Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones

Yun-qi Han¹, Jun-qing Li¹,², Zhengmin Liu³, Chuang Liu² and Jie Tian¹

Abstract
In some special rescue scenarios, the needed goods should be transported by drones because of the landform. Therefore, in this study, we investigate a multi-objective vehicle routing problem with time window and drone transportation constraints. The vehicles are used to transport the goods and drones to customer locations, while the drones are used to transport goods vertically and timely to the customer. Three types of objectives are considered simultaneously, including minimization of the total energy consumption of the trucks, total energy consumption of the drones, and the total number of trucks. An improved artificial bee colony algorithm is designed to solve the problem. In the proposed algorithm, each solution is represented by a two-dimensional vector, and the initialization method based on the Push-Forward Insertion Heuristic is embedded. To enhance the exploitation abilities, an improved employed heuristic is developed to perform detailed local search. Meanwhile, a novel scout bee strategy is presented to improve the global search abilities of the proposed algorithm. Several instances extended from the Solomon instances are used to test the performance of the proposed improved artificial bee colony algorithm. Experimental comparisons with the other efficient algorithms in the literature verify the competitive performance of the proposed algorithm.

Keywords
Vehicle routing problem, artificial bee colony, time window, drone transportation, multi-objective

Introduction
The utilization of smart manufacturing and intelligent devices has greatly promoted the transforming of traditional industries.¹ Thus far, many emerging computer technologies²,³ and sophisticated equipment such as the robotics⁴ and drones⁵ have extremely improved the production efficiency and reduced the cost of the enterprise. Recently, numerous excellent enterprises have invested various drones to deliver their commodities⁶,⁷ while many researchers have also concentrated on the vehicle routing problem with drones (VRPD),⁸-¹² which can be observed as an extended version of the vehicle routing problem (VRP).¹³ Many exhaustive overviews of the VRP could be found in the literature,¹⁴-¹⁶ where a number of customers need to be visited successively by a fleet of vehicles. In VRPD, the customers should be served by both vehicles and drones.⁸ The applications of drones in the VRP systems have attracted many unexpected benefits.¹⁰ Wang et al. implemented a set of thoroughly mathematical experiments to testify that combining the drones with trucks or vehicles can

¹ School of Information and Engineering, Shandong Normal University, Jinan, China
² School of Computer Science, Liaocheng University, Liaocheng, China
³ School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, China

Corresponding author:
Jun-qing Li, School of Information and Engineering, Shandong Normal University, Jinan 250014, Shandong, China.
Email: lijunqing@lcu-cs.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
In the last decades, many algorithms have been applied to the vehicle routing problem with time window (VRPTW), such as the genetic algorithm (GA), tabu search (TS) algorithm, variable neighborhood search (VNS) algorithm, gradient evolution algorithm, the particle swarm optimization, ant colony optimization, the adaptive large neighborhood search heuristic, and the simulated annealing algorithm. However, for some special requirements, the drones can complete transportation tasks. Chang and Lee established a model for one truck equipped a set of drones. Wang et al. studied a group of trucks with several drones. Murray and Chu designed a dispatching manner in which one drone cooperates with one truck to serve a fleet of customers. Furthermore, another valuable research direction involving the VRP is the vehicle routing problem with time window and drones (VRPTWD). Guerriero et al. proposed a VRPTWD model considering the soft time window and customer satisfactions. Phan and Suzuki studied a multi-objective optimization problem of VRPTWD with the constraints of simultaneous receiving and dispatching. In a nutshell, the VRP with drones should be taken as a competitive optimization method compared with other multi-objective optimization algorithms.

### Problem statement and MO-VRPTW-D model

#### Problem description

The proposed MO-VRPTW-D can be generalized from three aspects: customers, trucks and drones, objectives and constraints.

For the customers, the complex transportation scenario, such as the disaster relief and rescue or the transport of delicate goods, should be considered. Besides, each customer has multiple pairs of productions or demands. Except from the special of height and demand, the customers also have the synchronized visits and time windows constraints.

For the trucks and drones, each truck equips a drone and the multiple pairs of productions are delivered to the horizontal location of a customer by the truck and raised to the customer by the drone. Moreover, each truck has two types of capacities due to the multiple pairs of productions for customers and the capacity of each truck is different. The truck that has a greater capacity will consume more fuels and generate more energy consumption than the smaller one. Furthermore, each drone possesses different flight speed and different energy consumption, that is, the faster drone will create greater energy consumption.

Additionally, the objectives of MO-VRPTW-D are to minimize the total energy consumptions of trucks, the entire energy consumptions of drones, and the maximal number of the employed trucks. And the total energy consumptions are respected to the energy consumption index and the travel distances for each truck or drone. For the constraints, first, each truck has the limit of maximum capacity and maximal travel distance, and the maximum number of the trucks is fixed; secondly, each customer has a time windows and the penalty will emerge when the arrival time of a truck at a customer is overstepped the time window. Besides, all the truck must depart and return to the depot and each customer must be visited only once by exactly one truck.

#### MO-VRPTW-D model

The MO-VRPTW-D is modeled as a directed complete graph $G = (N, A)$: $N$ is the vertex set including the customers set $C = \{1, 2, ..., n\}$ and the depot node 0; here 0 and $0_c$ are respecting the start and end depot, respectively, and we set the $0 = 0_c = 0$, for clear expression; while $A$ is the arc set comprising the set of horizontal distances between the vehicles and drones.
customers $A_h = \{(i, j)|i, j \in C, i \neq j\}$ and a set of perpendicular distances $A_p = \{(i_b, i_t)|i \in C\}$; here $i_b$ and $i_t$ are symbolizing the bottom and top location of the perpendicular arc for customer $i$, respectively.

$V = \{1, 2, ..., v\}$ is the trucks set and $D = \{d_1, d_2, ..., d_v\}$ is a set of drones, where $d_i$ respects the drone of the truck $k$. Each truck $k$ has two types of capacities $b_{mk}$ and $b_{sk}$ while the former is the capacity of the main materials and the latter is the supplementations. Each customer $i \in C$ also has two kinds of demands $d_{mi}$ and $d_{si}$ such as one tent must be equipped four or more tent frames when building the temporary accommodations for the victims of a natural calamity. Notice that the amount of all these demands carried by truck $k$ must be less than or equal to the corresponding maximal capacities of truck $k$. There is a maximal travel distance $r_k$ of each truck $k$ which starts at depot $0$, and ends at depot $0$ after serving some customers.

Moreover, each customer must be visited only once by exactly one truck and each customer $i$ has a time window $[e_i, l_i]$. $[e_0, l_0]$ is the time window of depot, $e_0$ is equal to zero while $l_0$ is equal to $r_k$. The wait time $w_i$ will be produced if the arrival time $z_i$ of truck $k$ at customer $i$ is early than $e_i$ and when $z_i$ is later than $l_i$, the penalty will be attached to this route.

Specially, there is an energy consumption coefficient $w_{tk}$ for each truck $k$ while each drone $d_k$ has an energy consumption coefficient $w_{dk}$ and a flight speed $y_{dk}$. In this article, each customer $i$ has a vertical trip distance $t_{yi}$ and the serving duration $s_i$ of customer $i$ can calculate by dividing the double trip distance $2t_{yi}$ of the $y_{dk}$. Here $y_{dk}$ is the speed of drone equipped at truck $k$ which visiting the customer $i$. Besides, $\alpha$, $\beta$, and $\gamma$ are the weight coefficients of the total energy consumption of the trucks, total energy consumption of the drones, and the total number of trucks, respectively.

Finally, all of the travel distance $t_{ij}$ or $t_{ji}$ is numerically equal to the travel or trip time in the MO-VRPTW-D model. The mathematical formulations are described below.

Decision variables

\[ x_{ijk} = \begin{cases} 
1, & \text{if truck } k \text{ travel from vertex } i \text{ to vertex } j \\
0, & \text{otherwise}
\end{cases} \]

\[ y_{ijk} = \begin{cases} 
1, & \text{if vertex } i \text{ is served by vehicle } k \\
0, & \text{otherwise}
\end{cases} \]

Objective

\[
\min \alpha \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{v} x_{ijk} w_{tk} t_{ij} + \beta \sum_{i=0}^{n} \sum_{k=1}^{v} x_{ijk} 2t_{yi} \frac{w_{dk}}{y_{dk}} + \gamma \sum_{k=1}^{v} x_{ijk}, i \neq j, i, j \in N, k \in V
\]

Constraints

\[
\sum_{i=1}^{n} d_{mi} y_{ijk} \leq b_{mk}, \forall k \in V
\]

\[
\sum_{i=1}^{n} d_{si} y_{ijk} \leq b_{sk}, \forall k \in V
\]

\[
\sum_{k=1}^{v} y_{ijk} = 1, \forall i \in C
\]

\[
\sum_{i=0}^{n} x_{ijk} = y_{ijk}, \forall k \in V, \forall j \in C
\]

\[
\sum_{j=0}^{n} x_{ijk} = y_{ijk}, \forall k \in V, \forall i \in C
\]

\[
\sum_{i=1}^{n} x_{ijk} - \sum_{i=1}^{n} x_{ijk} = 0, \forall k \in V
\]

\[
s_i = 2t_{yi} \frac{1}{y_{dk}}, \forall i \in C, \forall k \in V
\]

\[
\sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{v} x_{ijk}(t_{ij} + s_i + w_i) \leq r_k, \forall k \in V
\]

\[
w_0 = s_0 = 0
\]

\[
\sum_{k=1}^{v} \sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk}(z_i + w_i + s_i + t_{ij}) = z_j, \forall j \in C
\]

\[
e_i \leq z_i + w_i \leq l_i, \forall i \in C
\]

\[
w_i = \max\{0, e_i - z_i\}, \forall i \in C
\]

\[
x_{ijk} \in \{0, 1\}, \forall i, j \in C, \forall k \in V
\]

\[
y_{ijk} \in \{0, 1\}, \forall i \in C, \forall k \in V
\]

\[
z_i \geq 0, \forall i \in C
\]

The objective (1) aims to minimize the weighted sum of the total energy consumption of the trucks, total energy consumption of the drones, and the total number of trucks. Constraints (2) and (3) ensure the two kinds of truck capacities are not exceeded; constraints (4) to (6) guarantee that each customer is visited by only one truck and served only once; constraint (7) safeguards that each truck starts and ends at the depot; constraint (8) sets the serving duration; constraint (9) guarantees the maximal trip distance is not exceeded; constraint (10) indicates the waiting and starting time of depot; constraint (11) gives the connection between the arrival time of a vertex and the departure time of its predecessor; constraints (12) and (13) illuminate that the arrival time should be within the time window and the wait time.
Example of MO-VRPTW-D

Figure 1 displays an example of the proposed problem, which includes seven customers, two trucks and drones, which is conducive to comprehending the dispatching process. Figure 1(a) is the whole delivery routes for all customers, while Figure 1(b) is the situation of dispatching goods to the customer 3. For convenience, we only discuss the delivery process of customer 3 and other conditions are as follows: (1) the route {0, 1, 2, 3, 4, 0} and {0, 5, 6, 7, 0} are executed by the truck 1 and truck 2, respectively; (2) drone 1 is equipped on the truck 1 and the speed of drone 1 is 0.5. It can be acquired from Figure 1 that, when the truck 1 arrival at customer 3, the drone 1 starts to dispatch goods. According to the height of customer 3 is 5 and the dispatching route is a double journey, thus, the total distance of drone 1 is 10. Moreover, on the basis of the speed of drone 1 is 0.5, thus, the trip time of the drone is 10 divided by 0.5 is 20. Furthermore, the serving duration of customer 3 is 20. And Figure 2 explains the more detailed distribution scheme, where each route is built by a truck with a drone and each customer has two types of demands. The bottom of Figure 2 shows a part of the route, in which the main materials and the supplantations of customer 2 are 1 and 3; of customer 4 are 1 and 4, respectively.

Proposed algorithm

The framework of the proposed IABC algorithm is shown in Algorithm 1.

Encoding and decoding methods

For the problem encoding, this study states each solution as a 2-D array where each array representing the sequence of a set of vertexes, and Figure 3 gives an example of the encoding. In Figure 3, {0, 1, 2, 3, 4, 0} represent a visiting sequence of a truck, where this truck starts at depot then visits customer 1, 2, and so on until it returns to the depot. And {0, 5, 6, 7, 0} is symbolizing another route. Notice that each customer in the constructed route must be selected under the constraints of the MO-VRPTW-D model.

Repair strategy

For each infeasible solution, Algorithm 2 gives the proposed repair strategy and the time complexity is $O(n^2m)$.

Initialization strategy

Solomon proposed the Push-Forward Insertion Heuristic (PFIH) for the VRPTW in 1987 and the general process of PFIH is as follow: first, select one of the unscheduled customers who must satisfy all the constraints of VRPTW model if insert it into the current route; then, calculate all the costs of inserting it into each location of the current route; thirdly, insert this customer into current route at the
position of minimum costs; fourthly, repeat the above three steps until all customers are arranged. In this study, we also embed the PFIH method in the proposed algorithm.

Enhanced employed bee strategy

Algorithm 3 gives an enhanced employed bee strategy of this study. In the traditional employed bee strategy, we select some customers randomly of certain route, then rearrange these customers to other routes. When some customers cannot be inserting into the current route, we could deploy a new truck if there was an empty one. This method is easy to accomplish but hard to receive a better solution. Thus, we propose a novel and effective strategy to obtain better routing schemes, where we expect rearranging a fleet of relevant customers with similar location. And the time complexity of our strategy is $O(n^2m)$.

Scout bee strategy

When the IABC algorithm comes to the scout phase, we expect to make large changes to a solution in order to enhance the fitness and reduce the energy consumptions of these routes. Thus, the scout bee strategy described

Algorithm 1. The framework of the IABC.

Input: the datas of customers, trucks and drones
Output: the best solution $X^*$
1 generate $P_s$ initial solutions $X$ by the strategy in sub-section 3.4
2 employed bee phase
   for each solution $x$ do
3 generate a new solution $x^*$ by the strategy in sub-section 3.5 for solution $x$
4 if $x^*$ is better than $x$
5 replace $x$ with $x^*$
6 update the best solution that has been found so far to $x^*$
7 update the local best to $x$
8 end
9 end
10 onlooker bee phase
   for each solution $x$ do
11 randomly select a solution $x_i$
12 implement the Employed bee phase on the best solution between $x$ and $x_i$
13 update the best solution $X^*$
14 end
15 scout bee phase
   implement the Scout bee strategy for the best solution $X^*$
16 update the iterative time $T$ of the IABC algorithm
17 update iteration times without improvement $I_s$ for each solution $x$
18 if $I_s$ exceeds the maximum no-improvement time $I_{n}$
19 implement the Scout bee strategy for the solution $x'$
20 if $T$ not exceeds the $T_{max}$
21 go to step 2

Figure 3. Example of the problem encoding.
Algorithm 2. Repair strategy.

| Input: an infeasible solution |
|-------------------------------|
| Output: an feasible solution |
| for each customer $i$ on each route of an infeasible solution $X$ do |
| if $i$ violates its time window |
| keep $i$ in the set $S$ and delete it from its route |
| end |
| for each customer $j$ in the set $S$ do |
| try to insert $j$ to each route of the current $X$ according the PFIH method in sub-section 3.4 |
| if customer $j$ still cannot be arranged and there has at least one available empty truck |
| put the customer $j$ into the truck $k$ |
| end |
| else |
| reject this solution $X$ |
| end |
| end |

Algorithm 3. Enhanced employed bee strategy.

| Input: an solution |
|-------------------|
| Output: an improved solution |
| set the number $n$ of selected customers set $D$ is 0, and put all the customers into set $D_n$ |
| While $n < \text{Psize}/10$ |
| random select a customer $r$ of $D_n$ and sent $r$ to the set $D$ |
| delete the customers in set $D$ from set $D_n$ |
| random select a customer $i$ of set $D$ |
| for each customer $j$ in $D_n$ do |
| calculate all the distance arc $a(i, j)$ |
| end |
| sort the set $D_n$ from smallest to biggest accord all the arc $a(i, j)$ |
| generate a number $d$ from $[1, \text{Psize}/10]$ |
| store the top $d$ customers of $D_n$ into $D$ and delete these customers from $D_n$ |
| update the customers number $n$ of set $D$ |
| end |
| delete set $D$ from current solution |
| for each customer $l$ of set $D$ do |
| insert $l$ to current solution according to the PFIH and delete $l$ from set $D$ |
| end |
| if the set $D$ is not empty |
| dispatch an available truck and reinsert these unscheduled customers |
| end |
Algorithm 4. Scout bee strategy.

**Input:** the best solution $X^*$  
**Output:** an improved solution $x^*$

1. random select $P_{size}/10$ customers from the best solution $X^*$  
2. put the $P_{size}/10$ customers into the set $D_{sc}$ and delete them from the best solution $X^*$

for each customer $i$ of $D_{sc}$ do

4. insert $i$ into the current best solution $X^*$ by the PFIH strategy

end and clean the set $D_{sc}$

6. random select a truck $k$ from the current best solution $X^*$

7. store the customers of truck $k$ in the set $D_{sc}$ and delete these customers from truck $k$

8. transform the truck $k$ of $X^*$ into another available empty truck

9. insert the customers of $D$ into current $X^*$ by the PFIH strategy

10. update the best solution $X^*$

end

---

Table 1. The levels of the two parameters.

| Parameter | Level 1 | Level 2 | Level 3 | Level 4 |
|-----------|---------|---------|---------|---------|
| $P_s$     | 50      | 100     | 150     | 200     |
| $L_n$     | 5       | 10      | 15      | 20      |

Figure 4. The result of the DOE experiments for $P_s$ and $L_n$. DOE: design of experiments.

Table 2. The comparison results of IABC-NE and IABC.

| Instance | Deviation |
|----------|-----------|
| Inst1    | 0.00      | 3.65     |
| Inst2    | 6.25      | 0.00     |
| Inst3    | 0.00      | 4.57     |
| Inst4    | 0.00      | 2.75     |
| Inst5    | 1.23      | 1.95     |
| Inst6    | 2.44      | 0.00     |
| Inst7    | 0.00      | 0.00     |
| Inst8    | 8.12      | 3.32     |
| Inst9    | 1.00      | 1.46     |
| Inst10   | 7.31      | 0.00     |
| Inst11   | 0.88      | 0.00     |
| Inst12   | 11.35     | 0.00     |
| Inst13   | 0.00      | 1.46     |
| Inst14   | 0.00      | 2.34     |
| Inst15   | 5.08      | 0.00     |
| Inst16   | 8.51      | 0.00     |
| Inst17   | 3.35      | 0.00     |
| Inst18   | 2.14      | 0.00     |
| Inst19   | 1.05      | 0.00     |
| Inst20   | 5.45      | 0.00     |
| Inst21   | 0.00      | 0.92     |
| Inst22   | 3.45      | 0.00     |
| Inst23   | 6.36      | 0.00     |
| Inst24   | 0.00      | 1.28     |
| Inst25   | 1.52      | 0.00     |
| Inst26   | 7.48      | 0.00     |
| Inst27   | 0.00      | 0.15     |
| Inst28   | 0.18      | 0.00     |
| Inst29   | 7.08      | 0.00     |
| Inst30   | 0.00      | 8.35     |

(continued)
Algorithm 4 is called the truck replacement strategy and the time complexity of the scout bee phase is $O(n^2m)$.

### Experimental comparisons

In this study, in order to testify the validity of IABC, three relevant and effective algorithms, that is, GA, TS, and

| Instance | Deviation |
|----------|-----------|
| Inst32   | 1.04      |
| Inst33   | 3.48      |
| Inst34   | 2.55      |
| Inst35   | 1.33      |
| Inst36   | 1.14      |
| Inst37   | 0.00      |
| Inst38   | 1.21      |
| Inst39   | 0.00      |
| Inst40   | 0.44      |
| Inst41   | 0.00      |
| Inst42   | 0.00      |
| Inst43   | 3.29      |
| Inst44   | 0.00      |
| Inst45   | 0.00      |
| Inst46   | 1.99      |
| Inst47   | 4.06      |
| Inst48   | 3.18      |
| Inst49   | 0.00      |
| Inst50   | 2.20      |
| Inst51   | 5.25      |
| Inst52   | 1.36      |
| Inst53   | 2.52      |
| Inst54   | 2.31      |
| Mean     | 2.30      |

IABC: improved artificial bee colony.

IABC-NE: improved artificial bee colony with a random employed bee strategy without the enhanced employed bee strategy.

| Instance | Deviation |
|----------|-----------|
| Inst1    | 0.29      |
| Inst2    | 10.22     |
| Inst3    | 0.53      |
| Inst4    | 0.00      |
| Inst5    | 2.20      |
| Inst6    | 0.25      |
| Inst7    | 0.00      |
| Inst8    | 2.16      |
| Inst9    | 1.54      |
| Inst10   | 0.00      |
| Inst11   | 0.00      |
| Inst12   | 6.98      |
| Inst13   | 17.64     |
| Inst14   | 0.00      |
| Inst15   | 0.00      |
| Inst16   | 0.00      |
| Inst17   | 14.80     |
| Inst18   | 0.68      |
| Inst19   | 0.56      |
| Inst20   | 0.10      |
| Inst21   | 5.46      |
| Inst22   | 6.08      |
| Inst23   | 3.87      |
| Inst24   | 0.00      |
| Inst25   | 0.06      |
| Inst26   | 1.21      |
| Inst27   | 2.84      |
| Inst28   | 0.00      |
| Inst29   | 1.07      |
| Inst30   | 0.68      |
| Inst31   | 0.00      |
| Inst32   | 1.69      |
| Inst33   | 4.78      |
| Inst34   | 0.00      |
| Inst35   | 0.90      |
| Inst36   | 0.00      |
| Inst37   | 0.00      |
| Inst38   | 3.46      |
| Inst39   | 0.00      |
| Inst40   | 5.51      |
| Inst41   | 0.00      |
| Inst42   | 1.78      |
| Inst43   | 3.16      |
| Inst44   | 2.95      |
| Inst45   | 2.64      |
| Inst46   | 1.59      |
| Inst47   | 2.33      |
| Inst48   | 3.04      |
| Inst49   | 0.77      |
| Inst50   | 2.17      |
| Inst51   | 0.00      |
| Inst52   | 0.00      |
| Inst53   | 6.72      |
| Inst54   | 9.65      |
| Inst55   | 0.00      |
| Mean     | 2.41      |

### Table 2. The results of the IABC-NS and IABC.

| Instance | Deviation |
|----------|-----------|
| Inst32   | 1.04      |
| Inst33   | 3.48      |
| Inst34   | 2.55      |
| Inst35   | 1.33      |
| Inst36   | 1.14      |
| Inst37   | 0.00      |
| Inst38   | 1.21      |
| Inst39   | 0.00      |
| Inst40   | 0.44      |
| Inst41   | 0.00      |
| Inst42   | 0.00      |
| Inst43   | 3.29      |
| Inst44   | 0.00      |
| Inst45   | 0.00      |
| Inst46   | 1.99      |
| Inst47   | 4.06      |
| Inst48   | 3.18      |
| Inst49   | 0.00      |
| Inst50   | 2.20      |
| Inst51   | 5.25      |
| Inst52   | 1.36      |
| Inst53   | 2.52      |
| Inst54   | 2.31      |
| Mean     | 2.30      |

IABC: improved artificial bee colony.

IABC-NE: improved artificial bee colony with a random employed bee strategy without the enhanced employed bee strategy.

Figure 5. The ANOVA comparisons of IABC-NE and IABC. ANOVA: analysis of variance; IABC: improved artificial bee colony. IABC-NE: improved artificial bee colony with a random employed bee strategy without the enhanced employed bee strategy.

Algorithm 4 is called the truck replacement strategy and the time complexity of the scout bee phase is $O(n^2m)$.
VNS\textsuperscript{25} are selected as the comparing algorithm. The “Experimental parameters and instances” subsection describes the parameters and instances setting of the MO-VRPTW-D and other experimental results are expressed in the rest of the fourth section.

**Experimental parameters and instances**

The two most significant parameters of IABC, that is, are the size of population \( P_s \) and the iteration times without improvement \( L_n \). We set four levels for each parameter in Table 1 and perform a mass of adjustable parameter experiments according to these levels. Accord to the result of design of experiments (DOE) in Figure 4, we set \( P_s = 100 \) and \( L_n = 10 \), respectively.

For the instances of this study, we design our instances based on the 55 famous Solomon Benchmark examples of

![Figure 6. The ANOVA of the IABC-NE and IABC. ANOVA: analysis of variance; IABC: improved artificial bee colony. IABC-NE: improved artificial bee colony with a random employed bee strategy without the enhanced employed bee strategy.](image)

**Table 4. The results of comparing with GA, TS, and VNS.**

| Instance | Best | GA | TS | VNS | IABC | GA | TS | VNS |
|----------|------|----|----|-----|------|----|----|-----|
| Inst1    | 1038.78 | 1571.34 | 1306.84 | 1351.27 | 1038.78 | 51.27 | 25.81 | 30.08 |
| Inst2    | 1001.63 | 1275.01 | 1163.90 | 1125.35 | 1001.63 | 27.29 | 16.20 | 12.35 |
| Inst3    | 950.11 | 1134.35 | 950.11 | 980.92 | 960.26 | 12.99 | 10.57 | 15.03 |
| Inst4    | 956.58 | 1217.11 | 981.48 | 1011.20 | 956.58 | 27.24 | 2.60 | 5.71 |
| Inst5    | 1010.41 | 1313.63 | 1108.59 | 1112.75 | 1010.41 | 30.01 | 9.72 | 10.13 |
| Inst6    | 1018.00 | 1443.83 | 1175.09 | 1220.65 | 1018.00 | 41.83 | 15.43 | 19.91 |
| Inst7    | 993.27 | 1246.36 | 1116.44 | 1151.58 | 993.27 | 25.48 | 12.40 | 15.94 |
| Inst8    | 898.22 | 1244.89 | 908.67 | 968.04 | 898.22 | 38.60 | 1.16 | 7.77 |
| Inst9    | 891.91 | 1152.61 | 975.02 | 968.04 | 891.91 | 29.23 | 9.32 | 6.08 |
| Inst10   | 401.30 | 927.18 | 783.94 | 747.19 | 401.30 | 131.05 | 95.35 | 86.19 |
| Inst11   | 519.29 | 802.33 | 733.91 | 801.66 | 519.29 | 54.51 | 41.33 | 54.38 |
| Inst12   | 525.56 | 779.52 | 657.00 | 661.34 | 525.56 | 48.32 | 25.01 | 25.83 |
| Inst13   | 518.87 | 751.77 | 591.87 | 600.71 | 518.87 | 44.89 | 14.07 | 15.77 |
| Inst14   | 503.33 | 932.10 | 811.10 | 791.39 | 503.33 | 85.19 | 61.15 | 57.23 |
| Inst15   | 464.63 | 796.87 | 680.49 | 707.22 | 464.63 | 71.51 | 46.46 | 52.21 |
| Inst16   | 558.53 | 839.41 | 664.32 | 714.66 | 558.53 | 50.29 | 18.94 | 27.95 |
| Inst17   | 499.76 | 680.18 | 499.76 | 513.54 | 509.22 | 36.10 | 2.76 | 1.89 |
| Inst18   | 1578.94 | 1643.01 | 1585.90 | 1578.94 | 1599.14 | 4.06 | 0.44 | 0.00 |
| Inst19   | 1355.03 | 1461.51 | 1355.03 | 1390.12 | 1355.08 | 7.86 | 2.59 | 0.00 |
| Inst20   | 1063.10 | 1151.01 | 1064.14 | 1082.83 | 1063.10 | 8.27 | 0.10 | 1.86 |
| Inst21   | 797.18 | 895.03 | 813.34 | 818.38 | 797.18 | 12.27 | 2.03 | 2.66 |
| Inst22   | 1117.63 | 1185.05 | 1142.73 | 1117.63 | 1135.03 | 6.63 | 2.25 | 1.56 |
| Inst23   | 981.92 | 1062.77 | 981.92 | 1002.99 | 999.78 | 8.23 | 2.15 | 1.82 |
| Inst24   | 832.25 | 887.26 | 870.08 | 892.30 | 832.25 | 6.61 | 4.55 | 7.22 |
| Inst25   | 687.64 | 769.06 | 687.64 | 730.08 | 694.18 | 11.84 | 6.17 | 0.95 |
| Inst26   | 955.97 | 1038.45 | 965.79 | 971.37 | 955.97 | 8.63 | 1.03 | 1.61 |
| Inst27   | 824.02 | 881.33 | 860.44 | 858.06 | 824.02 | 6.93 | 4.42 | 4.13 |
| Inst28   | 807.97 | 909.46 | 844.24 | 830.10 | 807.97 | 12.56 | 4.49 | 2.74 |
| Inst29   | 692.69 | 746.96 | 696.30 | 698.55 | 692.69 | 7.84 | 0.52 | 0.85 |
| Inst30   | 844.25 | 899.89 | 887.98 | 911.36 | 844.25 | 6.59 | 5.18 | 7.95 |
| Inst31   | 746.70 | 780.15 | 753.14 | 746.70 | 748.66 | 4.48 | 0.86 | 0.00 |
| Inst32   | 634.32 | 699.21 | 647.16 | 639.32 | 634.32 | 10.23 | 2.02 | 0.79 |
| Inst33   | 432.21 | 449.42 | 447.22 | 446.69 | 432.21 | 3.98 | 3.47 | 3.35 |
| Inst34   | 561.85 | 575.88 | 561.85 | 567.96 | 565.48 | 2.50 | 0.00 | 1.09 |
| Inst35   | 460.63 | 501.04 | 460.63 | 476.30 | 465.17 | 8.77 | 0.00 | 3.40 |

(continued)
The VRPTW. Each Solomon example has 100 customers and the distributions of customers’ location have three levels: random, cluster, and semi-cluster. Each customer in the Solomon examples has a time window, a demand, a serving duration, and a pair of coordinate points. The designed instances have the same customer numbers, locations, and time windows with the Solomon example and the name of our instances are corresponding to the order of Solomon instances. Due to the two kinds of demands constraints, we randomly set the $d_{mi}$: $d_{si}$ for each customer $i$ from [1:1] to [1:5]. Besides, each customer has an integer height which is generated randomly from 5 to 14. Moreover, the total number of trucks is 25, and each truck has different energy consumption coefficient and two kinds of capacities. And the speed and energy consumption coefficient of drone on each truck is also different.

Effectiveness of the enhanced employed bee strategy

To testify the effectiveness of the proposed employed bee strategy, we perform a detailed experiment. Table 2 illustrates the comparison results of the enhanced employed bee strategy and the canonical employed bee method. In Table 2, the last two columns introduce the deviation of the objective value of each strategy compared with the best value. According to the results in Table 2, the IABC obtains 35 best values while the improved artificial bee colony with a random employed bee strategy without the enhanced employed bee strategy (IABC-NE) only gets 20 best ones, which demonstrate that the proposed employed bee strategy of this study indeed contributes to improving the algorithm performance. Figure 5 exhibits the result of analysis of variance (ANOVA) for the two different methods.

Effectiveness of the scout bee strategy

To investigate the performance of the proposed truck replacement strategy in the scout bee method, we also made a detailed comparison of the scout bee strategy. The results are revealed in Table 3, where IABC-NS represents the IABC without the scout bee strategy. It can be concluded from Table 3 that: (1) IABC obtains 37 best values while

| Instance | Best | Deviation |
|----------|------|-----------|
| Inst36   | 435.31 | 11.46       |
| Inst37   | 351.92 | 1.94       |
| Inst38   | 527.66 | 3.10       |
| Inst39   | 538.09 | 14.41      |
| Inst40   | 428.06 | 11.92      |
| Inst41   | 1324.72 | 5.79       |
| Inst42   | 1087.30 | 13.24      |
| Inst43   | 936.53 | 6.88       |
| Inst44   | 878.78 | 6.23       |
| Inst45   | 1269.41 | 11.69     |
| Inst46   | 987.55 | 5.35       |
| Inst47   | 939.38 | 6.52       |
| Inst48   | 876.64 | 6.95       |
| Inst49   | 855.66 | 5.09       |
| Inst50   | 731.28 | 11.02      |
| Inst51   | 544.70 | 15.19      |
| Inst52   | 479.69 | 6.72       |
| Inst53   | 826.79 | 10.33      |
| Inst54   | 626.18 | 5.11       |
| Inst55   | 565.48 | 25.03      |
| Mean     | 787.92 | 20.61      |

GA: genetic algorithm; TS: tabu search; VNS: variable neighborhood search; IABC: improved artificial bee colony.
Figure 8. The convergence curves of instances. (a) Convergence curve for Inst 1. (b) Convergence curve for Inst 10. (c) Convergence curve for Inst 16. (d) Convergence curve for Inst 25. (e) Convergence curve for Inst 33. (f) Convergence curve for Inst 40. (g) Convergence curve for Inst 48. (h) Convergence curve for Inst 55.
IABC-NS only obtains 18 optimal solutions; (2) the deviation values from the last two columns also show the superiority of the proposed scout bee strategy; and (3) the average performance given in the last line verify the average performance of the proposed method. The ANOVA comparison of the two methods given in Figure 6 also verifies that the proposed scout bee strategy improved the performance significantly.

**Comparison of several efficient algorithms**

To further attest the powerful performance of the IABC, we compare the proposed algorithm with GA, TS, and VNS. These three selected algorithms are efficient for solving the VRPTW. The comparison results are given in Table 4, where the three columns from 3 to 6 correspond to the results obtained by GA, TS, VNS, and IABC, respectively. The best values of these four algorithms are shown in the second column. The deviations between each algorithm and the best values are present in the last four columns. Obviously, IABC gains 38 best values which account for about 70% of the 55 instances. The mean of deviation of IABC is only 0.42 which less than 5% of the second best result. Thus, in terms of the effectiveness and stability of the algorithm, the IABC is superior to the other three algorithms. Figure 7 gives the ANOVA among these four algorithms, which demonstrate that the IABC is significantly better than the three compared algorithms.

**Conclusion**

This article proposes a novel VRPTW for the special customers must be served by trucks and drones. Based on the problem features, the mathematic model of MO-VRPTW-D is designed and solved by the presented IABC Figure 8. However, there still have many challenges between the model and the practical applications. For trucks and drones, we usually deem the motion states of these as a linear process but the speeds are nonlinear in reality. The route also may be impacted by the emergency and weather. As a result, our future work will be devoted to following works: (1) refer to He et al., and extend the model from a static scenario to a dynamic scenario; and (2) considering the fuzzy features of the realistic system and explore the new mode of combining drone with vehicle under uncertain environments.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study is partially supported by National Science Foundation of China under Grants 61773192, 61803192, and 61773246, Shandong Province Higher Educational Science and Technology Program (J17KZ005), special fund plan for local science and technology development lead by central authority, major basic research projects in Shandong (ZR2018ZB0419), and a Grant of Key Laboratory of Intelligent Optimization and Control with Big Data.

**ORCID iD**

Jun-qing Li https://orcid.org/0000-0002-3617-6708

**References**

1. Kusiak A. Smart manufacturing. *Int J Prod Res* 2018; 56(1–2): 508–517.
2. He B, Shao Y, Wang S, et al. Product environmental footprints assessment for product life cycle. *J Clean Prod* 2019; 233(10): 446–460.
3. He B, Liu Y, Zeng L, et al. Product carbon footprint across sustainable supply chain. *J Clean Prod* 2019; 241: 118320.
4. He B, Wang S, and Liu Y. Underactuated robotics: a review. *Int J Adv Robot Syst* 2019; 16(4): 1292981419862164.
5. Floreano D and Wood RJ. Science, technology and the future of small autonomous drones. *Nature* 2015; 521(7553): 460–466.
6. Yurek EE and Ozmutlu HC. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transp Res Part C Emerg Technol* 2018; 91: 249–262.
7. Otto A, Agatz N, Campbell J, et al. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. *Networks* 2018; 72(4): 411–458.
8. Murray CC and Chu AG. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. *Transp Res Part C Emerg Technol* 2015; 54: 86–109.
9. Dorling K., Heinrichs J, Messier GG, et al. Vehicle routing problems for drone delivery. *IEEE Trans Syst Man Cybern Syst* 2016; 47(1): 70–85.
10. Wang X, Poikonen S, and Golden B. The vehicle routing problem with drones: several worst-case results. *Optim Lett* 2017; 11(4): 679–697.
11. Agatz N, Bouman P, and Schmidt M. Optimization approaches for the traveling salesman problem with drone. *Transp Sci* 2018; 52(4): 965–981.
12. Karak A and Abdelghany K. The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transp Res Part C Emerg Technol* 2019; 102: 427–449.
13. Dantzig GB and Ramser JH. The truck dispatching problem. *Manag Sci* 1959; 6(1): 80–91.
14. Eksioglu B, Vural AV, and Reisman A. The vehicle routing problem: a taxonomy review. *Comput Ind Eng* 2009; 57(4): 1472–1483.
15. Pillac V, Gendreau M, Guéret C, et al. A review of dynamic vehicle routing problems. *Eur J Oper Res* 2013; 225(1): 1–11.
16. Braeckers K, Ramaekers K, and Van Nieuwenhuyse I. The vehicle routing problem: state of the art classification and review. *Comput Ind Eng* 2016; 99: 300–313.
17. Wang Z and Sheu JB. Vehicle routing problem with drones. *Transp Res Part B Methodol* 2019; 122: 350–364.
18. Yurek EE and Ozmutlu HC. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. Transp Res Part C Emerg Technol 2018; 91: 249–262.
19. Ghoseiri K and Ghannadpour SF. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. Appl Soft Comput 2010; 10(4): 1096–1107.
20. Cordeau JF and Maischberger M. A parallel iterated tabu search heuristic for vehicle routing problems. Comput Oper Res 2012; 39(9): 2033–2050.
21. Belhaiza S, Hansen P, and Laporte G. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput Oper Res 2014; 52: 269–281.
22. Zulvia FE, Kuo RJ, and Nugroho DY. A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products. J Clean Prod 2019; 242: 118428.
23. Goksal FP, Karaoglan I, and Altiparmak F. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Comput Oper Res 2013; 65(1): 39–53.
24. Yu B and Yang ZZ. An ant colony optimization model: the period vehicle routing problem with time windows. Transp Res Part E Logist Transp Rev 2011; 47(2): 166–181.
25. Ribeiro GM and Laporte G. An adaptive large neighborhood search heuristic for the capacitated vehicle routing problem. Comput Oper Res 2012; 39(3): 728–735.
26. Mousavi SM and Tavakkoli-Moghaddam R. A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain. J Manuf Syst 2013; 32(2): 335–347.
27. Chang YS and Lee HJ. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst Appl 2018; 104: 307–317.
28. Guerriero F, Surace R, Loscri V, et al. A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints. Appl Math Model 2014; 38(3): 839–852.
29. Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 1987; 35(2): 254–265.
30. Phan DH and Suzuki J. Evolutionary multiobjective optimization for the pickup and delivery problem with time windows and demands. Mobile Netw Appl 2016; 21(1): 175–190.
31. Chowdhury S, Emelougu A, Marufuzzaman M, et al. Drones for disaster response and relief operations: a continuous approximation model. Int J Prod Econ 2017; 188: 167–184.
32. Rabta B, Wankmüller C, and Reiner G. A drone fleet model for last-mile distribution in disaster relief operations. Int J Disaster Risk Reduct 2018; 28: 107–112.
33. Karaboga D and Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm. J Glob Optim 2007; 39(3): 459–471.
34. Karaboga D and Akay B. A comparative study of artificial bee colony algorithm. Appl Math Comput 2009; 214(1): 108–132.
51. Wang GG. Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Comput* 2018; 10(2): 151–164.

52. Wang GG, Deb S, Gandomi AH, et al. Opposition-based krill herd algorithm with Cauchy mutation and position clamping. *Neurocomputing* 2016; 177: 147–157.

53. Li JQ, Pan QK, and Liang YC. An effective hybrid tabu search algorithm for multi-objective flexible job shop scheduling problems. *Comput Ind Eng* 2010; 59(4): 647–662.

54. Zhao XC. A perturbed particle swarm algorithm for numerical optimization. *Appl Soft Comput* 2010; 10(1): 119–124.

55. Li JQ, Tao XR, Jia BX, et al. Efficient multi-objective algorithm for the lot-streaming hybrid flowshop with variable sub-lots. *Swarm Evol Comput* 2020; 52: 100600, doi:10.1016/j.swevo.2019.100600.

56. Sun J, Miao Z, Gong D, et al. Interval multiobjective optimization with memetic algorithms. *IEEE Trans Cybern* 2020; 1–14. https://doi.org/10.1109/TCYB.2019.2908485.

57. Ropke S and Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp Sci* 2006; 40(4): 455–472.

58. He B, Huang S, and Wang J. Product low-carbon design using dynamic programming algorithm. *Int J Precis Eng Manuf Green Technol* 2015; 2(1): 37–42.

59. He B, Zhang P, and Wang J. Automated synthesis of mechanisms with consideration of mechanical efficiency. *J Eng Design* 2014; 25(4-6): 213–237.

60. Li CD, Yi JQ, Wang HK, et al. Interval data driven construction of shadowed sets with application to linguistic word modelling. *Inf Sci* 2020; 507: 503–521.

61. Li JQ and Pan QK. Solving fuzzy job-shop scheduling problems by a hybrid optimization algorithm. *Journal of Mechanical Engineering* 2013; 49(23): 142–149.