
14.	POA	15.24	14.14	13.62	14.05	13.24	14.06
15.	KOA	14.62	14.71	15.33	15.38	14.67	14.94
16.	NCHHO	16.86	16.67	16.19	15.19	15.14	16.01
17.	NMPA	14.14	17.62	17.29	17.71	17.71	16.90
18.	WSOA	16.81	17.05	17.14	17.38	17.29	17.13
19.	DAOA	23.57	20.95	20.00	20.14	20.29	20.99
20.	SAO	22.95	22.33	21.00	21.05	20.67	21.60
21.	HOA	22.33	21.71	21.14	22.52	21.81	21.90
22.	RSA	23.24	22.05	22.14	20.67	21.62	21.94
23.	PSO	23.57	22.71	22.52	22.14	22.67	22.72
24.	GJO	24.19	23.38	22.81	23.52	22.67	23.31
25.	CO	20.57	21.38	26.05	27.33	27.90	24.65
26.	EDO	26.05	25.95	25.43	24.33	24.57	25.27
27.	DO	20.14	27.67	27.43	26.57	26.29	25.62
28.	GEO	27.90	26.38	26.43	26.57	26.52	26.76

value of these two parameters, the FDB scores of the solution candidates in the population are calculated, and the ones with the best scores guide the search process. In order to achieve the exploitation-discovery balance in the search process, the "fitness" and "distance" parameters should be equally effective. To equalize the effect of these two parameters, the value of w in Eq. (18) in Algorithm 1 should be set to 0.5.

Based on the above explanations, three different guide selection methods, namely greedy, random and FDB-score-based, are used in the FDB-EA algorithm to provide exploitation, exploration and balanced search capabilities. Parameters a, b and c are defined to hybridize these three methods and adjust their usage rates in the guide selection process. Different strategies are developed by changing the values of parameters a, b and c. These three parameters, the strategies used to develop the hybrid guidance mechanism and the design of the FDB-EA algorithm are described in detail in the following paragraphs.

According to the above explanations, the main steps of the proposed FDB-EA-based routing optimization algorithm are shown in Fig. 2.

Step 1 (TSP-D routing problem): In this step, the decision variables of the TSP-D routing problem and the objective and constraint functions given in Eqs. (1)–(11) are defined.

Step 2 (Proposed hybrid selection mechanism): In this step, depending on the characteristics of the TSP-D routing problem defined in Step 1, a "matrix of solution candidates (population)" is created, and the fitness values of the population members are calculated. After this process, the search process life cycle starts. In the search process life cycle, the first step is to select the guides. When Step 2 is analyzed, greedy, FDB and random selection methods are used in this process depending on the values of parameters a, b and c, respectively, and guides are selected from the population. According to these explanations, the sum of the application rates of these three parameters is $a + b + c = 1$. Since the a-parameter represents the application rate of the fitness-based greedy selection method, a larger value of a than the values of b and c ensures an exploitation-weighted search process. Since the b-parameter represents the application rate of the FDB-based selection method, a larger value of b ensures a balanced search. Since the c-parameter represents the application rate of the random selection method, a larger value of c ensures an exploration-oriented search process. According to these explanations, five different parameter sets consisting of different values of parameters a, b and c were created. Thus, the hybridization rates of greedy, FDB and random selection methods were changed. According to these explanations, five hybrid strategies (HS-1, …, HS-5) with different application rates of parameters a, b and c are presented in Table 2.

Through the creation of different hybridization strategies presented in Table 2, the guidance mechanism of FDB-EA is designed in a dynamic structure. In this dynamic structure, FDB-EA's search process life cycle is divided into three phases and a different hybridization strategy (HS-1, …, HS-5 in Table 2) is applied in each phase. Thus, depending on the phases of the search process, the application rates of the guide selection methods were dynamically changed. Based on these explanations, six different versions of the guidance mechanism of FDB-EA were designed, as presented in Table 3.

Phase-1, *Phase-2* and *Phase-3* in Table 3 represent the first quarter (FEs < MaxFEs/4), the second quarter (MaxFEs/4 < FEs < MaxFEs/2) and the last two quarters (MaxFEs/2 < FEs) of the search process, respectively. According to this explanation, when the design of Case-1, which is the first version of the guidance mechanism of FDB-EA, is examined, the guide selections in the first, second and third phases of the search process life cycle are realized using HS-1, HS-2 and HS-5, respectively (see Table 2 for the definition of HS-1, 2 and 5). According to this design of Case-1, in the first phase of the search process, the guidance process is carried out according to HS-1. It should be considered that according to the information presented in Table 2, the values of the parameters a, b, and c in HS-1 are $a = 0.5$, $b = 0.5$ and $c = 0$, respectively. According to this information, in the Case-1 version, both greedy and FDB methods are applied in the first phase of the search process at a rate of 50 %. In this first phase, the random selection method is not applied at all. This is because the diversity within the population is already high in the first phase of the search process. Therefore, in the Case-1 version of FDB-EA, in the first phase of the search process (*Phase-1*), the HS-1 parameter set is applied (no random selection method is used). In the Case-1 version of FDB-EA, in the second phase of the search process (*Phase-2*), the guidance process is carried out according to the HS-2 set. Considering that the values of the parameters a, b, and c in HS-2 are $a = 0.5$, $b = 0.4$ and $c = 0.1$, respectively, it is understood that the random guide selection in Phase-2 increases by 10 % in contrast to Phase-1. That is, the guide selection strategy in the second phase is 50 % greedy (exploitation), 40 % FDB (exploitation-exploration) and 10 % (exploration). In the Case-1 version of FDB-EA, the guidance process in the last two quarters of the exploration process (*Phase-3*) is carried out according to the HS-5 set. Considering that the values of the parameters a, b, and c in HS-5 are $a = 0.2$, $b = 0.2$ and $c = 0.6$, respectively, it is understood that, unlike the previous two phases, the application rate of random guide selection increases to 60 % in Phase-3. That is, the guide selection strategy in the third phase is 20 % greedy (exploitation), 20 % FDB (exploitation-exploration) and 60 % (exploration). The primary motivation behind this design of Case-1 is to avoid a complete loss of diversity within the population in the final stages of the search process.

In summary, thanks to the hybrid strategies described above and the dynamic guidance mechanism in which these strategies are applied, it has been possible to design different versions of the FDB-EA. In this way, it is possible to dynamically adjust the exploitation, balanced search and exploration capabilities needed at different levels at different stages of the search process. Thus, it was possible to investigate and design a guidance mechanism that best suits the requirements of the search space of the TSP-D routing problem.

According to the above explanations, the pseudo-code of the dynamic guidance mechanism of FDB-EA is presented in Algorithm 2. When the pseudo-code presented in Algorithm 2 is analyzed, depending on the values of the input parameters a, b, and c, guides are selected from the population using greedy, FDB and random methods, respectively. The selected guides are directed to the mutation phase defined in Step 3, as shown in Fig. 2.

Step 3 (Mutation): As shown in the third step in Fig. 2, mutation is

Algorithm-2: Proposed hybrid guiding mechanism

Inputs: $a, b, c, P, F, n, F_{sort}$: sort the solution candidates according to the fitness value, P_{sort}: edit population solution candidate according to F_{sort}

Output: P_{guides} (guides selected from the population)

begin

$P_{size} = 5 * \text{n}$ (P_{size}: number of solution candidates)

for $i = 1{:}\frac{P_{size}}{5} * \boldsymbol{a}$ (select greedily according to the fitness value of solution candidates)

$P_{guides}(i,:) = P_{sort}(i,:)$

end for

for $j = 1{:}\ \frac{P_{size}}{5} * \boldsymbol{b}$ (select according to the FDB selection method)

$P_{guides}(i+j,:)$ → use the FDB selection method given in Algorithm-1

end for

for $k = 1{:}\frac{P_{size}}{5} * \boldsymbol{c}$ (select randomly from solution candidates)

$P_{guides}(i+j+k,:)$ → select a solution candidate from the population P randomly

end for

Table 8
Statistical results of the proposed FDB-EA and the compared algorithms.

Algorithms		$n = 30$	$n = 50$	$n = 60$	$n = 80$	$n = 100$
EA	Mean	53.96	50.27	58.77	69.20	78.07
	Standard Deviation	0.78	1.94	3.65	3.51	5.28
	Minimum	52.80	46.84	52.64	62.65	70.83
	Median	53.78	50.58	60.07	68.47	78.77
	Maximum	55.24	52.73	62.97	73.93	88.04
AGDE	Mean	53.99	50.96	57.68	68.18	77.29
	Standard Deviation	0.87	1.37	2.66	1.94	3.24
	Minimum	52.33	49.01	54.17	66.02	73.01
	Median	54.13	51.18	57.54	67.35	75.77
	Maximum	55.86	54.72	64.16	72.16	85.16
L-SHADE	Mean	54.81	51.67	57.81	68.89	78.05
	Standard Deviation	0.65	1.12	2.72	2.08	3.75
	Minimum	53.34	49.61	54.07	66.25	73.01
	Median	54.84	51.69	57.60	67.71	77.32
	Maximum	56.26	54.72	64.16	72.16	88.23
BESD	Mean	53.96	51.34	57.67	67.70	78.55
	Standard Deviation	0.89	1.09	2.14	1.87	3.05
	Minimum	52.05	49.61	54.41	65.16	73.04
	Median	54.12	51.28	57.60	67.23	79.15
	Maximum	54.99	54.25	62.16	71.32	85.23
WDE	Mean	54.43	51.17	58.55	69.58	79.07
	Standard Deviation	0.68	1.15	2.67	2.63	3.21
	Minimum	53.05	49.03	53.41	65.16	73.96
	Median	54.45	51.17	59.66	69.72	80.23
	Maximum	55.54	53.57	62.82	74.56	85.23
FDB-EA	Mean	53.06	47.41	53.59	63.65	72.12
	Standard Deviation	0.63	1.14	2.12	1.33	1.83
	Minimum	52.37	45.62	50.24	62.02	68.88
	Median	52.96	47.04	53.78	63.49	71.74
	Maximum	54.54	49.33	58.27	65.67	75.94

performed on the selected guides and the population is updated with those that have better fitness values than their competitors. The pseudocode of the mutation phase is presented in Algorithm 3.

Algorithm-3 Pseudocode of the mutation phase

```
   Input: P_guides, P_size, n
   Output: P_new
1  x = 1
2  for i = 1 : P_size/5
3    routeInsertionPoints = sort(ceil(n*rand(1,2)))I = routeInsertionPoints(1)J = routeInsertionPoints(2)
4    for k = 1 : 5
5      tmpPop(k,:) = P_guides (i,:) (tmpPop: matrix holding temporary solution candidates)
6        switch k
7          case 1 (flip)
8            → pass the guide directly to the next generation
9          case 2 (flip)
10           tmpPop(k,I : J) =tmpPop(k,J : −1 : I)
11         case 3 (swap route segments)
12           tmpPop(k,[IJ]) = tmpPop(k,[JI])
13         case 4 (slide route segments down)
14           tmpPop(k,I : J) = tmpPop(k,[I + 1 : JI])
15         case 5 (slide route segments down)
16           if binornd(1,.50)
17             tmpPop(k,[1I]) = tmpPop(k,[I1])
18           else
19             tmpPop(k,[Jend]) = tmpPop(k,[endJ])
20           end
21         otherwise (do nothing)
22       end switch
23     end for
24     P_new(x : x+ 4,:) = tmpPop
25     x = x + 5
26 end for
```

In Algorithm 3, the mutation process is performed with the guides selected in the population. In line 1 of Algorithm 3, the variable x is assigned a value of one. It is used as a matrix index when creating a new population. In line 2, a loop is created with the number of guides selected from the population. In row 3, two random numbers are selected depending on the decision variable n. The smaller value of these two numbers is assigned to variable I, and the larger value is assigned to variable J. In line 4, five loops are defined. In this loop, five offspring solution candidates are created from each guide by mutation. In line 5, the *i-th* guide is assigned to $p(k,:)$. In lines (6–22), the creation of five solution candidates is expressed with the switch-case command. The five solution candidates are saved in the *tmpPop* matrix. In line 8, the selected guide is passed directly to the next generation. In line 10, slip operation is performed depending on variables I and J. In line 12, swap route segments according to variables I and J. In line 14, slide route segments down according to variables I and J. The creation of offspring in lines 17 and 19 depends on the *if* condition in line 16. The command in line 16 states that there is a 50 % chance that line 17 or 19 will be active. Line 24 shows that the new solution candidates are transferred from the *tmpPop* matrix, where they are temporarily kept, to the P_{new} Matrix. Line 25

	Algorithm-4: The pseudo-code of FDB-EA
	Inputs: n, Max_{FEs}
	Output: T_{best}, F_{best}
1	***begin***
2	$F_{best} = \inf$
3	$T_{best} = \text{null}$
4	$P_{size} = 5 * \text{n}$ (P_{size}: number of solution candidates)
5	P: create a random initial population
6	F: calculate the fitness value based on the objective function and constraints given in Eq. (1-11)
7	***while*** $i < Max_{FEs}$ (search process lifecycle)
8	***if*** $\text{i} < \frac{Max_{FEs}}{4}$
9	***Phase-1*** (use parameter sets in Table 2)
10	P_{guides} (use the hybrid guiding mechanism given in Algorithm 2)
11	***end if***
12	***if*** $\frac{Max_{FEs}}{4} < i < \frac{Max_{FEs}}{2}$
13	***Phase-2*** (use parameter sets in Table 2)
14	P_{guides} (use the hybrid guiding mechanism given in Algorithm 2)
15	***end if***
16	***if*** $\frac{Max_{FEs}}{2} < i < Max_{FEs}$
17	***Phase-3*** (use parameter sets in Table 2)
18	P_{guides} (use the hybrid guiding mechanism given in Algorithm 2)
19	***end if***
20	→updating the population with the pseudocode of the mutation process given in Algorithm-3
21	***for*** $j = 1 : P_{size}$
22	DT: assigning customers in P_j to truck or drones
23	→ calculate the fitness value based on the objective function and constraints given in Eq. (1-11)
24	***if*** $F_j < F_{best}$
25	$F_{best} = F_j$ (the best time)
26	$T_{best} = P_j$ (the best route)
27	***end if***
28	$j = j + 1$
29	***end for***
30	$i = i + 1$
31	***end while***
32	***return*** T_{best}, F_{best}

explains that the number of rows of the new population is increased by the number of new solution candidates created in each cycle.

As shown in Fig. 2, the second and third steps of the search process are applied iteratively until the termination condition is met. Once the termination condition is met, the solution candidate with the best fitness value in the population is presented as the optimal solution to the TSP-D routing problem.

All the steps of FDB-EA and its application to the TSP-D problem are presented in the pseudo-code presented in Algorithm-4.

When the pseudo-code of FDB-EA presented in Algorithm-4 is analyzed, default values are assigned for the parameters F_{best} and T_{best} between lines 1–6. Depending on the characteristics of the TSP-D problem, solution candidates are generated, and the fitness values of the solution candidates are calculated using the objective function and constraint functions defined in Eq. (1)–(11). In line 7, the search process life cycle starts. As presented in lines 8–19, firstly, guides are identified in the search process lifecycle. The guide selection is carried out according to the procedure defined in Algorithm 2. After the guide selection, the mutation phase defined by Algorithm 3 is executed in line 20. In lines 21–29, the fitness values of the solution candidates in the population are calculated using the objective function and constraint functions defined in Eq. (1)–(11). Then, the route and the fitness value where the calculated fitness values are less than F_{best} are recorded. Line 30 indicates that the variable i defined in line 7 is incremented by one in each cycle.

5. Experimental study

This section describes the experimental study process. The first purpose of experimental studies is to determine the case that can best solve the problem. Then, a comparison with current evolutionary algorithms is presented with statistical analysis methods. Specific settings are described in the subtitle of this section to provide comparable precision and fairness in experimental studies. Many algorithm variations were tried during the experimental studies, and the best six cases were determined. The simulation results and statistical analysis methods to determine these algorithms are given in another subtitle. In the last subtitle of this section, the best of the six determined cases is taken, and their comparison with the current evolutionary algorithms is presented.

5.1. Settings

A comprehensive experimental study has been carried out to test and validate the search performance of the proposed algorithm. The aim is to demonstrate the performance of solving the TSP-D problem using the proposed hybrid selection mechanism. For this purpose, the number of customers n, an important variable for the TSP-D problem, is taken as 30, 50, 60, 80, and 100. Statistical analysis is applied by considering the experimental study results of the proposed algorithms. In order to carry out the experimental studies in an objective and fair manner, the following procedures are carried out:

i. In the study of the algorithms, the population size was taken as 5*n.
ii. In order to ensure equality of opportunity between the proposed algorithm and the compared algorithms, the objective function is terminated over the maximum number of evaluations.

$$\text{MaxFEs} \equiv \begin{cases} 3{*}10^5 if 10 < n \le 30 \\ 4{*}10^5 if 30 < n \le 50 \\ 8{*}10^5 if 50 < n \le 150 \end{cases} \tag{21}$$

In Eq. (21), MaxFEs is the maximum number of function evaluations allowed, and n is the size of the problem (number of decision variables) (Kumar et al., 2020).

iii. In order to reveal the performance of the proposed method in different dimensional search fields, problems consisting of 30, 50,

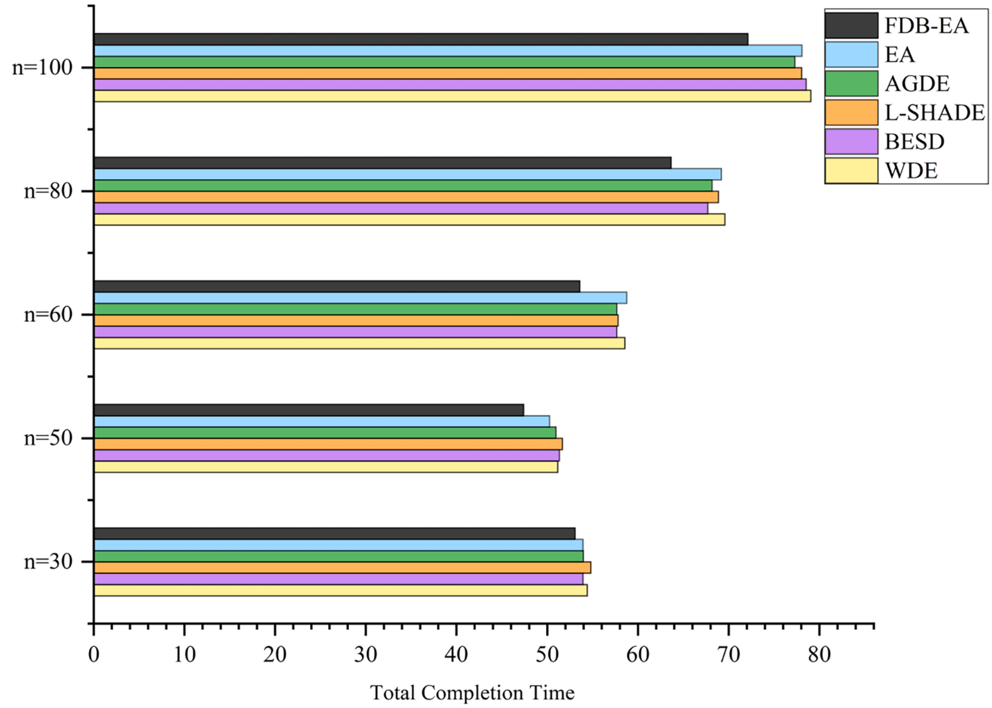

Fig. 4. Comparison of the proposed algorithm with other algorithms.

60, 80, and 100 customer numbers are used (Ham, 2018; Zhang et al., 2020).

iv. The ratio of the drone speed to the truck speed is taken as 1.2 in the problem.

Competitors: The main goal of the experimental study is to search for competitive algorithms that can find optimal solutions for the TSP-D problem. For this purpose, we have chosen the most up-to-date and powerful meta-heuristic search algorithms in the literature as competitors of FDB-EA. These are the competitive evolutionary algorithms we used in the article: EA (Rich, 2020), Adaptive Guided Differential Evolution (AGDE) (Mohamed & Mohamed, 2019), Linear Population Size Reduction Adaptive Differential Evolution (L-SHADE) (Tanabe & Fukunaga, 2014), Bezier Search Differential Evolution (BESD) (Civicioglu & Besdok, 2021), and Weighted Differential Evolution (WDE) (Civicioglu, Besdok, Gunen, & Atasever, 2020).

We also used recently introduced algorithms and well-known meta-heuristics as competitors in the experimental studies. These algorithms are: Artificial Rabbits Optimization (ARO) (Wang et al., 2022), Black Widow Optimization Algorithm (BWOA) (Hayyolalam & Kazem, 2020), Artificial Bee Colony (ABC) (Karaboga & Basturk, 2008), Spider Wasp Optimizer (SWO) (Abdel-Basset & Mohamed et al., 2023), RIME (Su et al., 2023), Flow Direction Algorithm (FDA) (Karami et al., 2021), Cuckoo Search (CS) (Yang and Deb, 2009), Pelican Optimization Algorithm (POA) (Trojovský & Dehghani, 2022), Kepler Optimization Algorithm (KOA) (Abdel-Basset & Mohamed et al., 2023), Nonlinear-based Chaotic Harris Hawks Optimizer (NCHHO) (Dehkordi et al., 2021), Nonlinear Marine Predator Algorithm (NMPA) (Sadiq et al., 2022), War Strategy Optimization Algorithm (WSOA) (Ayyarao & Kumar, 2022), Dynamic Arithmetic Optimization Algorithm (DAOA) (Khodadadi et al., 2022), Smell Agent Optimization (SAO) (Salawudeen et al., 2021), Horse herd Optimization Algorithm (HOA) (MiarNaeimi et al., 2021), Reptile Search Algorithm (RSA) (Abualigah et al., 2022), Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), Golden Jackal Optimization (GJO) (Chopra & Ansari, 2022), Cheetah Optimizer (CO) (Akbari et al., 2022), Exponential Distribution Optimizer (EDO) (Abdel-Basset & Mohamed et al., 2023), Dandelion Optimizer (DO) (Zhao et al., 2022), Golden Eagle Optimizer (GEO) (Mohammadi-Balani et al., 2021). The information on the twenty-seven algorithms used in the comparisons is presented in Appendix.

Parameter settings of competitor algorithms: In the experimental studies, we adopted the Max_{FEs} settings defined in Eq. (21) as the search process termination criterion to ensure fairness among competing algorithms. In the process of determining the parameter settings of competing algorithms, we used the settings recommended by the developers of these algorithms in their article. Thus, the optimization process was carried out fairly for all competitors.

Statistical test methods: Meta-heuristic search algorithms have a non-deterministic search process. Statistical analysis methods are used to analyze the results of simulation studies conducted with these algorithms. There are two widely used non-parametric methods among these analyses. The first one is the Friedman statistical analysis, in which the performance of the algorithms is converted into scores (Friedman, 1940). The other is the Wilcoxon analysis method, which enables pairwise comparison of algorithms (Wilcoxon, 1992).

Suppose we compare k algorithms on N datasets and let R_i represent the average ordinal value of the i th algorithm, then Friedman's test is calculated as follows.

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_i R_i^2 - \frac{k(k+1)^2}{4}\right] \tag{22}$$

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \tag{23}$$

In Eq. (23), the test statistic follows an F distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom under the null hypothesis of equal performance of all algorithms.

To calculate the Wilcoxon test for two dependent samples X_i and Y_i, where $i = 1, \dots, m$, we first calculate the difference between the dependent values, Z_i. Once the differences are calculated, the absolute values of the differences are used to generate the rankings. It is important to pay attention to the original sign of the differences. Below is the pseudo-

Table 9
Friedman analysis scores of the compared algorithms.

	Algorithms					
	EA	AGDE	L-SHADE	BESD	WDE	**FDB-EA**
$n = 30$	3.52	3.12	4.79	3.55	4.26	1.76
$n = 50$	3.67	3.83	4.52	4.19	3.69	1.10
$n = 60$	4.00	3.64	3.86	3.60	4.24	1.67
$n = 80$	4.43	3.64	3.93	3.48	4.52	1.00
$n = 100$	3.76	3.26	3.86	4.14	4.26	1.71
Mean	3.88	3.50	4.19	3.79	4.20	1.45

Table 10
Wilcoxon pairwise comparison of competing algorithms versus FDB-EA algorithm.

Algorithms					
	EA	**AGDE**	**L-SHADE**	**BESD**	**WDE**
vs. FDB-EA $+/=/-$	0/0/5	0/0/5	0/0/5	0/0/5	0/0/5

code of the Wilcoxon statistical analysis method in Algorithm-5.

Algorithm-5: Steps of Wilcoxon statistical analysis method (Wilcoxon, 1992)

```
start
for i = 1 : functionsNumber
  for k = 1 : experimentNumber
    for m = 1 : algorithmsNumber
      result(algorithmsNumber*(i−1) + m,6*k−5) = min(solution(algorithmsNumber*(k−1) + m,i,:));
      result(algorithmsNumber*(i−1) + m,6*k−4) = mean(solution(algorithmsNumber*(k−1) + m,i,:));
      result(algorithmsNumber*(i−1) + m,6*k−3) = std(solution(algorithmsNumber*(k−1) + m,i,:));
      result(algorithmsNumber*(i−1) + m,6*k−2) = median(solution(algorithmsNumber*(k−1) + m,i,:));
      result(algorithmsNumber*(i−1) + m,6*k−1) = max(solution(algorithmsNumber*(k−1) + m,i,:));
    end for
    for m=1 : algorithmsNumber −1
      for n = 1 : run
        firstWilcon(n) = solution(algorithmsNumber*(k−1) + 1,i,n)
        secondWilcon(n) = solution(algorithmsNumber*(k−1) + m + 1,i,n)
      end for
      [p,h,stats] = ranksum(firstWilcon,secondWilcon)
      wilcon = 0
      if (h)
        if (p < 0.05)
          if (stats.zval < 0)
            wilcon = 2
          else
            wilcon = 1
          end if
        end if
      end if
      result(algorithmsNumber*(i−1) + m + 1,6*k) = wilcon
    end for
  end for
end for
end start
```

5.2. Simulation results and statistical analysis

Simulation results are presented in two subsections. In the first sub-section, the performances of the FDB-EA versions introduced in the method section on TSP-D are investigated. As a result of this research, the most effective FDB-EA version for TSP-D is determined. In the second sub-section, the performances of FDB-EA and current and powerful competing algorithms in the literature on TSP-D are compared using nonparametric statistical test methods. According to the results of the statistical analysis, the algorithms with competitive performance on TSP-D are identified, and their stability and computational complexity are analyzed.

5.2.1. Investigating the performance of FDB-EA versions on TSP-D

In this section, firstly, the convergence performance of the FDB-EA versions introduced in the methodology section and presented in Table 3 on TSP-D is investigated. This section aims to identify the most efficient search performance on TSP-D among the six different FDB-EA versions.

The optimum routing results obtained by performing twenty-five runs of the proposed FDB-EA versions were subjected to Friedman and Wilcoxon analysis. Table 4 shows the scores of the Friedman analysis, and Table 5 shows the Wilcoxon analysis results.

The Friedman scores obtained by EA and six different FDB-EA versions for five different TSP-D problems are presented in Table 4. The last row of Table 4 presents the mean scores obtained by the algorithms for the TSP-D problems. According to these explanations, the best Friedman score obtained for each TSP-D problem is shown in bold font in Table 4. According to the scores presented in Table 4, all FDB-EA versions (Case-1, …, Case-6) were more successful than EA in finding optimal routes in all TSP-D problems (n = 30, …, 100). This result is clear proof that the hybrid guidance mechanism introduced with FDB-EA is successful. A comparison between the FDB-EA versions shows that the first three versions (Case-1, Case-2, and Case-3) exhibit superior search performance compared to their competitors. To understand the reason for this, it is necessary to examine the designs of the guidance mechanisms of the FDB-EA versions presented in Tables 2 and 3 in the Method section. When the designs of the guidance mechanisms of the first three versions of FDB-EA are examined, it is seen that the random selection method is used at a very low rate in the early stages of the search process, while the random selection method is used at a high rate in the last stages of the search process. This situation made greedy and FDB-based selection methods more effective in the early stages of the search process and ensured that exploitation-exploration capabilities could be maintained in a balanced manner. Because diversity in the population is high in evolutionary search algorithms when solution candidates are first created (in the early stages of the search process), it is not recommended to use the random selection method, which increases diversity, at a high rate in the early stages of the search process. Case-4, Case-5 and Case-6 were more unsuccessful among the FDB-EA versions compared to their competitors. When the designs of the guidance mechanisms of these three FDB-EA versions are analyzed, it is understood that the random selection method is applied at a higher rate in the early stages of the search process compared to its competitors. This led to the inadequate exploitation capabilities of these three FDB-EA versions compared to their competitors, and it is understood that these three FDB-EA versions could not maintain the exploitation-exploration balance as successfully as other versions.

Using the data obtained from the simulations for five different TSP-D problems, pairwise comparisons were made between the EA algorithm and the versions of FDB-EA. The nonparametric Wilcoxon pairwise comparison results are presented in Table 5.

According to Table 5, all six versions of FDB-EA outperform the EA in pairwise comparison. Better results were also obtained in the problem dimensions with 30, 50, 60, 80, and 100 customers. According to the pairwise comparison results, even the versions of FDB-EA with poor guidance mechanisms can exploit-discover TSP-D's search space more successfully than EA. This is mainly due to the fact that all versions of FDB-EA use the FDB guide selection method and apply different guidance strategies at different stages of the search process. In summary, the methods and strategies applied in the design of FDB-EA have been successful.

The mean objective function values and standard deviation values for different customer numbers of the proposed FDB-EA versions and EA are given in Table 6. In comparing algorithms with FDB-EA, superiority equal or loss is given by $-$, $+$, and $=$ signs.

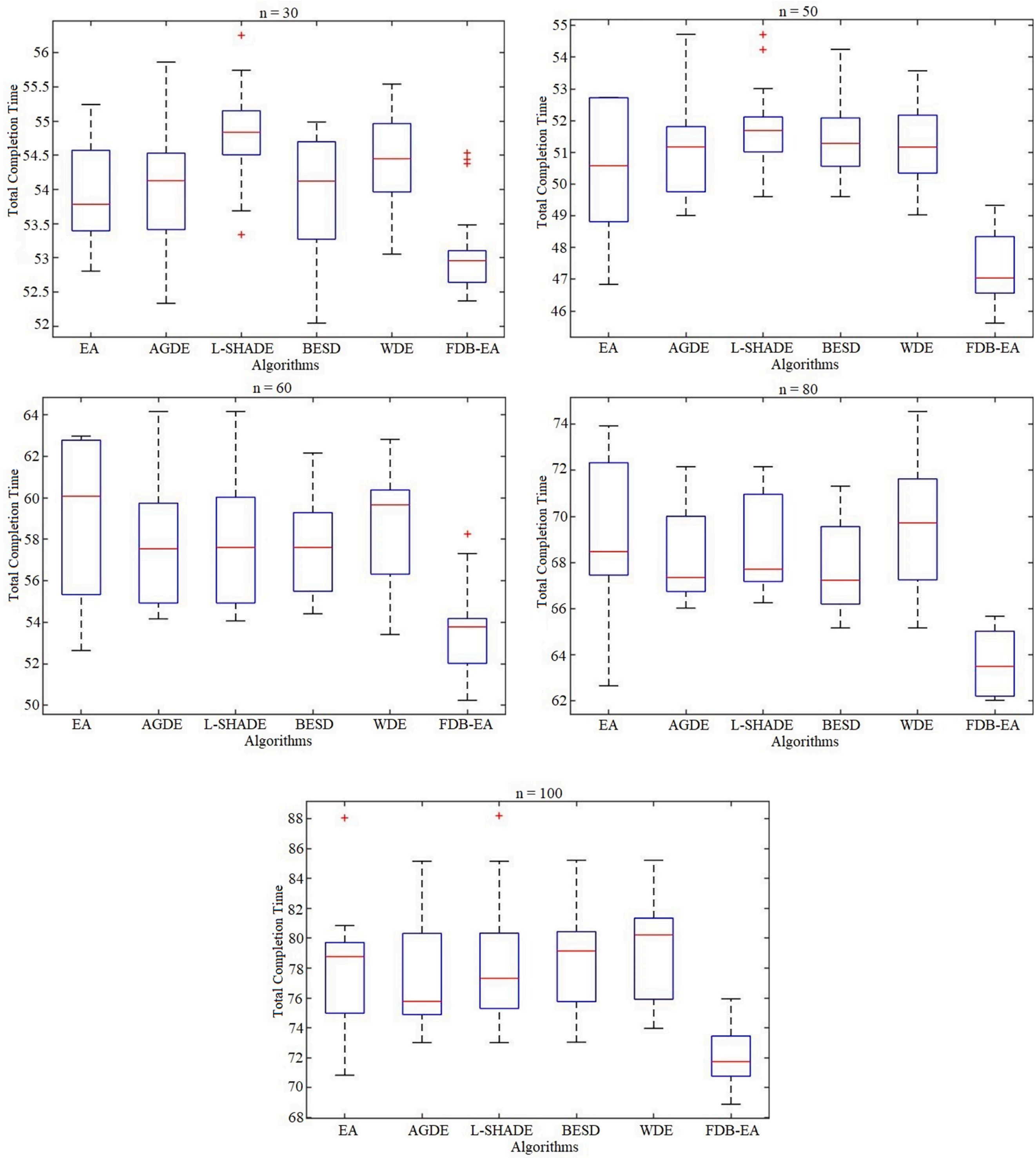

Fig. 5. Box plot comparisons of competing algorithms for different-sized problems.

When Table 6 is examined line by line, it is clearly seen that FDB-EA versions obtain better routing times than EA for all TSP-D problems. According to these results, the EA algorithm gets stuck in the local solution without finding the optimum routes. FDB-EA versions, on the other hand, managed to avoid the local solution traps in the search space of TSP-D, thanks to the superior exploration ability and exploitation-exploration balance provided by the guidance mechanism.

One of the most effective methods used to visualize the performance of evolutionary search algorithms is to display numerical data obtained from simulation studies through box-plots. When the box plot in Fig. 3 is examined, the optimum routing times of the EA and the FDB-EA versions presented in the experimental study with twenty-five run results are given in the box plot.

When the box plots of the EA algorithm in Fig. 3 are examined, it is seen that there are large differences between the best and worst values of routing completion times. This clearly shows that EA has low stability in optimizing TSP-D problems. The lower edges of the box plots correspond to the best routing times found by the algorithms, while the upper edges

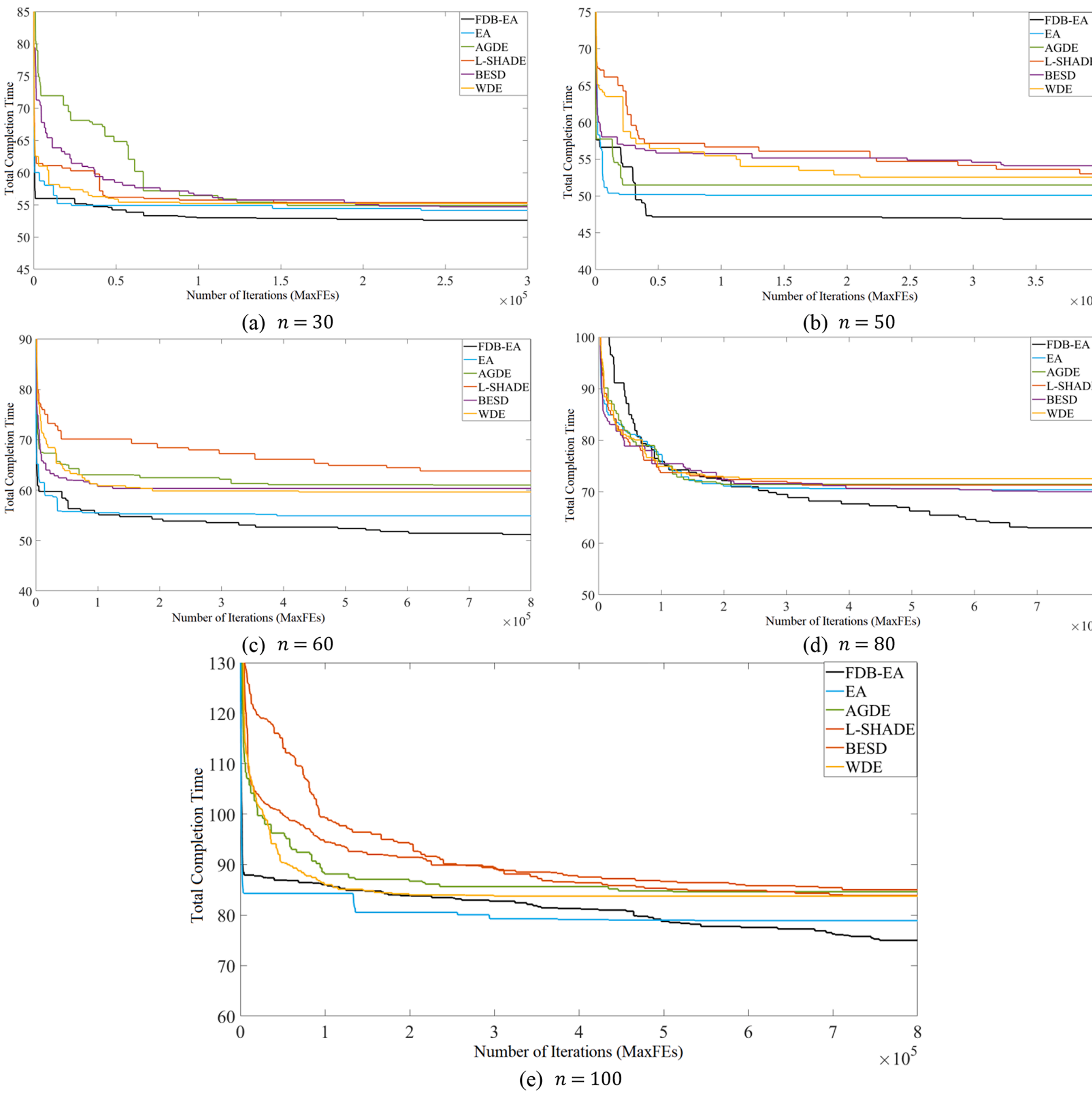

Fig. 6. Convergence curves of competing algorithms for different-sized problems.

Table 11
Feasible solutions for problems with different customer numbers.

n = 30	n = 50	n = 60	n = 80	n = 100
53.67	49.54	56.67	67.01	75.83

of the box plots correspond to the worst routing times found by the algorithms. According to this explanation, the worst routing completion times found by FDB-EA versions are even better than the best ones found by EA. In summary, all FDB-EA versions (Cases 1, …, 5) appear to have better performance than the EA algorithm. This situation reveals that solution candidates to be transferred to new generations in the search process life cycle are effectively selected. It can be said that Case 1 gives more effective results than other cases in general.

5.2.2. Comparison with competitors

This section presents a comparison of the best FDB-EA version (Case-1) proposed in the above section with 27 different well-known and up-to-date competing MHS algorithms in the literature. Simulation studies were carried out based on fairness between algorithms and by adopting the standards in the literature. For simulation settings, please read the "5.1 Settings" section. It was run twenty-five times on five different customer sizes and ranked according to Friedman's statistical analysis scores. The Friedman scores of the algorithms in the five problem dimensions are presented in Table 7. In the ranking table for each customer number, the colors green, yellow and red represent the first, second and third algorithms, respectively. The other algorithms in the top ten are indicated in blue.

When Friedman scores are analyzed according to different customer dimensions, the proposed FDB-EA algorithm outperforms all the

Table 12
SR, *AFES*, *AD* performances of algorithms.

Number of customers	SR (%)			AFES			AD (sec)		
	FDB-EA	EA	AGDE	FDB-EA	EA	AGDE	FDB-EA	EA	AGDE
$n = 30$	84	72	32	44771.10	55444.28	170720.25	532.00	761.44	13957.13
$n = 50$	96	68	4	108939.54	81844.41	75031.00	5827.88	2027.35	32599.00
$n = 60$	84	60	12	125287.62	150213.40	112102.00	3857.52	4433.53	26169.00
$n = 80$	92	52	12	107783.61	128969.46	234577.67	10013.70	11377.62	48125.33
$n = 100$	84	40	8	112750.14	182838.50	188291.50	11312.86	18287.20	52470.50
Average	88.00	58.40	13.60	99906.40	119862.01	156144.48	6308.79	7377.43	34664.19

compared algorithms and ranks first. For problems with 30 and 100 customers, the second and third algorithms are AGDE and EA algorithms, respectively. For the 50 customers problem, the second and third algorithms are EA and WDE, respectively. For the 60 and 80 customer problems, the second and third algorithms are AGDE and BESD algorithms, respectively. The L-SHADE algorithm gave closer results to the other algorithms in high customer number problems than in low customer number problems. In terms of the overall mean, the three algorithms are FDB-EA, AGDE and BESD algorithms, respectively. Friedman scores presented in Table 7 statistically prove that FDB-EA exhibits superior search performance compared to its competitors for all TSP-D problems. This shows that competing algorithms cannot avoid local solution traps in the search space of the TSP-D problem and experience premature convergence problems because the requirements of the search space of the TSP-D problem were not taken into account in the design of the guidance mechanism in competing algorithms.

Moreover, an FDB-based guide selection method and dynamically switching hybrid strategies were not used in the design of the guidance mechanisms of the competing algorithms. Unlike competing algorithms, FDB-EA has all of these features in the design of the guidance mechanism. Thanks to these features, FDB-EA was able to successfully maintain the exploitation and exploration capabilities required by the TSP-D problem during the search process and discovered the best routes.

A detailed comparison between FDB-EA and evolutionary algorithms (AGDE, L-SHADE, BESD, WDE) is presented in Table 8. Experimental results were obtained by running the algorithms independently for twenty-five runs. Experimental studies were conducted following the experimental settings in the problems of 30, 50, 60, 80, and 100 customer numbers. As a result of the experimental study, the mean objective function value and standard deviation values obtained from twenty-five runs of the compared algorithms are given in Table 8 below.

The FDB-EA algorithm proposed according to Table 8 showed superior performance to the EA and other evolutionary algorithms. The results of the objective function values obtained from twenty-five runs are given. The mean, standard deviations, minimum values, median values, and maximum values of the results obtained in minimizing the completion time of the job in the delivery problem are given in Table 8. The FDB-EA algorithm, given in bold type, gave effective results in 5 different customer number dimensions compared to other algorithms. In Fig. 4, the solution values obtained in the solution of the optimization problem consisting of 30, 50, 60, 80, and 100 customer numbers are given as a bar plot.

According to Fig. 4, the vertical axis is the number of customers, and the horizontal axis gives the mean time it has completed the delivery operation in twenty-five runs. The FDB-EA algorithm given in Bold Fig. 4 has optimized the minimization problem better than other algorithms. In Table 9, the results of the algorithms according to the Friedman statistical analysis are given.

According to Table 9, the proposed FDB-EA algorithm gave better results than other algorithms. The proposed FDB-EA algorithm in all problems with 30, 50, 60, 80, and 100 customers performed better than the other algorithms by showing a balanced search process. The algorithm that achieves the best results in all problem dimensions is in bold in Table 9. Although there are problems where other evolutionary algorithms are superior to the EA, it has been seen that they are defeated in some problems. In Table 10, the results of the Wilcoxon pairwise comparison of the compared algorithms with the proposed FDB-EA algorithm are given.

When examined according to Wilcoxon scores given in Table 10, the proposed FDB-EA algorithm showed superior performance compared to the compared algorithms in all five problems consisting of 30, 50, 60, 80, and 100 customers. Below are the box plots of the proposed algorithm and other compared algorithms in Fig. 5.

In Fig. 5, box plots of the proposed algorithm FDB-EA and the compared algorithms are given. All algorithms in Fig. 5 were run for twenty-five runs for all problems, and box plots of the results were presented. The proposed algorithm in all problem dimensions gave very successful results compared to the compared algorithms. If other evolutionary algorithms are compared among themselves, it can be said that they obtain results close to each other. With the proposed FDB-EA algorithm, the total completion time of the job in truck-drone synchronous package delivery is more successful than that of other algorithms. It can be deduced that this mechanism, which guides the search process, will facilitate the solution of combinatorial problems. Moreover, it can be seen from the box plots that the proposed algorithm is more stable than the other compared algorithms in terms of stability. A more detailed stability analysis is carried out in section 5.4 for stability analysis.

The convergence curves of the compared algorithms for the five samples are shown in Fig. 6. In the preparation of the convergence curves, the maxFEs value for all algorithms was set to the value given in Eq. (21) as the criterion for terminating the search process. This setting is a standard termination criterion recommended and adopted in reputable scientific publications in the literature. Moreover, the termination value assigned to maxFEs is the largest value used in the literature to demonstrate the performance of the algorithms. For detailed information, please review the explanations in the “5.1. settings” section and the references (Kumar et al., 2020; Liang et al., 2019).

When the convergence curves in Fig. 6 are examined, the fact that there has been no change in these curves for a long time shows that the algorithms cannot improve the routing time and are caught in a local solution. The breaking points in the convergence curves indicate that the algorithms avoid local solution traps by providing diversity in the search process and thus find better solutions. When the convergence curves in Fig. 6 are examined based on these explanations, it is seen that all of the algorithms converge to a constant value when the last quarter of the search process is reached. This shows that the value assigned for the search process termination criterion (MaxFEs) is sufficient to demonstrate the search performance of the algorithms. When the convergence curves of the algorithms are compared, it is seen that there are more break points in the convergence curve of FDB-EA compared to its competitors, starting from the first quarter of the search process. The reason for this situation is the dynamic structure of FDB-EA’s guidance mechanism. As detailed in the “4.2 Proposed FDB-EA algorithm” section, FDB-EA uses a different hybrid guidance strategy in each quarter of the search process. Thanks to this feature, the FDB-EA algorithm is more

Table A1
Algorithm Settings.

Algorithm	Parameters	Values
AGDE	Population size (NP)	50
	p	0.1
BESD	Population size	K*N
	K	5
WDE	Size of population	30
EA	Population size	5*D
LSHADE	Population size (N)	18*D
	Memory size (H)	5
	Archive rate	1.4
	Pbest individuals rate (p)	0.11
SWO	Number of search agents	100
	Crossover Probability (CR)	0.2
	Trade-off probability between hunting and mating behaviours (TR)	0.3
CO	Population size (N)	6
	Number of search agents in a group (m)	2
ARO	Population size	50
RIME	Population size (N)	30
ABC	Colony size	50
	Number of food sources (SN)	Colony size/2
	Limit	D*SN
DO	Population size	30
KOA	Number of search agents	100
	Tc	3
	M0	0.1
	Lambda	15
NMPA	Number of search agents	25
	FADs	0.2
	P	0.5
CS	Number of Host Nests	25
	pa	0.25
FDA	Number of flows (Alpha)	50
	Number of neighborhoods (Beta)	1
POA	Search Agents	30
NCHHO	Population Size (N)	100
PSO	Swarm size	30
	Cognitive constant	2
	Social constant	2
	Decreasing Inertia weight	Linearly reduction from 0.9 to 0.4
GJO	Number of search agents	30
	Decreasing energy of the prey (E1)	Linearly reduction from 1.5 to 0
	Random number (r)	$r \in [0, 1]$
	Constant value (c1)	1.5
WSOA	Number of Soldiers	30

(*continued on next page*)

Table A1 (*continued*)

Algorithm	Parameters	Values
EDO	Population Size (NP)	30
BWO	Population Size	50
	Procreate Rate	0.6
	Mutation Rate (PM)	0.4
	Cannibalism Rate	0.44
DAOA	Population Size (N)	5
	Mü	0.001
	Alpha	25
HOA	Number of Horses	50
	W	1
	Phi D	0.02
	Phi L	0.02
	Grazing (g_Alpha)	1.5
	Defense Mechanism (d_Alpha)	0.5
	Hierarchy (h_Alpha)	1.5
	Grazing (g_Beta)	1.5
	Hierarchy (h_Beta)	0.9
	Sociability (s_Beta)	0.2
	Defense Mechanism (d_Beta)	0.2
	Grazing (g_Gamma)	1.5
	Hierarchy (h_Gamma)	0.5
	Sociability (s_Gamma)	0.1
	Imitation(i_Gamma)	0.3
	Defense Mechanism (d_Gamma)	0.1
	Random (Wandering and Curiosity) (r_Gamma)	0.05
	Grazing (g_Delta)	1.5
	Random (Wandering and Curiosity) (r_Delta)	0.1
RSA	Population Size (N)	10
	Alpha	0.1
	Beta	0.005
GEO	Population Size	50
	Attack Propensity	$linspace(0.5, 2, MaxIterations)$
	Cruise Propensity	$linspace(1, 0.5, MaxIterations)$
SAO	*nMole*	50
	Olf	0.9
	K	0.6
	T	0.95
	M	0.9
	Step	0.02

successful than its competitors in providing diversity and avoiding local solution traps. The success of FDB-EA in finding optimal routes is clearly seen from the convergence graphs obtained for five different TSP-D problems.

In summary, the proposed FDB-EA algorithm converged to lower values with fast convergence in all considered problem dimensions. The experimental results show the superior performance of FDB-EA over other comparing algorithms in solving the considered problem.

5.3. Stability analysis

The optimal routes found by the algorithms for the TSP-D problems were analyzed in sections 5.2.1 and 5.2.2. Statistical analyses and comparisons were carried out on the optimal routes found by the algorithms. The analysis results presented in these two sections do not provide information about the computational complexity and stability of the algorithms. In this section, the stability of the algorithms is analyzed. The purpose of stability analysis is to investigate (i) success rates and (ii) search times of algorithms in finding feasible solutions for problems. Thus, it is aimed to analyze and compare the stability and computational complexity of algorithms. These explanations, definitions and parameters used in stability analysis are explained below.

- Feasible solution: Algorithms provide the solution to the problem depending on the number of customers. By running the algorithms twenty-five times, a feasible solution to the problem is expected to be found. Three algorithms (FDB-EA, EA and AGDE) with the best performance among six competing algorithms were considered for the feasible solution. The mean of the completion times of the delivery operations of the algorithms given in Table 7 is taken as a reference.
- Success rate: It is saved whether the algorithms find a feasible solution as a result of each study. The ratio of the number of studies with a feasible solution as a result of twenty-five runs to the total number of runs gives the success rate, *SR*.
- Average number of fitness evaluations: The ratio of the number of fitness evaluations in which the algorithms find each feasible solution to the runs that find the solution gives the average number of fitness evaluations, *AFES*.

- Average duration: Similar to the *AFES* parameter, the average of the durations found by the feasible solution gives the average duration, *AD*.

According to the mean values of FDB-EA, EA and AGDE algorithms given in Table 7, feasible solution values for five different customer numbers were determined. These feasible solution values are given in Table 11.

In stability analysis, algorithms are expected to find feasible solutions for problems of different difficulty. The value of the *SR* parameter is given in Eq. (24). Here, the number of runs reaching a feasible solution is expressed by the *nsr* parameter (Duman et al., 2021).

$$SR = \frac{numberofsuccessfulruns(nsr)}{numberoftotalruns} x100 \tag{24}$$

The number of fitness evaluations *fes* and elapsed time *et* when the algorithms find the feasible solution are recorded. The *AFES* parameter is determined by Eq. (25) (Duman et al., 2021).

$$AFES = \frac{1}{nsr}\sum\nolimits_{j=1}^{nsr} fes_j \tag{25}$$

Eq. (26) is used to calculate the *AD* parameter (Duman et al., 2021).

$$AD = \frac{1}{nsr}\sum\nolimits_{j=1}^{nsr} AD_j \tag{26}$$

The pseudo code for *SR*, *AFES* and *AD* parameters is given in Algorithm-6.

Algorithm-6: Pseudo-code for calculation of SR, AFES and AD (Duman et al., 2021)

fs: feasible solution to the problem
for $i = 1 : run$
  while $fes < Max_{FEs}$
    start time, exploitation, exploration and updating process
    if *solution* $\leq fs$ (is the solution found by the algorithm smaller than *fes*?)
      save the elapsed time et_i and current value of fes_i
      $nsr = nsr + +;$
  break while
    end if
  end while
end for
return nsr, fes_i, et_i (Calculate the *SR*, *AFES* and *AD* using Eq. (24), 25 and 26)

Algorithm-6 shows the pseudo-code for the *nsr*, *fes* and *et* parameters required to calculate the *SR*, *AFES* and *AD* parameters. *SR*, *AFES* and *AD* parameters obtained from running the algorithms are given in Table 12.

The *SR*, *AFES* and *AD* parameters of the compared algorithms are given in Table 12. Table 12 shows the superiority of the FDB-EA algorithm over the other two algorithms in the problem with five different numbers of customers in the *SR* column. FDB-EA algorithm showed a stable performance compared to competitor algorithms in finding the feasible solution with 84 % to 96 %. The EA algorithm with an *SR* value of 40 % to 72 % shows lower stability than the FDB-EA algorithm. The stability performance of the AGDE algorithm, which has an *SR* value of 4 % to 32 %, is quite poor compared to the other two algorithms. The fact that the AGDE algorithm cannot find a feasible solution indicates that it is stuck in local traps. *AFES* and *AD* columns have been created by taking into account the successful runs in which algorithms have reached a feasible solution. When the *AFES* column is examined, it indicates that the FDB-EA algorithm finds a feasible solution with less effort than other algorithms in all customer numbers except n = 50 customers. Similar to the *AFES* parameter, the FDB-EA algorithm has higher performance in the *AD* parameter. It is seen that it has reached a feasible solution in a short time compared to its competitors. The computational complexity of the FDB-EA algorithm is higher than the EA algorithm in a fitness evaluation. FDB-EA reaches the feasible solution in a shorter time than the EA algorithm due to the number of fitness evaluations in which it finds the feasible solution. The *SR* parameter value does not significantly reflect the *AFES* and *AD* parameters of the AGDE algorithm, which is very low.

6. Conclusion and future research

This paper introduced a powerful FDB-EA algorithm to optimize synchronized truck and drone routing problems in package delivery logistics. The results from the research and simulation studies conducted in this article are as follows:

i) *In studies on TSP-D, computational complexity and stability analysis information of algorithms is needed.* The main shortcomings in research on the optimization of synchronized truck-drone delivery problems are the computational complexity of the algorithms and the fact that stability analysis information is not presented. Stability analysis and computational complexity information must be investigated to compare the performances of competing algorithms in studies on the optimization of routing problems. In this paper, the information on stability analysis presented in section 5.3- “Stability analysis” can be taken as a reference for future research on TSP-D.
ii) *The search operators of meta-heuristic algorithms should be designed according to the requirements of the problem:* In synchronized truck-drone delivery problems, trucks and drones are vehicles with different constraints and features. Research should be conducted on improving the design of convergence operators of metaheuristic search algorithms in routing studies on the synchronization of these two vehicles. According to the results of the statistical analysis, research on the design of the guide selection mechanism affects the search performance of the algorithms. The only difference between the FDB-EA and EA algorithms is the design of the guidance mechanisms of these two algorithms. Thanks to the guiding mechanism designed using FDB score-based and search phase-based dynamic strategies, FDB-EA has demonstrated a superior exploration and exploitation ability compared to EA and all other competitors. According to Wilcoxon pairwise comparison analysis results between the FDB-EA algorithm and EA algorithm, the FDB-EA algorithm outperformed its competitor in all 5-different TSP-D problems with $n = \{30, 50, 60, 80, 100\}$. Moreover, the FDB-EA algorithm showed superior search performance in finding optimal routes for five different TSP-D dimensions compared to other recent and powerful evolutionary-based competing algorithms. According to the results in the experimental study section, the method used in designing the guidance mechanism of FDB-EA has provided a remarkable improvement in providing the constraints and finding the optimum route in synchronized truck-drone delivery problems.

In summary, many papers have been reported in the literature on the optimization of TSP-D problems. Unlike these papers, the FDB-EA algorithm, in which the search operators were explicitly designed for the problem, is successfully introduced in this paper. The search performance of FDB-EA was verified by non-parametric statistical analysis results. The procedures used in the design of the guidance mechanism of FDB-EA increased the computational complexity of the algorithm. As a result, the computational complexity of FDB-EA was found to be higher compared to its competitors.

On the other hand, it was observed that FDB-EA displayed a more stable search performance than its competitors in synchronized truck-drone delivery problems. Moreover, FDB-EA needed fewer iterations to discover the optimal route than its competitors, even though, on average, its computational complexity in one iteration was higher than its competitors. This enabled FDB-EA to find feasible solutions for TSP-D problems in less time than its competitors. Moreover, the routes found by FDB-EA for five different TSP-D problems were 5 % better than those found by all competitors. This is evident from the data presented in Table 8 and clearly seen in the box plots in Fig. 5. It is hoped that the guidance mechanism and the FDB-EA algorithm introduced in the 4.- “Method” section will be a reference for future research on TSP-D.

Future works are presented below:

- The FDB-EA algorithm outperforms all competing algorithms in all five different TSP-D problems due to its guidance mechanism. This mechanism is designed using hybrid strategies and has a dynamic functioning depending on the phases of the search process life cycle. In the future, the guidance mechanism is planned to be applied in different types of meta-heuristic search algorithms.

- The results of the Wilcoxon pairwise comparison test show that the FDB-EA algorithm outperforms its competitors on all TSP-D problems. This success motivates the authors to apply the FDB-EA algorithm to other types of TSP-D problems in the future.

- Due to the time-consuming nature of simulation studies, this article investigates optimal routes for TSP-D problems with five different delivery point numbers ($n = \{30,50,60,80,100\}$). As shown in Table 12 of the stability analysis section, while the success rates of competing algorithms decrease with increasing delivery point number, the success rate of FDB-EA remains relatively stable. Therefore, the plan is to investigate the performance of FDB-EA on large-scale TSP-D problems in the future.

- In the TSP-D problem, one truck is considered. Using more than one truck in the distribution system will enable delivery to customers in a wider area. In addition, this study tries to optimize the TSP-D problem. Future studies aim to investigate the performance of the proposed FDB-EA in FSTSP or PDSTSP problems besides TSP-D.

7. Code sharing

Source codes: https://www.mathworks.com/matlabcentral/fileexchange/156906-fdb-ea-a-new-routing-algorithm-and-tsp-d-problem-codes

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A

References

Abdel-Basset, M., Mohamed, R., Azeem, S. A. A., Jameel, M., & Abouhawwash, M. (2023). Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowledge-Based Systems, 268*, Article 110454. https://doi.org/10.1016/j.knosys.2023.110454

Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications, 191*, Article 116158. https://doi.org/10.1016/j.eswa.2021.116158

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science, 52*(4), 965–981. https://doi.org/10.1287/trsc.2017.0791

Akbari, M. A., Zare, M., Azizipanah-Abarghooee, R., Mirjalili, S., & Deriche, M. (2022). The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems. *Scientific Reports, 12*(1), Article 10953. https://doi.org/10.1038/s41598-022-14338-z

Arishi, A., Krishnan, K., & Arishi, M. (2022). Machine learning approach for truck-drones based last-mile delivery in the era of industry 4.0. *Engineering Applications of Artificial Intelligence, 116*, Article 105439. https://doi.org/10.1016/j.engappai.2022.105439

Ayyarao, T. S., & Kumar, P. P. (2022). Parameter estimation of solar PV models with a new proposed war strategy optimization algorithm. *International Journal of Energy Research, 46*(6), 7215–7238. https://doi.org/10.1002/er.7629

Baik, H., & Valenzuela, J. (2021). An optimization drone routing model for inspecting wind farms. *Soft Computing, 25*(3), 2483–2498. https://doi.org/10.1007/s00500-020-05316-6

Boccia, M., Masone, A., Sforza, A., & Sterle, C. (2021). A column-and-row generation approach for the flying sidekick traveling salesman problem. *Transportation Research Part C: Emerging Technologies, 124*, Article 102913. https://doi.org/10.1016/j.trc.2020.102913

Bouman, P., Agatz, N., & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks, 72*(4), 528–542. https://doi.org/10.1002/net.21864

Bruni, M. E., Khodaparasti, S., & Moshref-Javadi, M. (2022). A logic-based Benders decomposition method for the multi-trip traveling repairman problem with drones. *Computers & Operations Research, 145*, Article 105845. https://doi.org/10.1016/j.cor.2022.105845

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies, 54*, 86–109. https://doi.org/10.1016/j.trc.2015.03.005

Chen, Y. S., Chu, H. H., & Sangaiah, A. K. (2019). Identifying ambient service location problems and its application using a humanized computing model. *Journal of Ambient Intelligence and Humanized Computing, 10*(6), 2345–2359. https://doi.org/10.1007/s12652-018-0838-8

Choi, Y., & Schonfeld, P. M. (2017, January). Optimization of multi-package drone deliveries considering battery capacity. In *Proceedings of the 96th Annual Meeting of the Transportation Research Board, Washington, DC*, USA (pp. 8-12).

Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications, 198*, Article 116924. https://doi.org/10.1016/j.eswa.2022.116924

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research, 123*, Article 105004. https://doi.org/10.1016/j.cor.2020.105004

Civicioglu, P., Besdok, E., Gunen, M. A., & Atasever, U. H. (2020). Weighted differential evolution algorithm for numerical function optimization: A comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. *Neural Computing and Applications, 32*, 3923–3937. https://doi.org/10.1007/s00521-018-3822-5

Civicioglu, P., & Besdok, E. (2021). Bezier Search Differential Evolution Algorithm for numerical function optimization: A comparative study with CRMLSP, MVO, WA, SHADE and LSHADE. *Expert Systems with Applications, 165*, Article 113875. https://doi.org/10.1016/j.eswa.2020.113875

Dayarian, I., Savelsbergh, M., & Clarke, J. P. (2020). Same-day delivery with drone resupply. *Transportation Science, 54*(1), 229–249. https://doi.org/10.1287/trsc.2019.0944

Dehkordi, A. A., Sadiq, A. S., Mirjalili, S., & Ghafoor, K. Z. (2021). Nonlinear-based chaotic Harris Hawks optimizer: Algorithm and internet of vehicles application. *Applied Soft Computing, 109*, Article 107574. https://doi.org/10.1016/j.asoc.2021.107574

Deng, Y., Zheng, Y., & Li, J. (2019). Route optimization model in collaborative logistics network for mixed transportation problem considered cost discount based on GATS. *Journal of Ambient Intelligence and Humanized Computing, 10*(1), 409–416. https://doi.org/10.1007/s12652-018-0749-8

de Freitas, J. C., & Penna, P. H. V. (2020). A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research, 27*(1), 267–290. https://doi.org/10.1111/itor.12671

Duman, S., Akbel, M., & Kahraman, H. T. (2021). Development of the multi-objective adaptive guided differential evolution and optimization of the MO-ACOPF for wind/PV/tidal energy sources. *Applied Soft Computing, 112*, Article 107814. https://doi.org/10.1016/j.asoc.2021.107814

Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). https://doi.org/10.1109/ICNN.1995.488968.

Euchi, J., & Sadok, A. (2021). Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication, 44*, Article 101236. https://doi.org/10.1016/j.phycom.2020.101236

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., & Rich, R. (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management (JIEM), 9*(2), 374–388. https://doi.org/10.3926/jiem.1929

Frachtenberg, E. (2019). Practical drone delivery. *Computer, 52*(12), 53–57. https://doi.org/10.1109/MC.2019.2942290

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92.

Gu, R., Poon, M., Luo, Z., Liu, Y., & Liu, Z. (2022). A hierarchical solution evaluation method and a hybrid algorithm for the vehicle routing problem with drones and multiple visits. *Transportation Research Part C: Emerging Technologies, 141*, Article 103733. https://doi.org/10.1016/j.trc.2022.103733

Gunay-Sezer, N. S., Cakmak, E., & Bulkan, S. (2023). A hybrid metaheuristic solution method to traveling salesman problem with drone. *Systems, 11*(5), 259. https://doi.org/10.3390/systems11050259

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies, 86*, 597–621. https://doi.org/10.1016/j.trc.2017.11.015

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *Journal of Heuristics, 26*(2), 219–247. https://doi.org/10.1007/s10732-019-09431-y

Ham, A. M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies, 91*, 1–14. https://doi.org/10.1016/j.trc.2018.03.025

Hamid, M., Nasiri, M. M., & Rabbani, M. (2023). A mixed closed-open multi-depot routing and scheduling problem for homemade meal delivery incorporating drone and crowd-sourced fleet: A self-adaptive hyper-heuristic approach. *Engineering Applications of Artificial Intelligence, 120*, Article 105876. https://doi.org/10.1016/j.engappai.2023.105876

Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 87*, Article 103249. https://doi.org/10.1016/j.engappai.2019.103249

Jeong, H. Y., Song, B. D., & Lee, S. (2019). Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. *International Journal of Production Economics, 214*, 220–233. https://doi.org/10.1016/j.ijpe.2019.01.010

Joshi, S. K., & Bansal, J. C. (2020). Parameter tuning for meta-heuristics. *Knowledge-Based Systems, 189*, Article 105094. https://doi.org/10.1016/j.knosys.2019.105094

Kahraman, H. T., Aras, S., & Gedikli, E. (2020). Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms. *Knowledge-Based Systems, 190*, Article 105169. https://doi.org/10.1016/j.knosys.2019.105169

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing, 8*(1), 687–697. https://doi.org/10.1016/j.asoc.2007.05.007

Karami, H., Anaraki, M. V., Farzin, S., & Mirjalili, S. (2021). Flow direction algorithm (FDA): A novel optimization approach for solving optimization problems. *Computers & Industrial Engineering, 156*, Article 107224. https://doi.org/10.1016/j.cie.2021.107224

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering, 129*, 14–30. https://doi.org/10.1016/j.cie.2019.01.020

Khodadadi, N., Snasel, V., & Mirjalili, S. (2022). Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints. *IEEE Access, 10*, 16188–16208. https://doi.org/10.1109/ACCESS.2022.3146374

Kloster, K., Moeini, M., Vigo, D., & Wendt, O. (2023). The multiple traveling salesman problem in presence of drone-and robot-supported packet stations. *European Journal of Operational Research, 305*(2), 630–643. https://doi.org/10.1016/j.ejor.2022.06.004

Kumar, A., Wu, G., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., & Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation, 56*, Article 100693. https://doi.org/10.1016/j.swevo.2020.100693

Kuo, R. J., Lu, S. H., Lai, P. Y., & Mara, S. T. W. (2022). Vehicle routing problem with drones considering time windows. *Expert Systems with Applications, 191*, Article 116264. https://doi.org/10.1016/j.eswa.2021.116264

Lei, D., Cui, Z., & Li, M. (2022). A dynamical artificial bee colony for vehicle routing problem with drones. *Engineering Applications of Artificial Intelligence, 107*, Article 104510. https://doi.org/10.1016/j.engappai.2021.104510

Li, J., Deng, G., Luo, C., Lin, Q., Yan, Q., & Ming, Z. (2016). A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems. *IEEE Transactions on Vehicular Technology, 65*(12), 9585–9596. https://doi.org/10.1109/TVT.2016.2623666

Liang, Y. J., & Luo, Z. X. (2022). A survey of truck-drone routing problem: Literature review and research prospects. *Journal of the Operations Research Society of China, 10*(2), 343–377. https://doi.org/10.1007/s40305-021-00383-4

Liang, J., Suganthan, P. N., Qu, B. Y., Gong, D. W., & Yue, C. T. (2019). Problem definitions and evaluation criteria for the CEC 2020 special session on multimodal multiobjective optimization, 201912. *Zhengzhou University*. https://doi.org/10.13140/RG.2.2.31746.02247

Liu, Z., Li, X., & Khojandi, A. (2022). The flying sidekick traveling salesman problem with stochastic travel time: A reinforcement learning approach. *Transportation Research Part E: Logistics and Transportation Review, 164*, Article 102816. https://doi.org/10.1016/j.tre.2022.102816

Lu, K. D., Wu, Z. G., & Huang, T. (2022). Differential evolution-based three stage dynamic cyber-attack of cyber-physical power systems. *IEEE/ASME Transactions on Mechatronics, 28*(2), 1137–1148. https://doi.org/10.1109/TMECH.2022.3214314

Lu, K. D., & Wu, Z. G. (2022). Genetic algorithm-based cumulative sum method for jamming attack detection of cyber-physical power systems. *IEEE Transactions on Instrumentation and Measurement, 71*, 1–10. https://doi.org/10.1109/TIM.2022.3186360

Luo, Z., Poon, M., Zhang, Z., Liu, Z., & Lim, A. (2021). The multi-visit traveling salesman problem with multi-drones. *Transportation Research Part C: Emerging Technologies, 128*, Article 103172. https://doi.org/10.1016/j.trc.2021.103172

Luo, Z., Gu, R., Poon, M., Liu, Z., & Lim, A. (2022). A last-mile drone-assisted one-to-one pickup and delivery problem with multi-visit drone trips. *Computers & Operations Research, 148*, Article 106015. https://doi.org/10.1016/j.cor.2022.106015

Masmoudi, M. A., Mancini, S., Baldacci, R., & Kuo, Y. H. (2022). Vehicle routing problems with drones equipped with multi-package payload compartments. *Transportation Research Part E: Logistics and Transportation Review, 164*, Article 102757. https://doi.org/10.1016/j.tre.2022.102757

Meng, S., Guo, X., Li, D., & Liu, G. (2023). The multi-visit drone routing problem for pickup and delivery services. *Transportation Research Part E: Logistics and Transportation Review, 169*, Article 102990. https://doi.org/10.1016/j.tre.2022.102990

MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems, 213*, Article 106711. https://doi.org/10.1016/j.knosys.2020.106711

Moeini, M., & Salewski, H. (2019). A genetic algorithm for solving the truck-drone-ATV routing problem. In *World congress on global optimization* (pp. 1023–1032). Cham: Springer. https://doi.org/10.1007/978-3-030-21803-4_101.

Mohamed, A. W., & Mohamed, A. K. (2019). Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *International Journal of Machine Learning and Cybernetics, 10*, 253–277. https://doi.org/10.1007/s13042-017-0711-7

Mohammadi-Balani, A., Nayeri, M. D., Azar, A., & Taghizadeh-Yazdi, M. (2021). Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Computers & Industrial Engineering, 152*, Article 107050. https://doi.org/10.1016/j.cie.2020.107050

Moshref-Javadi, M., & Winkenbach, M. (2021). Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications, 177*, Article 114854. https://doi.org/10.1016/j.eswa.2021.114854

Murray, C. C., & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies, 110*, 368–398. https://doi.org/10.1016/j.trc.2019.11.003

Nguyen, M. A., Sano, K., & Tran, V. T. (2020). A Monte Carlo Tree search for traveling salesman problem with drone. *Asian Transport Studies, 6*, Article 100028. https://doi.org/10.1016/j.eastsj.2020.100028

Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks, 72*(4), 411–458. https://doi.org/10.1002/net.21818

Pehlivanoglu, Y. V., & Pehlivanoglu, P. (2021). An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. *Applied Soft Computing, 112*, Article 107796. https://doi.org/10.1016/j.asoc.2021.107796

Peng, K., Liu, W., Sun, Q., Ma, X., Hu, M., Wang, D., & Liu, J. (2018). Wide-area vehicle-drone cooperative sensing: Opportunities and approaches. *IEEE Access, 7*, 1818–1828. https://doi.org/10.1109/ACCESS.2018.2886172

Peng, K., Du, J., Lu, F., Sun, Q., Dong, Y., Zhou, P., & Hu, M. (2019). A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. *IEEE Access, 7*, 49191–49200. https://doi.org/10.1109/ACCESS.2019.2910134

Pina-Pardo, J. C., Silva, D. F., & Smith, A. E. (2021). The traveling salesman problem with release dates and drone resupply. *Computers & Operations Research, 129*, Article 105170. https://doi.org/10.1016/j.cor.2020.105170

Poikonen, S., Golden, B., & Wasil, E. A. (2019). A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing, 31*(2), 335–346. https://doi.org/10.1287/ijoc.2018.0826

Rajmohan, S., & Ramasubramanian, N. (2023). Improved Symbiotic organisms search for path planning of unmanned combat aerial vehicles. *Journal of Ambient Intelligence and Humanized Computing, 1–23*. https://doi.org/10.1007/s12652-023-04540-w

Rich, R. (2020). Inverting the truck-drone network problem to find best case configuration. *Advances in Operations Research, 2020*. https://doi.org/10.1155/2020/4053983

Roberti, R., & Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone. *Transportation Science, 55*(2), 315–335. https://doi.org/10.1287/trsc.2020.1017

Rojas Viloria, D., Solano-Charris, E. L., Muñoz-Villamizar, A., & Montoya-Torres, J. R. (2021). Unmanned aerial vehicles/drones in vehicle routing problems: A literature review. *International Transactions in Operational Research, 28*(4), 1626–1657. https://doi.org/10.1111/itor.12783

Sadiq, A. S., Dehkordi, A. A., Mirjalili, S., & Pham, Q. V. (2022). Nonlinear marine predator algorithm: A cost-effective optimizer for fair power allocation in NOMA-VLC-B5G networks. *Expert Systems with Applications, 203*, Article 117395. https://doi.org/10.1016/j.eswa.2022.117395

Sajid, M., Mittal, H., Pare, S., & Prasad, M. (). Routing and scheduling optimization for UAV assisted delivery system: A hybrid approach. *Applied Soft Computing, 126*, Article 109225. https://doi.org/10.1016/j.asoc.2022.109225

Salama, M., & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies, 114*, 620–642. https://doi.org/10.1016/j.trc.2020.01.019

Salawudeen, A. T., Mu'azu, M. B., Yusuf, A., & Adedokun, A. E. (2021). A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowledge-Based Systems, 232*, Article 107486. https://doi.org/10.1016/j.knosys.2021.107486

Schermer, D., Moeini, M., & Wendt, O. (2020). A branch-and-cut approach and alternative formulations for the traveling salesman problem with drone. *Networks, 76*(2), 164–186. https://doi.org/10.1002/net.21958

Shi, Y., Lin, Y., Li, B., & Li, R. Y. M. (2022). A bi-objective optimization model for the medical supplies' simultaneous pickup and delivery with drones. *Computers & Industrial Engineering*. , Article 108389. https://doi.org/10.1016/j.cie.2022.108389

Su, H., Zhao, D., Heidari, A. A., Liu, L., Zhang, X., Mafarja, M., & Chen, H. (2023). RIME: A physics-based optimization. *Neurocomputing, 532*, 183–214. https://doi.org/10.1016/j.neucom.2023.02.010

Tamke, F., & Buscher, U. (2023). The vehicle routing problem with drones and drone speed selection. *Computers & Operations Research, 152*, Article 106112. https://doi.org/10.1016/j.cor.2022.106112

Tanabe, R., & Fukunaga, A. S. (2014, July). Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE Congress on Evolutionary*

Computation (CEC) (Pp. 1658-1665). IEEE. https://doi.org/10.1109/CEC.2014.6900380.

Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors, 22*(3), 855. https://doi.org/10.3390/s22030855

Ulmer, M. W., & Thomas, B. W. (2018). Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks, 72*(4), 475–505. https://doi.org/10.1002/net.21855

Wang, Z., & Sheu, J. B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological, 122*, 350–364. https://doi.org/10.1016/j.trb.2019.03.005

Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 114*, Article 105082. https://doi.org/10.1016/j.engappai.2022.105082

Wen, X., & Wu, G. (2022). Heterogeneous multi-drone routing problem for parcel delivery. *Transportation Research Part C: Emerging Technologies, 141*, Article 103763. https://doi.org/10.1016/j.trc.2022.103763

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution* (Pp. 196-202). New York, NY: Springer, New York. https://doi.org/10.1007/978-1-4612-4380-9_16.

Xu, Y., Xue, X., Sun, Z., Gu, W., Cui, L., Jin, Y., & Lan, Y. (2022). Joint path planning and scheduling for vehicle-assisted multiple Unmanned Aerial Systems plant protection operation. *Computers and Electronics in Agriculture, 200*, Article 107221. https://doi.org/10.1016/j.compag.2022.107221

Yurek, E. E., & Ozmutlu, H. C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies, 91*, 249–262. https://doi.org/10.1016/j.trc.2018.04.009

Zang, X., Jiang, L., Liang, C., Dong, J., Lu, W., & Mladenovic, N. (2022). Optimization approaches for the urban delivery problem with trucks and drones. *Swarm and Evolutionary Computation, 75*, Article 101147. https://doi.org/10.1016/j.swevo.2022.101147

Zhang, J., & Li, Y. (2023). Collaborative vehicle-drone distribution network optimization for perishable products in the epidemic situation. *Computers & Operations Research, 149*, Article 106039. https://doi.org/10.1016/j.cor.2022.106039

Zhang, K., He, F., Zhang, Z., Lin, X., & Li, M. (2020). Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transportation Research Part C: Emerging Technologies, 121*, Article 102861. https://doi.org/10.1016/j.trc.2020.102861

Zhao, S., Zhang, T., Ma, S., & Chen, M. (2022). Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence, 114*, Article 105075. https://doi.org/10.1016/j.engappai.2022.105075