Discrete Firefly Algorithm for Clustered Multi-Temperature Joint Distribution with Fuzzy Travel Times

Sichao Lu[1 *], **Xifu Wang** [2]

[1] *School of Traffic and Transportation, Beijing Jiaotong University,*
Beijing, 100044, China
E-mail: lusichao@163.com

[2] *School of Traffic and Transportation, Beijing Jiaotong University,*
Beijing, 100044, China
E-mail: wxfky@vip.sina.com

Received 9 February 2017

Accepted 15 October 2017

Abstract

This study proposes a mathematical model of clustered multi-temperature joint distribution in fuzzy environment. In this model, a Z-shaped function is used to depict customer satisfaction. For the imprecise model, triangular fuzzy numbers are used to represent travel times. By redefining the movement procedure of fireflies, two versions of discrete firefly algorithms are developed, which differ in the population initialization strategy. Finally, experiments are carried out and computational results are reported to confirm the effectiveness of the proposed algorithms.

Keywords: MTJD, distribution; Z-shaped; Firefly algorithm; Perishable food.

1. Introduction

Nowadays, many logistic enterprises have configured the food cold chain to maintain food quality, because refrigeration can reduce the deterioration rate of perishable food products. Consequently, optimization of perishable food delivery has been widely studied by academics and practitioners in the catering industry and the logistics industry. A comprehensive review of the literature shows that, many mathematical models which deal with this problem are based on the model of the vehicle routing problem (VRP)[1-6]. However, these models are constructed on the assumption that perishable food products with different temperature demands can be put together in a vehicle with a single temperature zone. Therefore, they are not applicable to the route optimization of multi-temperature joint distribution (MTJD), which is a novel approach to the distribution of perishable food products.

In recent years, due to rapid advances in e-business and web technologies, perishable food products such as vegetables and ice creams can be ordered and purchased conveniently through the Internet. As a result, the cold distribution market and the home delivery market have undergone a rapid expansion, which contributes to the development of multi-temperature joint distribution (MTJD). The MTJD approach can be divided into two types[7]. The first type is the mechanical refrigerated compartment division. This uses a technique which can divide a single vehicle compartment into different temperature zones. In the second type, standardized cold insulated boxes and cabinets are placed in regular vehicles to store perishable food products at varying temperatures. The MTJD system is able to reduce logistics costs significantly and attain customer satisfaction, as well as guaranteeing food safety and food quality[8]. However, only a few papers refer to route optimization and load planning in MTJD. Cho and

---

[*] Corresponding author.

Li[9] formulated the mathematical model of the multi-temperature refrigerated container vehicle routing problem and proposed a two-stage heuristic algorithm. Hsu et al.[10] modelled facilities planning for MTJD in a hierarchical hub-and-spoke network. Sun[11] proposed a bi-objective MTJD which aims at minimizing cost and total carbon emissions, and solved it by using the augmented ε-constraint method. Lu and Wang[12] put forward two suggestions for extending the mathematical model of MTJD to a further step. First, travel times can be defined as fuzzy numbers. Second, a Z-shaped curve can be used to depict the relationship between customer satisfaction and arrival time, rather than linear curves.

It is clear that MTJD is highly combinatorial. Consequently, exact algorithms which aim at obtaining a global optimal value may be far less efficient than heuristic algorithms for most medium-sized and large-sized problem instances. Hence, this paper attempts to develop an evolving algorithm to obtain near-optimal solutions. As a popular evolutionary algorithm which is inspired by the collective behaviour of fireflies, the firefly algorithm (FA) is applicable to almost all engineering areas[13]. This algorithm was first proposed by Yang, who claimed it was superior to genetic algorithms and particle swarm optimization[14]. Given that the original FA was designed to solve continuous optimization problems, some researchers have proposed new versions of the firefly algorithm to deal with discrete optimization problems such as the two-dimensional min-cost covering problem[15], the asymmetric and clustered vehicle routing problem with simultaneous pickups and deliveries[16] and the fuzzy cold storage problem[17].

This paper presents a study of a clustered MTJD, which uses cabins in conventional vehicles to distribute perishable food products. Section 2 gives the problem definition and the mathematical formulation of clustered MTJD. Section 3 presents the discrete firefly algorithm-based solution approaches in detail. In section 4, the computational performances of the proposed algorithms are evaluated, followed by a conclusion in section 5.

2. Mathematical Model

2.1. Fuzzy preliminaries

Since the clustered MTJD involves fuzziness, various fuzzy concepts are discussed briefly in this subsection.

2.1.1 LR-type fuzzy number

A fuzzy number $\widetilde{M}=(m, \alpha, \beta)_{LR}$ is called LR-type if there exist decreasing functions L (for left), R (for right), and real numbers $\alpha>0$, $\beta>0$ with m, whose membership function can be characterized by Eq. (1)[18].

$$\mu_{\widetilde{M}}(x)=\begin{cases} L\left(\dfrac{m-x}{\alpha}\right) & x\le m \\ R\left(\dfrac{x-m}{\beta}\right) & x\ge m \end{cases} \tag{1}$$

$\widetilde{M}$ is called positive if $\mu_{\widetilde{M}}(x)=0$, $\forall x<0$. Based on Eq.(1), an LR-type trapezoidal fuzzy number $\widetilde{T}=(m_1, m_2, \alpha, \beta)_{LR}$ can be defined as[19]:

$$\mu_{\widetilde{T}}(x)=\begin{cases} L\left(\dfrac{m_1-x}{\alpha}\right) & x\le m_1 \\ 1 & m_1\le x\le m_2 \\ R\left(\dfrac{x-m_2}{\beta}\right) & x\ge m_2 \end{cases} \tag{2}$$

2.1.2 Triangular fuzzy number

A triangular fuzzy number $\widetilde{A}$ can be defined by a triplet (a^l, a^c, a^r), where the membership function can be determined by:

$$\mu_{\widetilde{A}}(x)=\begin{cases} 0 & x<a^l \\ L(x)=1-\dfrac{a^c-x}{a^c-a^l} & a^l\le x\le a^c \\ R(x)=1-\dfrac{x-a^c}{a^r-a^c} & a^c\le x\le a^r \\ 0 & a^r<x \end{cases} \tag{3}$$

2.1.3 Graded mean integration representation

The graded mean integration representation (GMIR) method is a popular defuzzification technique and has many applications, such as defuzzifying the parameters of an economic order quantity model[20] and those of the shortest path problem[21]. Let L^{-1} and R^{-1} be the inverse functions of the functions $L(x)$ and $R(x)$, respectively. The graded mean h-level value of the LR-type

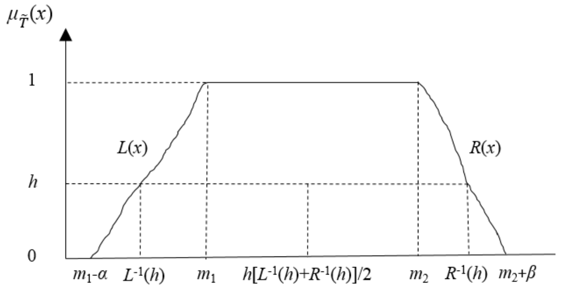


Fig. 1. The graded mean h-level value of $\widetilde{T}=(m_1, m_2, \alpha, \beta)$.

trapezoidal fuzzy number $\tilde{T}=(m_1, m_2, \alpha, \beta)_{LR}$ is $h[L^{-1}(h)+R^{-1}(h)]/2$ as in Fig. 1. Then the GMIR of $\tilde{T}$ is determined by[22]:

$$P(\tilde{T}) = \int_0^1 h[L^{-1}(h)+R^{-1}(h)]/2\,\mathrm{d}h \Big/ \int_0^1 h\,\mathrm{d}h \tag{4}$$

The positive triangular fuzzy number $\tilde{A}=(a^l, a^c, a^r)$ can be regarded as an LR-type trapezoidal fuzzy number $\tilde{A}=(a^c, a^c, a^c\text{-}a^l, a^r\text{-}a^c)_{LR}$. Thus, $L^{-1}(x)$ and $R^{-1}(x)$ of $\tilde{A}$ are given in Eq. (5) and Eq. (6) respectively.

$$L^{-1}(x) = (a^c - a^l)\,x + a^l \tag{5}$$

$$R^{-1}(x) = (a^c - a^r)\,x + a^r \tag{6}$$

Based on Eq. (4)-Eq. (6), the GMIR of $\tilde{A}$ can be acquired by using Eq. (7)

$$\begin{aligned} P(\tilde{A}) &= \int_0^1 h[L^{-1}(h)+R^{-1}(h)]/2\,\mathrm{d}h \Big/ \int_0^1 h\,\mathrm{d}h \\ &= \int_0^1 h[(a^c - a^l)\,h + a^l+(a^c - a^r)h + a^r]/2\,\mathrm{d}h \Big/ \int_0^1 h\,\mathrm{d}h \\ &= \int_0^1 (a^ch^2 - a^lh^2+a^lh+a^rh+a^ch^2 - a^rh^2)/2\,\mathrm{d}h \Big/ \int_0^1 h\,\mathrm{d}h \\ &= \frac{1}{2}\left(\frac{a^c}{3}-\frac{a^l}{3}+\frac{a^l}{2}+\frac{a^r}{2}+\frac{a^c}{3}-\frac{a^r}{3}\right)\Big/\frac{1}{2} = \frac{a^l+4a^c+a^r}{6} \end{aligned} \tag{7}$$

2.1.4 Representation of operations between a triangular fuzzy number and a crisp number

Let $\tilde{A}=(a^l, a^c, a^r)$ and $\tilde{B}=(b^l, b^c, b^r)$ be two triangular fuzzy numbers and let k be a positive crisp number. According to the fuzzy arithmetical operations under the function principle, the addition operation $\oplus$, the subtraction operation $\ominus$, the multiplication operation $\otimes$, and the division operation $\oslash$ on $\tilde{A}$ and $\tilde{B}$ are expressed as[20,23]:

$$\tilde{A} \oplus \tilde{B} = (a^l+b^l, a^c+b^c, a^r+b^r) \tag{8}$$

$$\tilde{A} \ominus \tilde{B} = (a^l - b^r, a^c - b^c, a^r - b^l) \tag{9}$$

$$\tilde{A} \otimes \tilde{B} = (a^lb^l, a^cb^c, a^rb^r) \tag{10}$$

$$\tilde{A} \otimes k = (ka^l, ka^c, ka^r) \tag{11}$$

$$\tilde{A} \oslash \tilde{B} = (a^l/b^r, a^c/b^c, a^r/b^l) \tag{12}$$

A crisp number k can be regarded as (k, k, k). Therefore, based on the GMIR, the addition operation and the subtraction operation between a triangular fuzzy number $\tilde{A}=(a^l, a^c, a^r)$ and a crisp number b are defined as:

$$P(\tilde{A} \oplus k) = P[(a^l+k, a^c+k, a^r+k)] = \frac{a^l+4a^c+a^r}{6} + k \tag{13}$$

$$P(\tilde{A} \ominus k) = P[(a^l - k, a^c - k, a^r - k)] = \frac{a^l+4a^c+a^r}{6} - k \tag{14}$$

2.1.5 Ranking of triangular fuzzy numbers and crisp numbers by GMIR

The problem of ordering fuzzy numbers has been addressed by many researchers and consequently many fuzzy ranking methods exist. In this paper, the GMIR method is used to define the ranking of triangular fuzzy numbers and crisp numbers.

Let $\tilde{A}=(a^l, a^c, a^r)$ be a triangular fuzzy number and k be a crisp number. Based on the GMIR method, the ranking number has the following properties:

$P(\tilde{A}) > k$ if and only if $\tilde{A} \succ k$
$P(\tilde{A}) \geq k$ if and only if $\tilde{A} \succsim k$
$P(\tilde{A}) = k$ if and only if $\tilde{A} \sim k$
$P(\tilde{A}) < k$ if and only if $\tilde{A} \prec k$
$P(\tilde{A}) \leq k$ if and only if $\tilde{A} \precsim k$

The notations $\tilde{A} \succ k$, $\tilde{A} \succsim k$, $\tilde{A} \sim k$, $\tilde{A} \prec k$, $\tilde{A} \precsim k$ signify that $\tilde{A}$ has a higher ranking than k, at least the same ranking as k, the same ranking as k, a lower ranking than k and at most the same ranking as k respectively. It is also easy to determine the ranking orders among multiple triangular fuzzy numbers in a similar way.

2.2. Representation of the customer satisfaction value

For the distribution of durable goods such as books and consumer electronic devices, many VRP models aim to minimize the distribution costs and use the time window to represent the customers' time constraints. However, short product lives and fast transportation are two characteristics of the supply chain for fresh and perishable foods[4]. Chen suggested using an S-shaped curve to depict the relationship between the freshness of perishable food products and profit[24]. Therefore, a Z-shaped curve is adopted here to depict the relationship between customer satisfaction and arrival time as was suggested in Ref. 12.

Fig. 2 gives a graphical method for illustrating this curve. For example, let a_i=2 and b_i=6. When the arrival time of the vehicle to customer i is within two hours, the customer will be very satisfied. As the arrival time passes two hours, the customer satisfaction value (*csv*) starts to decrease. After six hours, the customer will be very dissatisfied with the delivery.

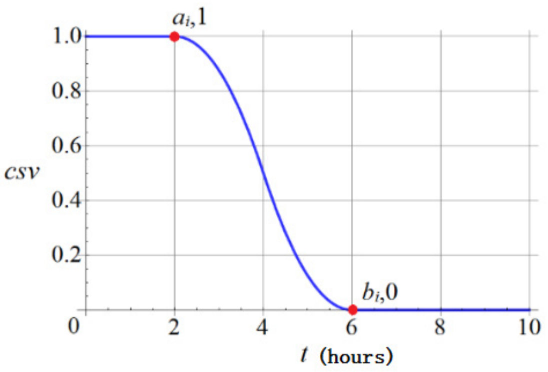


Fig. 2. Z-Shaped customer satisfaction function.

To describe the Z-shaped curve mathematically, Eq. (1) is used to depict the customer satisfaction value based on the built-in membership function in Matlab[25].

$$\mathrm{CSV}(t)=\begin{cases} 1 & t \le a_i \\ 1-2\left(\dfrac{t-a_i}{b_i-a_i}\right)^2 & a_i \le t \le \dfrac{a_i+b_i}{2} \\ 2\left(\dfrac{t-b_i}{b_i-a_i}\right)^2 & \dfrac{a_i+b_i}{2} \le t \le b_i \\ 0 & t \ge b_i \end{cases} \tag{15}$$

2.3. Notations and model formulation

The proposed model consists of the following indices, parameters, and decision variables:

- An index k for the set of m vehicles $\boldsymbol{V}=\{v_1, v_2,\ldots, v_m\}$.
- Indices h, i, j for a set of n nodes with customers $\boldsymbol{C}=\{c_1, c_2,\ldots, c_n\}$.
- $\boldsymbol{C}$ is divided into u mutually exclusive nonempty subsets $\boldsymbol{C}=\{\boldsymbol{C}_1, \boldsymbol{C}_2,\ldots, \boldsymbol{C}_u\}$, each subset representing a cluster of customers.
- $\boldsymbol{C}^{+}=\boldsymbol{C} \cup \{c_0\}$ (where c_0 represents the distribution centre).
- Perishable food products are broadly categorized into s types according to their temperature demands for storage. Correspondingly, there should be s types of cabins with different temperature values. The index z is used for the set of temperature values $\boldsymbol{E}=\{e_1, e_2,\ldots, e_s\}$.
- Customer demand for z types of perishable food products for customer i is d_i^z.
- The load capacity of each cabin is q.
- The vehicle capacity is denoted by the set $\boldsymbol{B}$. For example, vehicle k can accommodate B_k cabins.
- p_k^z cabins are used to store the z type of perishable food products in vehicle k.
- The satisfaction value of the customer i is csv_i, which is generated using Eq. (15). The left extreme and right extreme of the i-th customer satisfaction curve are a_i and b_i respectively.
- Given that the travel time between customer i and customer j cannot be predicted precisely[12,26], it is considered as a triangular fuzzy number $\tilde{t}_{ij}=(t_{ij}^{l}, t_{ij}^{c}, t_{ij}^{r})$.
- The moment when customer i begins to be served by any vehicle is denoted by the time variable t_i (where t_0 represents the time when vehicles depart from the distribution centre).
- Service time at node i is denoted by w_i.
- M is a large positive number.
- If there exists a vehicle that travels from customer i to customer j ($i \neq j$), then $x_{ij}^k=1$; otherwise $x_{ij}^k=0$.
- If customer i is served by vehicle k, then $y_i^k=1$; otherwise $y_i^k=0$.

The mathematical formulation for the clustered MTJD can be stated as follows:

$$\max Z=\textstyle\sum_{i=1}^{n} \mathrm{CSV}(t_i) \tag{16}$$

s.t.

$$\textstyle\sum_{k=1}^{m}\sum_{i=1}^{n} x_{ij}^k=1, \qquad \forall j \in \boldsymbol{C} \tag{17}$$

$$\textstyle\sum_{k=1}^{m}\sum_{i=1}^{n} x_{ji}^k=1, \qquad \forall j \in \boldsymbol{C} \tag{18}$$

$$\textstyle\sum_{i=1}^{n} x_{0i}^k=1, \qquad \forall k \in \boldsymbol{V} \tag{19}$$

$$\textstyle\sum_{i=1}^{n} x_{i0}^k=1, \qquad \forall k \in \boldsymbol{V} \tag{20}$$

$$\textstyle\sum_{k=1}^{m} y_i^k=1, \qquad \forall i \in \boldsymbol{C} \tag{21}$$

$$y_i^k y_i^k \ge x_{ij}^k, \qquad \forall i,j \in \boldsymbol{C}, \forall k \in \boldsymbol{V}\, \forall k \tag{22}$$

$$\textstyle\sum_{i=1}^{n} x_{0i}^k\, y_i^k=1, \qquad \forall k \in \boldsymbol{V} \tag{23}$$

$$\textstyle\sum_{i=1}^{n} x_{i0}^k\, y_i^k=1, \qquad \forall k \in \boldsymbol{V} \tag{24}$$

$$\textstyle\sum_{i=0}^{n} x_{ih}^k - \sum_{j=0}^{n} x_{hj}^k=0, \qquad \forall h \in \boldsymbol{C}, \forall k \in \boldsymbol{V} \tag{25}$$

$$\tilde{t}_{0i} \precsim t_j, \qquad \exists i \in \boldsymbol{C}, \forall j \in \boldsymbol{C} \tag{26}$$

$$t_i+w_i \oplus \tilde{t}_{ij} \ominus M(1-x_{ij}^k) \le t_j, \qquad \forall i \in \boldsymbol{C}^{+}, \forall j \in \boldsymbol{C}, \forall k \in \boldsymbol{V}, i \neq j \tag{27}$$

$$\textstyle\sum_{z=1}^{s} p_k^z \ge s, \qquad \forall k \in \boldsymbol{V} \tag{28}$$

$$\textstyle\sum_{i=1}^{n} d_i^z y_i^k \le p_k^z q, \qquad \forall k \in \boldsymbol{V}, \forall z \in \boldsymbol{E} \tag{29}$$

$$\textstyle\sum_{z=1}^{s} p_k^z \le B_k, \qquad \forall k \in \boldsymbol{V} \tag{30}$$

$$\boldsymbol{C}_i \cap \boldsymbol{C}_j=\emptyset \qquad \forall i,j,\ i \neq j \tag{31}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall i,j \in \boldsymbol{C}^{+}, \forall k \in \boldsymbol{V} \tag{32}$$

$$y_i^k \in \{0,1\}, \qquad \forall i \in \boldsymbol{C}, \forall k \in \boldsymbol{V} \qquad (33)$$

$$t_i \geq 0 \qquad \forall i \in \boldsymbol{C}^+ \qquad (34)$$

The objective function (16) aims to maximize the sum of the customer satisfaction value. Eq. (17) and Eq. (18) ensure that all customers are served only once. Eq. (19) and Eq. (20) indicate that each route starts and ends at the distribution centre and only one trip is allowed. Eq. (21) states that each customer is served by only one vehicle. Eqs. (22)-(24) are used to represent the relationship between x_{ij}^k and y_i^k. Eq. (25) guarantees that flow conservation is satisfied. Eq. (26) and Eq. (27) guarantee the consistency of time variables. Eq. (28) indicates that each vehicle accommodates all types of cabins. Eq. (29) guarantees that the total capacity of cabins in any vehicle for accommodating the z type of perishable food products is not exceeded. Eq. (30) ensures that the aggregated capacity of refrigerated containers does not exceed the loading capacity of the vehicle. Eq. (31) guarantees that each customer node belongs to only one cluster. Eqs. (32)-(34) give range constraints on the decision variables.

3. Solution Approaches

3.1. Coding method

Each firefly represents a possible solution. For any firefly, its position can be denoted as the set $\{(v_k, c_i, t_i) | i \in \boldsymbol{C},\ k \in \boldsymbol{V}\}$. This indicates that the elapsed time when vehicle v_k arrives at customer node c_i is t_i. The light intensity of firefly i is denoted by I_i, which is the total customer satisfaction value.

3.2. Fuzzification and defuzzification

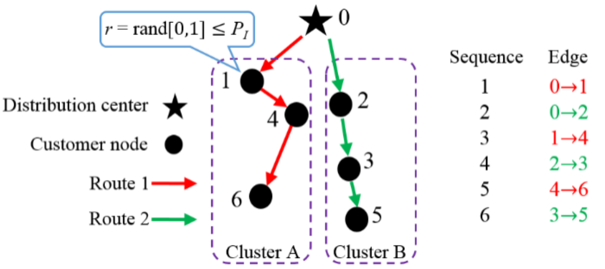
Based on Eq. (7), the GMIR of the fuzzy travel time $\tilde{t}_{ij} = (t_{ij}^l, t_{ij}^c, t_{ij}^r)$ is defined via Eq. (35).

$$P(\tilde{t}_{ij}) = \frac{1}{6}[t_{ij}^l + 4t_{ij}^c + t_{ij}^r] \qquad (35)$$

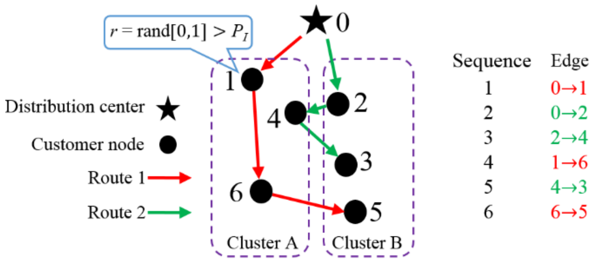
After the algorithm reads the instance data, t_{ij}^c is generated via the distance between node i and node j divided by the speed. Next, t_{ij}^l and t_{ij}^r are generated based on the value of t_{ij}^c. The relationship among the three numbers can be determined by the decision maker. Finally, the triangular fuzzy number $\tilde{t}_{ij} = (t_{ij}^l, t_{ij}^c, t_{ij}^r)$ is defuzzified by applying Eq. (35).

3.3. Population initialization

In many meta-heuristic algorithms including some variants of the firefly algorithm, the initial population is generated randomly[16,27,28]. Thus, *pop_size* is defined as the population size, and all fireflies $F = \{F_0, F_1, \ldots, F_{pop\_size\text{-}1}\}$ are generated at random. Although the initial population of reasonably structured solutions lacks sufficient diversity, there is also a possibility that it could generate high-quality near-optimal values[29]. For example, Refs. 15 and 17 choose items with high value density to generate initial solutions. Therefore, Algorithm 1 is designed to generate the initial firefly population with structured solutions. Additionally, experiments are carried out with the initial fireflies generated both randomly and by using Algorithm 1 for the purpose of comparison.



(a) Situation under $r \leq P_I$



(b) Situation under $r > P_I$

Fig. 3. Examples of population initialization.

Algorithm 1

```
for $i$ =0: $pop\_size$-1
    Allocate $m$ neighbouring nodes of the distribution centre from different clusters to each vehicle respectively;
    while not all of the customer nodes are visited
        for $j$=0: $m$-1
            if $r$=rand[0,1] ≤ $P_I$&&all constraints are satisfied
                Allocate the nearest unvisited customer node to vehicle $j$;
            end if
        end for
    end while
Initialize light intensity $I_i$;
end for $i$
Find the current brightest firefly $F_{opt}$;
```

To better present Algorithm 1, two examples are shown in Fig. 3. As shown in Fig. 3(*a*), for node 1, if the random number r is less than or equal to P_I, an edge is generated from itself to its nearest neighbour - node 4. Fig. 3(*b*) shows the opposite situation. For node 1, if the random number r is larger than P_I, it will give up connecting to its nearest neighbour and wait for the next turn. Thus, node 4 will be connected with node 2 and therefore a solution which differs from that in Fig. 3(a) will be generated. From the above-mentioned presentation, it can easily be inferred that the uniform random variable P_I can prevent the generation of multiple duplicate solutions.

3.4. Movement of fireflies

Given that discrete optimization problems cannot be solved by the original firefly algorithm which was designed for coping with continuous optimization problems, several versions of the discrete firefly algorithm have been proposed[15-17]. In this algorithm, the arrival time series are used to measure the distance between two fireflies. For example, let $\boldsymbol{t}^i=(t_1^i,\ldots,t_n^i)$ be the arrival time series vector of firefly i. In this sequence, the arrival time of a vehicle to customer h is denoted by t_h^i. Thus, the distance between any two fireflies F_i and F_j is determined by

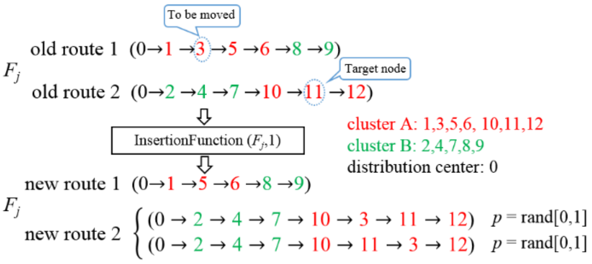
$$r_{ij}=\left|\sum_{h=1}^{n}(t_h^j - t_h^i)\right| \tag{36}$$

In the basic firefly algorithm, γ denotes the light absorption coefficient. It is crucial to the convergence speed of the algorithm[13]. Based on Eq. (36), the movement of firefly F_j toward firefly F_i is determined by

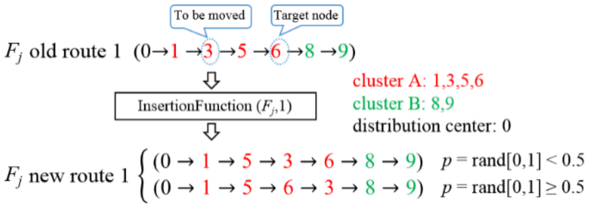
$$n=\lceil r_{ij}/\gamma \rceil \tag{37}$$

$$F_j=\text{InsertionFunction}\ (F_j, n) \tag{38}$$

Eq. (37) indicates that the length of the movement of a firefly will be the smallest integral value that is greater than or equal to r_{ij}/γ. The InsertionFunction is similar to that given in Ref. 16. It selects and extracts one random node from a random route. After that, this node is randomly moved beside a node for which the cluster is same as the extracted one. Fig. 4 shows two examples of how the InsertionFunction works. In Fig. 4(*a*), node 3 is randomly selected to be moved. Node 11, which belongs to same cluster, is randomly selected as the target node. Next, node 3 is moved to the left of node 11 if the random variable p is less than or equal to 0.5; otherwise it is moved to the right of node 11. As shown in Fig. 4(*b*), if the extracted node and the target node belong to the same route, the method is also viable.


To be moved
old route 1 (0→1 →3→5 →6 →8 →9)
F_j
Target node
old route 2 (0→2 →4 →7 →10 →11 →12)
InsertionFunction (F_j,1)
cluster A: 1,3,5,6, 10,11,12
cluster B: 2,4,7,8,9
distribution center: 0
new route 1 (0→1 →5 →6 →8 →9)
F_j
new route 2
(0 → 2 → 4 → 7 → 10 → 3 → 11 → 12) p = rand[0,1]
(0 → 2 → 4 → 7 → 10 → 11 → 3 → 12) p = rand[0,1]

(*a*) operations when the selected node and the targeted node belong to different routes.


To be moved
Target node
F_j old route 1 (0→1 →3→5 →6→8 →9)
InsertionFunction (F_j,1)
cluster A: 1,3,5,6
cluster B: 8,9
distribution center: 0
F_j new route 1
(0 → 1 → 5 → 3 → 6 → 8 → 9) p = rand[0,1] < 0.5
(0 → 1 → 5 → 6 → 3 → 8 → 9) p = rand[0,1] ≥ 0.5

(*b*) operations when the selected node and the targeted node belong to the same route.

Fig. 4. Examples of the InsertionFunction.

Instead of using a repair operator, capacity constraint is taken into account to avoid generating infeasible solutions. This function is repeated n times to enable a movement of firefly F_j. If the new position is superior to the original position, then the firefly is updated.

If F_j is the firefly with the largest light intensity, it moves randomly[30] in the basic firefly algorithm. Tilahun and Ong pointed out that this behaviour may lead to a reduction in the algorithm performance, because the light intensity of the brightest firefly may decrease in certain directions. They suggested randomly choosing a direction in which the brightness of the brightest firefly increases if the firefly moves in that direction. If there exists no such direction, the brightest firefly must stay in its current position[31]. However, the brightest firefly F_j will cease moving in the discrete firefly algorithm proposed here, because r_{jj}=0 and consequently n=0 by applying Eq. (36) and Eq. (37). Therefore, n=1 is set in Eq. (38) for the brightest firefly.

3.5. Repair operation

Sometimes, several customer nodes may exist which are at the end of the route with a *csv* of zero. The approach mentioned above may generate an impractical route for these, because the visit order does not affect the objective value of the solution. Considering the two routes in Fig.

5 for example, it can be seen that the distance of route 1 is clearly longer than that of route 2, but the sum of *csv* is the same for the two routes. Therefore, route 1 should be modified by reordering the visiting sequence.

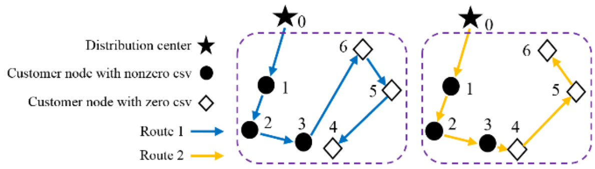

Distribution center ★
Customer node with nonzero csv ●
Customer node with zero csv ◇
Route 1
Route 2
0
1
2
3
4
5
6
0
1
2
3
4
5
6

Fig. 5. An example of the repair operation.

Assume the *csv* of the successor of node *h* is zero and the index of the route to be repaired is *k*. To remedy the problem mentioned above, Algorithm 2 is proposed and presented as follows.

Algorithm 2

```
Set $J=\{j \mid \mathrm{CSV}(t_j)=0, y_j^k=1\}$;
Set $p = h, y_j^k=0, x_{ij}^k=0, t_j=0, \forall i \in C, \forall j \in J$;
repeat
    Randomly select an element $s$ from $\{j \mid t_{pj} \le t_{pl}, \forall l \in J, \exists j \in J\}$;
    Connect node $p$ and node $s$;
    Calculate $t_s$; Set $x_{ps}^k=1$; Set $y_s^k=1$; Set $p = s$; Set $J=J-\{s\}$;
until $J=\emptyset$
```

3.6. Procedure for the discrete firefly algorithm

As in the basic firefly algorithm, the movement procedure is repeated until the best solution remains unchanged over *rep_gen* iterations. To sum up, the pseudocode of the proposed discrete firefly algorithm (DFA) can be summarized according to Algorithm 3.

Algorithm 3

```
Read instance data (customer locations, customer demands, vehicle capacity, travel times, etc.);
Set the parameters of the discrete firefly algorithm;
for $i$=0: $n$
    for $j$=0: $n$
        if $i \ne j$
            Fuzzify $t_{ij}^c$ and defuzzify $\tilde{t}_{ij}$ using Eq. (35);
        else
            $t_{ij}=0$;
        end if
    end for
end for
Initialize the firefly populations;
repeat
    for $i$=0: $pop\_size$-1
        for $j$ =0: $pop\_size$-1
            if $i \ne j$ && $I_i > I_j$
                Move $F_j$ using Eqs. 36-38;
            end if
        end for
    end for
    Find $F_{opt}$;
    Move $F_{opt}$ using Eq. (38) ($n$=1);
until $F_{opt}$ remains unchanged over $rep\_gen$ iterations
for $k$=0: $m$-1
    if a node with zero $csv$ is found in route $k$
        Repair $F_{opt}$ using Algorithm 2;
    end if
end for
```

4. Simulation Experiment

In this section, experiments are carried out to validate the proposed mathematical formulation and algorithm. All computational experiments are coded with Visual C# in a PC with a 3.2GHz Intel Pentium G3420 processor and 8GB RAM.

4.1. Data generation

In the experiments, problems are generated to evaluate the performance of the proposed algorithm. In each instance, the values s=5, q=160L, γ =50, t_0=0, *rep_gen*=2000 and speed=500m/minute are set. All the customer nodes have corresponding (x,y) coordinates, where both x and y are between -25000m and 25000m. Parameters w_i , a_i , and b_i which refer to time are measured in minutes. In the numerical example about joint distribution in Ref. 10, perishable food products are measured in litres and are categorized into five groups according to their temperature ranges[10]: (1) tuna, (2) ice cream, ice cubes, and frozen dumplings, (3) milk and juice; (4) biscuits and medicines and (5) lunch boxes and hot meals. Therefore, s is set at a value of 5 and d_i^z is measured in litres in this example. According to general consumer behaviour, customer demand for each type of perishable food product is generated with a uniform distribution, where $d_i^1 \sim U(0,5)$, $d_i^2 \sim U(0,8)$, $d_i^3 \sim U(0,10)$, $d_i^4 \sim U(0,15)$ and $d_i^5 \sim U(0,10)$. As a matter of experience, it may be satisfactory for many consumers to receive the food products within about two-and-a-half hours, and a delivery time of over about six hours could result in customer dissatisfaction. Given that customer attitudes differ with regard to delivery time, the variances of the customer satisfaction parameters are at 900, which can take into account the diversity of customer attitudes appropriately. Therefore, the parameters a_i and b_i are generated with normal distributions *N*(150,900) and *N*(360,900) respectively. Given that five minutes or so is enough time for serving a consumer, the parameter w_i is generated with normal distribution *N*(5,1). Travel times are fuzzified as triangular fuzzy numbers with $t_{ij}^{l}=0.9\,t_{ij}$ and $t_{ij}^{r}=1.1\,t_{ij}$. The starting point for each vehicle is (0,0). Given that 10-25 fireflies can deal with most applications[32], *pop_size* was set to 20.

4.2. Experimental results

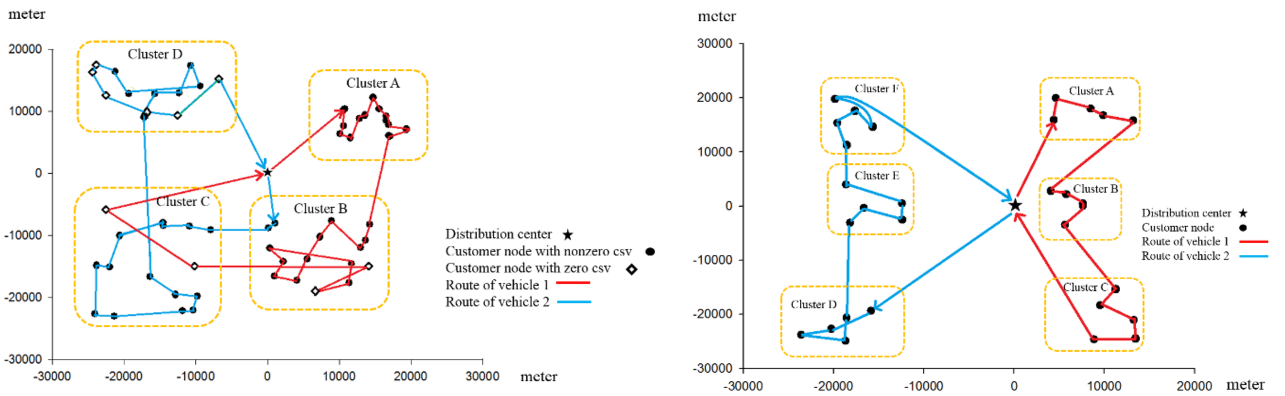
The proposed model and algorithm was applied to 22 instances. Each of these instances was run 10 times. The experimental results are shown in Table 1. DFA$_1$ was implemented with a randomly generated initial population, and DFA$_2$ was implemented with an initial population generated by Algorithm 1. Since that the mathematical model of the clustered MTJD is first proposed in this paper, there are no pre-existing dedicated algorithms. Given that Algorithm 1 is a type of greedy algorithm which connects the current node with the nearest unvisited node at each stage when P_I is set to 100%, it can be used as the benchmark for evaluating the effectiveness of the solution approaches, because a greedy algorithm is simple to develop and can solve many applications including the travelling salesman problem and the vehicle scheduling problem[33].

According to the results presented in Table 1, DFA$_1$ outperforms DFA$_2$ in terms of the best value for all the instances except for instances 2, 19, 21 and 22. The median values obtained by DFA$_2$ are better than those generated by DFA$_1$ for instances with 120 customer nodes, except for instance 18. All the solutions obtained by DFA$_2$ are better than those obtained by the greedy algorithm, but the worst value of 10 instances and the mean value of three instances obtained by DFA$_1$ are even worse than those obtained by the greedy algorithm. Therefore, it can be inferred that DFA$_1$ is suitable for solving small and medium-sized problem instances. It is also worth mentioning that DFA$_2$ presents stronger robustness than DFA$_1$ in general, as shown by the standard deviation values. Additionally, for customers with identical numbers and clusters, more vehicles contribute to higher customer satisfaction values, which partially proves the effectiveness of the algorithms developed in this paper.

Fig. 6 (some dots are overlapping) and Fig. 7 present detailed results of several instances, in order to further investigate the effectiveness of the proposed algorithm. Fig. 8 and Table 2 reveal the efficiency of the algorithm over instances 6 and 16. It can be clearly observed that a larger population size consumes more computational time for both versions of the discrete firefly algorithm proposed in this paper.

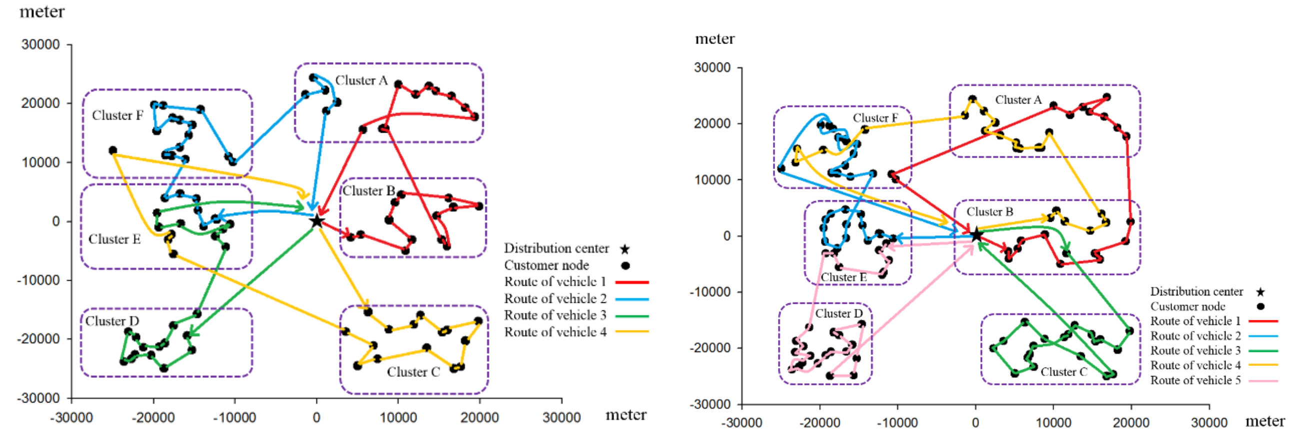
Table 1. Computational results and comparisons for the three algorithms in 22 problem instances.

No.	*n*	*u*	*m*	*B*	DFA$_1$				DFA$_2$				Greedy Algorithm
					best	median	worst	Std.dev	best	median	worst	Std.dev	
1	30	4	1	(6)	**14.320**	12.877	12.067	0.903	13.130	12.960	12.097	0.465	11.795
2	30	4	2	(5,5)	27.245	26.553	24.757	1.020	**27.389**	27.245	26.385	0.302	25.021
3	30	6	1	(8)	**14.680**	13.156	11.330	1.335	14.059	14.055	14.050	0.005	14.037
4	30	6	2	(5,5)	**26.921**	25.641	24.564	0.901	26.908	26.630	26.524	0.157	20.083
5	60	4	1	(15)	**23.509**	22.237	21.427	0.680	19.216	19.092	18.968	0.131	18.765
6	60	4	2	(8,7)	**40.476**	36.761	33.766	1.792	39.605	38.583	36.461	1.270	34.015
7	60	4	3	(5,5,6)	**53.864**	52.146	48.698	1.879	51.402	50.530	49.011	0.721	45.517
8	60	6	1	(15)	**21.217**	20.458	18.861	0.644	20.869	20.869	20.568	0.101	20.557
9	60	6	2	(8,7)	**42.481**	37.911	34.768	2.145	42.014	38.515	37.865	1.241	37.428
10	60	6	3	(5,5,6)	**53.565**	51.927	48.908	1.662	53.293	52.196	51.212	0.605	47.083
11	90	4	2	(11,10)	**49.032**	47.215	46.343	0.796	45.525	44.247	43.846	0.601	40.681
12	90	4	3	(7,7,8)	**68.326**	66.570	61.686	1.928	63.795	61.877	58.119	1.809	52.656
13	90	4	4	(6,6,6,6)	**84.826**	82.958	77.088	2.371	80.670	78.474	78.040	0.985	70.656
14	90	6	2	(11,10)	**42.913**	40.997	38.602	1.443	41.073	39.747	39.737	0.420	39.202
15	90	6	3	(7,7,8)	**58.706**	58.011	57.384	0.452	58.303	57.374	54.761	1.165	52.687
16	90	6	4	(6,6,6,6)	**78.424**	72.977	69.619	2.981	77.733	76.826	74.740	1.088	69.922
17	120	4	3	(8,8,9)	**71.352**	67.656	60.933	2.935	69.584	68.273	64.361	2.019	58.110
18	120	4	4	(7,7,7,7)	**97.067**	92.987	88.357	2.245	88.341	85.315	82.112	1.244	71.950
19	120	4	5	(5,6,6,6,6)	109.326	104.839	97.800	3.205	**110.084**	107.392	105.322	1.611	85.931
20	120	6	3	(8,8,9)	**66.527**	62.470	58.152	2.270	64.457	63.261	62.307	0.874	60.261
21	120	6	4	(7,7,7,7)	84.652	80.586	76.911	1.461	**88.977**	87.367	85.957	0.599	82.972
22	120	6	5	(5,6,6,6,6)	105.389	102.865	92.887	3.660	**105.982**	103.702	100.049	1.464	93.592



(*a*) best solution in instance 4.

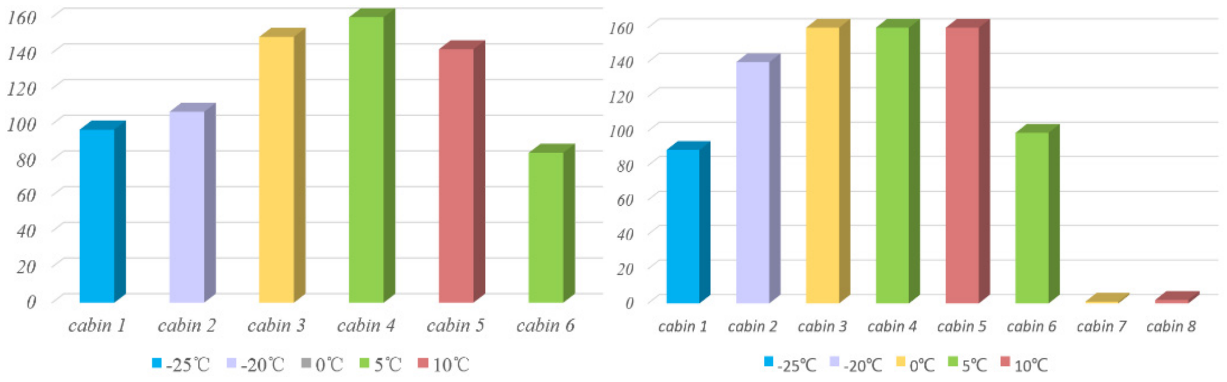
(*b*) best solution in instance 6.



(*c*) best solution in instance 16.

(*d*) best solution in instance 22.

Fig. 6. Graphical description for best operation routes over four instances with *pop_size*=20.



(*a*) vehicle v_1.

(*b*) vehicle v_2.

Fig. 7. Best vehicle load planning results over No.6 instance.

Table 2. Small study of solution quality and average runtime of two instances with different population size.

Algorithm	No.	pop_size	Z_{best}	Z_{median}	t_{meidan}(s)
DFA$_1$	6	10	40.906	37.128	105
DFA$_1$	6	20	40.476	36.761	336
DFA$_1$	6	30	41.617	38.837	746
DFA$_1$	6	40	41.622	38.152	1089
DFA$_1$	16	10	77.842	73.152	167
DFA$_1$	16	20	78.424	72.977	406
DFA$_1$	16	30	78.179	72.889	709
DFA$_1$	16	40	78.896	74.685	1478
DFA$_2$	6	10	39.605	39.023	68
DFA$_2$	6	20	39.605	38.583	203
DFA$_2$	6	30	41.564	39.384	444
DFA$_2$	6	40	41.082	37.701	656
DFA$_2$	16	10	78.025	76.288	138
DFA$_2$	16	20	77.733	76.826	595
DFA$_2$	16	30	77.877	76.261	785
DFA$_2$	16	40	78.257	77.261	1739

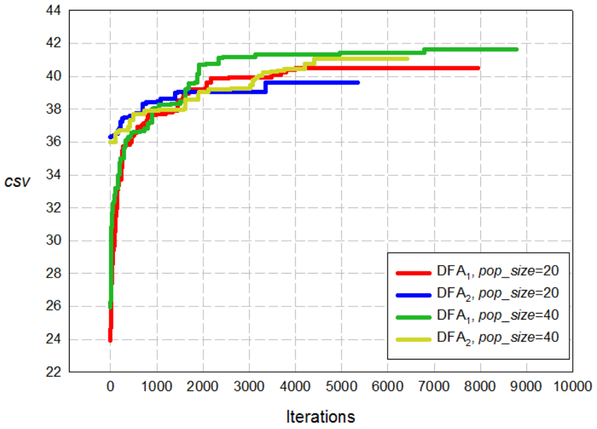


Fig. 8. Convergence curves of the discrete firefly algorithms over instance 6.

5. Conclusions

In this paper, a mathematical model for a clustered multi-temperature joint distribution with fuzzy travel times was proposed. It can be applied to solve the distribution of perishable food products with different temperature demands. In contrast to existing MTJD models, the proposed model aims at maximizing the customer satisfaction value by using a Z-shaped function, regards travel times as triangular fuzzy numbers, and considers a heterogeneous set of vehicles with different capacities.

Due to the complexity of the problem, the paper also considers the design of two versions of the discrete firefly algorithm which differ in the population initialization method but share the same movement scheme for fireflies. To validate the two algorithms, 22 problem instances are considered. The simulation indicates that the discrete firefly algorithms developed can effectively solve the clustered MTJD.

Future research may consider integrating the clustered MTJD with inventory optimization, time-varying networks, or the hub location problem. Sensitivity analysis can be conducted by comparing fuzzy parameters and optimal solutions in alternative defuzzification techniques such as graded k-preference integration and α-cuts. Establishing multiple objectives to increase the practicability of this model is also a possible direction for future research.

Acknowledgement

This research was funded by Beijing Jiaotong University scientific research projects T14L00360 and T14L00070.

References

1. H.K. Chen, C.F. Hsueh and M.S. Chang, Production scheduling and vehicle routing with time windows for perishable food products, *Comput. Oper. Res.* **36**(7) (2009) 2311–2319.
2. J. Brito, F.J. Martinez, J.A. Moreno and J.L. Verdegay, Fuzzy optimization for distribution of frozen food with imprecise times, *Fuzzy Optim. Decis. Ma.* **11**(3) (2012) 337–349.
3. D. Xu and R. Xiao, Modelling and intelligent solving of foodstuff distribution VRP based on disruption management, *Int. J. Comput. Appl. Tech.* **44**(2) (2012) 80–87.
4. K. Govindan, A. Jafarian, R. Khodaverdi and K. Devika, Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food, *Int. J. Prod. Econ.* **152**(2) (2014) 9–28.
5. P. Amorim and B. Almada-Lobo, The impact of food perishability issues in the vehicle routing problem, *Comput. Ind. Eng.* **67** (2014) 223–233.
6. B.D. Song and Y.D. Ko, A vehicle routing problem of both refrigerated- and general-type vehicles for perishable food products delivery, *J. Food Eng.* **169** (2016) 61–71.
7. C.I. Hsu and K.P. Liu, A model for facilities planning for multi-temperature joint distribution system, *Food Control* **22**(12) (2011) 1873–1882.

8. J.C. Kuo and M.C. Chen, Developing an advanced Multi-Temperature Joint Distribution System for the food cold chain, *Food Control* **21**(4) (2010) 559–566.
9. Y.J. Cho and C.C. Li, Application of Multi-temperature Refrigerated Container to Improve the Distribution of Cold Logistics, *J. East. Asia Soc. Transp. Stud.* **6** (2005) 2794–2808.
10. C.I. Hsu, W.T. Chen and W.J. Wu, Optimal delivery cycles for joint distribution of multi-temperature food, *Food Control* **34**(1) (2013) 106–114.
11. H. Sun, A Model for Vehicle Routing Problem for Multi-Temperature Joint Distribution System with Carbon Emission Constraints, in *Proc. Int. Conf. Logistics Engineering, Management and Computer Science* (Atlantis, Paris, 2015), pp.1022-1026.
12. S. Lu and X. Wang, From a Literature Review to a Theoretical Framework for Cloud-Based Internet of Things and Optimization Enabled Perishable Food Cold Chain Management, in *Advances in Energy Science and Equipment Engineering II*, (CRC, Leiden, 2017). (In Press)
13. I. Fister, I. Jr. Fister, X.S. Yang and J. Brest, A comprehensive review of firefly algorithm, *Swarm Evol. Comput.* **13** (2013) 34–46.
14. X.S. Yang, Firefly algorithms for multimodal optimization, *Stochastic Algorithms: Foundations and Applications.* (Springer, Berlin, 2009), pp.169–178.
15. S. Lu and X. Wang, A Firefly Algorithm Based Approach for the Two-dimensional Min-Cost Covering Problem, in *Proc. 2015 IEEE 6th Int. Conf. Software Engineering and Service Sciences*, eds. M.S.P. Babu and W. Li (IEEE, NewYork, 2015), pp. 1003–1006.
16. E. Osaba, X.S. Yang, F. Diaz, E. Onieva, A.D. Masegosa and A. Perallos, A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy, *Soft Comput.* (2016). (In Press)
17. S. Lu and X. Wang, Modeling the Fuzzy Cold Storage Problem and Its Solution by a Discrete Firefly Algorithm, *J. Intell. Fuzzy Syst.* **31**(4) (2016) 2431–2440.
18. H.J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 3rd edn. (Kluwer Academic Publishers, Norwell, 1996).
19. M.S. Yang, W.L. Hung and S.J.C. Chien, On a Similarity Measure between LR-Type Fuzzy Numbers and Its Application to Database Acquisition, *Int. J. Intell. Syst.* **20**(10) (2010) 1001–1016.
20. E. Shekarian, E.U. Olugu, S.H. Abdul-Rashid and N. Kazemi, An economic order quantity model considering different holding costs for imperfect quality items subject to fuzziness and learning, *J. Intell. Fuzzy Syst.* **30**(5) (2016) 2985–2997.
21. Y. Deng, Y. Chen, Y. Zhang and S. Mahadevan, Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, *Appl. Soft Comput.* **12** (2012) 1231–1237.
22. S.H. Chen, S.T. Wang and S.M. Chang, Some Properties of Graded Mean Integration Representation of L-R Type Fuzzy Numbers, *Tamsui Oxford J. Math. Sci.* **22**(2) (2006) 185-208.
23. C.H. Hsieh, Optimization of fuzzy production inventory models. *Inform. Sciences*. **146**(1-4) (2002) 29-40.
24. H. Chen and L. Fang, Research on shelf life-constrained production scheduling for perishable food, *J. Guangxi U. : Nat. Sci. Ed.* **38**(3) (2013) 729-737.
25. Z-shaped membership function. https://cn.mathworks.com/help/fuzzy/zmf.html.
26. M. Wei, X. Chen, B. Sun and Y.Y. Zhu, Model and algorithm for resolving regional bus scheduling problems with fuzzy travel times, *J. Intell. Fuzzy Syst.* **29**(6) (2015) 2689-2696.
27. A. Baykasoğlu and F.B. Ozsoydan, An improved firefly algorithm for solving dynamic multidimensional knapsack problems, *Expert Syst. Appl.* **41**(8) (2014) 3712-3725.
28. M.K. Marichelvam, T. Prabaharan and X.S. Yang, A Discrete Firefly Algorithm for the Multi-Objective Hybrid Flowshop Scheduling Problems, *IEEE T. Evolut. Comput.* **18**(2) (2014) 201-305.
29. B.M. Baker and M.A. Ayechew, A genetic algorithm for the vehicle routing problem, *Comput. Oper. Res.* **30**(5) (2003) 787–800.
30. X.S. Yang, Multiobjective firefly algorithm for continuous optimization, *Eng. Comput-Germany*. **29**(2) (2013) 175-184.
31. S.L. Tilahun and H.C. Ong, Modified Firefly Algorithm, *J. Appl. Math.* **2012** (2012) 1-12.
32. A.H. Gandomi, X.S. Yang and A.H. Alavi, Mixed variable structural optimization using Firefly Algorithm, *Comput. Struct.* **89**(23-24) (2016) 2325–2336.
33. J.B. Atkinson, A Greedy Look-ahead Heuristic for Combinatorial Optimization: An Application to Vehicle Scheduling with Time Windows, *J. Opl. Res. Soc.* **45**(6) (1994) 673-684.