Systems and Soft Computing 7 (2025) 200424

Contents lists available at ScienceDirect

Systems and Soft Computing

journal homepage: www.journals.elsevier.com/soft-computing-letters

Firefly algorithm driven intelligent system for multi-objective multimodal transportation optimization

Tarun Kumar [a], Kapil Kumar [a,b], Kailash Dhanuk [a,c], Anirudh Kumar Bhargava [d,e], M.K. Sharma [a,*]

[a] *Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India*
[b] *Department of Mathematics, Atma Ram Sanatan Dharma College (University of Delhi), New Delhi 110021, India*
[c] *Department of Mathematics, Dyal Singh College (University of Delhi), New Delhi 110003, India*
[d] *Department of Mathematics, MMH College, Ghaziabad, India*
[e] *Ch. Charan Singh University, Meerut, Uttar Pradesh, India*

ARTICLE INFO

Keywords:
Optimization
Firefly algorithm (FA)
Multimodal transportation system (MTS)
Intelligent system
MATLAB

ABSTRACT

This paper presents an intelligent system based on the Firefly Algorithm (FA) designed to optimize the Multi-Objective Multimodal Transportation System (MTS). The inherent complexity of MTS, stemming from the integration of road, rail, air, and sea transport modes, necessitates an efficient optimization strategy to minimize costs and adhere to time constraints. To address this complexity, a sophisticated methodology utilizing the Firefly Algorithm (FA) has been developed. The FA, inspired by the bioluminescent communication of fireflies, offers a robust solution through its ability to effectively explore and exploit the search space. The proposed methodology is fully implemented in MATLAB, making it a versatile and intelligent system. Users only need to input parameters such as the capacity of transport modes and the time taken by each mode, and the system outputs the cost in a solution space. Extensive case studies demonstrate the applicability of the FA in scenarios requiring cost reduction and strict schedule adherence. The performance of the FA is evaluated against other optimization methods, including the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Electron Radar Search Algorithm (ERSA), Grey Wolf Optimization (GWO), Teaching-Learning-Based Optimization (TLBO), and brute-force methods. The results underscore the FA's superiority in terms of cost-efficiency, variability and time management, establishing it as a valuable tool for real-world transportation optimization. By demonstrating how the FA may improve the operational effectiveness of multimodal transportation networks, this study makes a substantial contribution to the fields of supply chain management and logistics. In the future, the proposed methodology can be further enhanced by incorporating additional constraints such as security of products, carbon emission into the MTS problem.

1. Introduction

Logistics' fast-growing subject of Multimodal Transportation System (MTS) optimization is a primary focus. The MTS is complex due to its combination of road, rail, air, and sea transportation modes. These issues usually involve synchronizing several modes economically, quickly, and smoothly. Therefore, decision-makers must choose the optimum route combinations, transshipment costs, and mode-specific constraints. Fig. 1 shows the different multimodal transportation modes. The increasing demands of trade have made Multimodal Transportation Systems (MTS) incredibly important. MTS serves as the foundation, for trade and supply chains. It must be both resilient and adaptable to the ever-changing dynamics of the global market and cross border logistics. MTS is complex due to the need to comply with regulations consider environmental factors and prioritize sustainability in logistics operations. The interconnected nature of MTS means that even a small change can have an impact making system optimization crucial not for logistics but also for economic efficiency and environmental responsibility. Decision makers face a task; finding solutions that are economically viable, efficient, sustainable and compliant with a complex network of regulations. Optimizing MTS is not just necessary from a standpoint; it also plays a role, in gaining competitive advantage in the global marketplace.

---

* Corresponding author.
E-mail address: drmukeshsharma@gmail.com (M.K. Sharma).

https://doi.org/10.1016/j.sasc.2025.200424
Received 16 January 2025; Received in revised form 11 November 2025; Accepted 27 November 2025

Given the tremendous popularity of nature-based optimization methodologies for tackling complicated real-world problems, a multiplicity of studies has already been done and also are currently underway. In recent literature, researchers from around the world have developed a variety of nature-inspired optimization algorithms. These innovative algorithms, as well as their upgrades, modifications, and hybridizations, provide researchers with a wide range of possibilities for effectively addressing domain-specific challenges. These nature-inspired optimization algorithms or meta-heuristics are becoming increasingly popular because the concept and implementation of these algorithms are simple, they do not require gradient information, they have the mechanisms to bypass local optima, and they can be utilized to tackle a varied choice of problems from various disciplines. Despite their diversity, all of these algorithms share a similar feature. The search process of each algorithm initiates with a arbitrary population and progresses through two distinct stages: exploration and exploitation. During the exploration phase, the entire search space is thoroughly investigated to obtain all viable solutions. During the exploitation phase, the potential area of the search space identified during the exploration stage is meticulously examined to uncover a more accurate answer that aligns with the previously obtained probable solutions.

Algorithms in optimization are typically categorized into four groups based on the underlying metaphors used for their development [1,2]. These groups include evolutionary algorithms, swarm intelligence-based algorithms, physical science-inspired algorithms, and human behavior-based algorithms. Examples from each category are as follows: evolutionary algorithms, such as GA [3] and BSA [4]; swarm intelligence-based algorithms, including PSO [5], FA [6] and GWO [33]; physical science-inspired algorithms, like the GSA [7], CSSA [8] and ERSA [50]; and human behavior-based algorithms, such as the TLBO [9] and BSO [10]. These algorithms, along with their various adaptations, have been widely applied to tackle complex real-world problems due to their effectiveness. Applications of these methods can be found in numerous studies [11–20].

FA introduced by Xin-She Yang in 2008, is a swarm intelligence-based optimization technique inspired by the flashing patterns and behaviours observed in tropical fireflies. The FA is simple, flexible, and easy to implement algorithm and for this reason FA is popular among the researchers to solve various real-world problems. Among various metaheuristic algorithms, the Firefly Algorithm (FA) is particularly well-suited for solving multimodal, nonlinear, and combinatorial optimization problems like the one addressed in this study. Its ability to balance exploration and exploitation, coupled with its simplicity and low parameter dependency, makes it an effective and adaptable approach for transportation optimization. This work utilizes the FA to solve the complicated MTS optimization problem. The applicability of FA has been shown with extensive case studies. These case studies include scenarios including lowering transportation expenses within strict schedule limitations employing road, rail, air, and marine transport. The complex nature of MTS is given below:

- MTS is vital to modern logistics, which requires efficient coordination of many transportation options.
- Combining modes cost-effectively and quickly is the biggest challenge, where optimization of route choices, reducing transshipment costs, and overcoming mode-specific restrictions are required.

Table 1 demonstrates how the current study contributes to filling the research gaps observed in existing literature within the relevant field.

The key contributions of the present study are given below:

- The study uses the Firefly Algorithm to navigate MTS complexity. Unlike typical optimization methods, this algorithm uses firefly

Table 1
Evaluation of the proposed work in relation to existing research.

Authors	Multi-modal Transportation	Firefly Algorithm
Osaba et al. [36]	No	Yes
Osaba et al. [37]	No	Yes
Goel & Maini [38]	No	Yes
Aggarwal and Kumar [39]	No	Yes
Micale et al. [41]	No	Yes
Khalifehzadeh and Fakhrzad [43]	No	Yes
Matthopoulos and Sofianopoulou [44]	No	Yes
Trachanatzi et al. [46]	No	Yes
Goel & Maini [49]	No	Yes
Altabeeb et al. [51]	No	Yes
Yesodha and Amudha [59]	No	Yes
Utamima and Indramawan [60]	No	Yes
Pratiba et al. [71]	No	Yes
Proposed	Yes	Yes

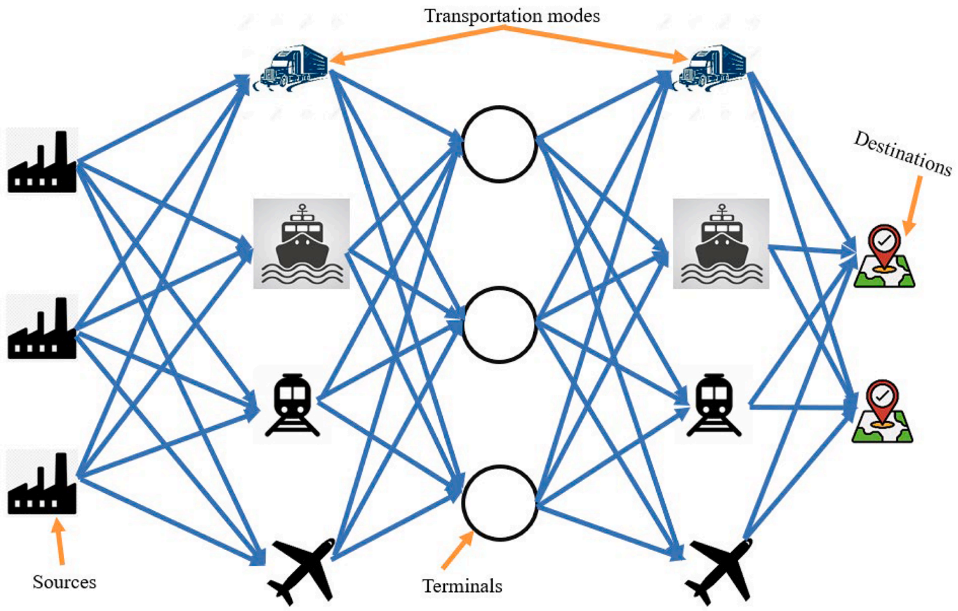

Fig. 1. Multimodal transportation network system.

bioluminescence. Brighter and more appealing solutions encourage other options, which helps convergence toward the perfect solutions.

- The algorithm's applicability is demonstrated for extensive case studies. These case studies include scenarios including lowering transportation expenses within strict schedule limitations employing road, rail, air, and marine transport.
- The FA has been applied to address the challenging MTS problem and the results are contrasted with those of GA, PSO, ERSA, GWO, TLBO, PSO, brute force algorithm [21] and recent developed algorithm GOHBA [73] to validate the proposed work. The FA is found to be competitive in MTS scenarios in cost and time efficiency compared to its competitors.

This work is broken into six larger portions starting with the introduction. The introduction is followed by a literature review for context and background. The second section deals with basic algorithms and their mathematical formulations. The proposed Firefly Method for MTS mathematical modelling is covered in the third segment. The fourth segment covers numerical calculations and algorithm-based programs in various scenarios. The fifth part analyses and interprets these cases, leading to the last section, which draws conclusions and highlights our study's key findings.

1.1. Related work

Pratt and Lomax [22] introduced a new MTS performance assessment paradigm. This study highlighted the importance of cooperation across transportation modes, laying the groundwork for future research. Performance evaluation criteria set the standard for multimodal system effectiveness and integration. Rondinelli and Berry [23] studied the environmental impacts of multimodal logistics in a global market. Their analysis showed the necessity to incorporate environmentally friendly methods into transportation and logistics design, paving the way for a healthier logistics business. Castelli et al. [24] addressed the key MTS scheduling issue, where the authors increased logistics operational research by recognizing and addressing the issues of integrating multiple transportation modes and emphasizing the necessity for smart scheduling. Zografos and Androutsopoulos [25] invented MTS itinerary algorithms, where they showed that computer methods can optimize complex transportation networks, furthering intelligent transport. Yang [26] investigated nature-inspired metaheuristic algorithms and also showed the applicability of these strategies to optimize transportation, shedding light on logistical issues. Kengpol et al. [27] studied Greater Mekong MTS decision support system development to improve multimodal transportation. Fister et al. [28] thoroughly reviewed the FA and demonstrated the algorithm's adaptability and efficiency to solve complex issues in various real-world scenarios. Jati and Manurung [29] solved the Traveling Salesman Problem using discrete fireflies. Yang [30] modified FA to solve the multi-objective optimization problems and thereby introduced multi-objective FA. The strategy worked effectively in complex scenarios like multidimensional optimization. Peters and colleagues analyzed the Midwest corridor's multimodal transportation system's high-speed rail potential. This study helps integrate high-speed rail into Mass Transit System (MTS) networks by showing its socio-economic and environmental effects [31]. Rahmani and MirHassani [32] used FA and GA to locate facilities. This hybrid technique showed the synergistic potential of combining optimization strategies to improve problem-solving. Mirjalili et al. [33] proposed the Grey Wolf Optimizer (GWO) as a novel swarm-based metaheuristic for solving engineering design problems. By mimicking the hierarchical leadership and hunting mechanism of grey wolves, the GWO algorithm has demonstrated impressive convergence behaviour in high-dimensional search spaces. Nekouie and Yaghoobi [34] used the FA to develop multimodal optimization methods. They showed that the approach is flexible and effective, making it useful for complex logistics optimization challenges. Container multimodal transport routes and modes were optimized using dynamic programming by Hao and Yue [35]. Their technique to solving these complex difficulties expanded multimodal transport optimization. Osaba et al. [36] solved the tough vehicle routing with time windows problem, where a discrete FA was utilized for handling multivariable, non-linear problems. Their method is unique because new operators improved the algorithm's ability to find optimal or near-optimal solutions. Osaba et al. [37] solved automobile navigation problems using the discrete FA. This transportation and logistics use proved the method's viability. To solve vehicle routing challenges, Goel and Maini [38] created a hybrid ant colony-firefly algorithm. This new approach demonstrated how metaheuristic strategies increase solution quality. Aggarwal and Kumar improved the Firefly Algorithm for time-window vehicle routing. The system worked well in time-sensitive transportation planning with limited resources [39]. Chandravati and colleagues extensively examined firefly algorithms. The FA was used in route planning, vehicle routing, and Traveling Salesman Problems [40]. Micale et al. [41] routed vehicles sustainably using the FA and TOPSIS. Their optimization-sustainability method advanced green logistics. Chen et al. [42] optimized container-based multimodal transportation routes. This study revealed shipping efficiency and sustainable logistics. To optimize uncertain multi-stage supply chain networks, Khalifehzadeh and Fakhrzad [43] improved the FA and applied it to solve supply chain management uncertainty and complexity handling. Matthopoulos and Sofianopoulou [44] solved fleet vehicle routing problems with the help of FA. They found that the method is adaptable and efficient for logistical issues with predetermined resources. Maity et al. [45] used AI to study MTS. This work integrates computational methods with transportation and logistics, expanding research and applications. Trachanatzi et al. [46] used FA to route environmental prize-collecting vehicles. Sun et al. [47] addressed MTS sea level protection. The study found climate change required resilient and flexible transportation systems. Elhoseny [48] created a cutting-edge vehicular communication network improvement solution by Levy distribution-based FA, where Levy distribution was used to increase FA's adaptability in dynamic vehicle contexts. Goel and Maini [49] designed an evolutionary ant colony algorithm for vehicle routing using firefly-based transition techniques. Rahmanzadeh and Pishvaee [50] introduced the Electron Radar Search Algorithm, a physics-inspired metaheuristic aimed at improving exploration and exploitation in complex optimization tasks. Altabeeb et al. [51] used a cooperative FA to route capacitated vehicles. They demonstrated the benefits of collaborative logistics problem-solving and provided a new routing optimization perspective. Liu et al. [52] predicted MTS urban network capacity using second-best limitations. This study integrated traffic theory and multimodal logistics to get insights into urban transportation network administration. Pavlenko et al. [53] showed that mathematical modeling can aid MTS rational logistical route selection. They demonstrated the need for rigorous analytical methods in improving transportation routes. Archetti et al. [54] provided a multimodal freight transportation optimization assessment. This survey is useful for understanding current approaches and trends. Wu et al. [55] developed an MTS recommendation system through the incorporation of individual preferences and requirements into the design of transportation systems. This study marks a major step forward in the process of developing user-centric transportation solutions. An investigation into multi-objective optimization for the vehicle routing issue within the context of low-carbon intelligent transportation is presented by Yin [56]. A complete review of computational intelligence in intelligent transportation systems is presented by Hina et al. [57]. In addition, Chen et al. [58] provided a comprehensive assessment of evolutionary computation for intelligent mobility in smart cities. Yesodha and Amudha [59] applied a bio-inspired Firefly Algorithm to address the multi-depot vehicle routing problem with time windows. Utamima and Indramawan [60] introduced a Hybrid Firefly Algorithm to optimize garbage collection routes, blending firefly-based exploration with problem-specific heuristics. Vasheghani and Abtahi [61] concentrated their efforts on strategic planning for multimodal transportation in port scenarios. Through their efforts, logistics management is brought into alignment with port operations. Akuh et al. [62] presented a system for determining how well new towns and cities strike a balance between land use and MTS measures. This method makes use of an integrated modeling framework, which makes a contribution to the academic area of urban planning and transportation. Feng et al. [63] examined the benefits and difficulties that are linked with the use of MTS in connection with cargo containerization technology. Their predictions for future trends and technologies are crucial to understanding multimodal transportation's shifting environment. Lv and Shang [64] comprehensively examined the effects of smart transport systems on energy conservation and pollution reduction. Wang et al. [65] studied deep learning for transshipment route planning. They used machine learning to bridge the gap between multiple transportation systems (MTS) and computational approaches to improve transportation efficiency. Zhang and Zhu [66] applied PSO to solve multi-objective optimization issues in 2023. Brar et al. [67] utilized the TLBO algorithm to enhance the efficiency and effectiveness of Multimodal Transportation Models. Wang et al. [68] proposed a multi-objective FA, which was developed with the purpose of addressing the complicated problem of rescheduling hybrid flowshops efficiently in terms of energy consumption. Hashemi et al. [69] developed a novel methodology using the Cat swarm optimization algorithm to tackle service activation in fog computing. In the same period, Kartli et al. [70] optimized fuzzy TP using a heuristic algorithm. Pratiba et al. [71] combined Q-learning with the Firefly Algorithm, creating a hybrid optimization framework for handling transportation problems under dynamic conditions.

2. Basic definitions, algorithms and mathematical formulations

2.1. Mathematical formulation of MTS to optimize the cost while adhering to time constraints

Let x_{ijk} be a binary variable that equals 1 of mode k is used to transport goods from city i to city j, and 0 otherwise.

The objective is to minimize the total cost of transportation (Eq. 1):

$$Minimize\ Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m} c_{ijk}x_{ijk} \tag{1}$$

here; c_{ijk} is the cost of transporting goods using mode k from city i to city j, n is the number of cities and m are the available transportation modes.

Constraints

- The total transport period must not exceed the time limit T (Eq. 2):

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m} t_{ijk}x_{ijk} \leq T \tag{2}$$

here; t_{ijk} is the time taken of transport goods using mode k from city i to city j.

- Mode usage constraint

To ensure that at least two modes are used, we can add a constraint for each pair of cities (Eq. 3):

$$\sum_{k=1}^{n} x_{ijk} \geq 2\ \forall\ i,j \tag{3}$$

- Binary constraint

Each x_{ijk} can only take the value 0 or 1

$x_{ijk} \in \{0,1\}\ \forall\ i,j,k$

- Flow conservation constraint

These constraints ensure that the number of goods leaving a city equals the number of goods entering the city (excluding the origin and destinations) (Eq. 4):

$$\sum_{j=1}^{n} x_{jik} - \sum_{j=1}^{n} x_{ijk} = 0\ \forall\ i \neq origin,\ destination \tag{4}$$

Here, x_{jik} represents commodities transported from city j to city i (inflow), and x_{ijk} represents commodities transported from city i to city j (outflow). This equation balances the inflow and outflow for intermediate nodes (cities), ensuring conservation of flow.

2.2. Extended MTS model with environmental and safety considerations

We extend the baseline cost–time formulation by incorporating carbon emissions, energy consumption, and safety. The model remains a network flow with binary mode selection and adds quantity variables so that environmental terms scale with shipped amount.

Sets and Indices

- N: set of nodes (origins, destinations, transshipment nodes).
- $A \subseteq N \times N$: set of feasible directed arcs.
- M: set of transportation modes.
- Indices: $i,j \in A$, $m \in M$.

Parameters

c_{ij}^{m}: cost per unit shipped on (i,j) by mode m.
t_{ij}^{m}: travel time on (i,j) by mode m.
e_{ij}^{m}: carbon emission factor (e.g. kg CO_2 per unit) on (i,j), m
p_{ij}^{m}: energy use (e.g. kWh or liters per unit) on (i,j), m
s_{ij}^{m}: safety risk index (dimensionless; larger = riskier) on (i,j), m
u_{ij}^{m}: capacity on (i,j), m (maximum shippable units)
b_i: net supply at node i (Positive for sources, negative for sinks, zero otherwise)

- Optional system limits: T_{max}, E_{max}, P_{max}, S_{max}.
- Weights for single-objective aggregation: w_c, $w_t, w_e, w_p, w_s \geq 0$ with $\sum w_{\cdot} = 1$

Decision Variables

- $x_{ij}^{m} \in \{0,1\}$: 1 if arc (i,j) is used by mode m, else 0.
- $q_{ij}^{m} \geq 0$: quantity shipped on (i,j) by mode m.
- $z_m \in \{0,1\}$: 1 if mode m is used anywhere in the network.

Objective

$$\begin{aligned} min\ Z = \ & w_c \sum_{(i,j)\in A}\sum_{m\in M} c_{ij}^{m}q_{ij}^{m} + w_t \sum_{(i,j)\in A}\sum_{m\in M} t_{ij}^{m}x_{ij}^{m} + w_e \sum_{(i,j)\in A}\sum_{m\in M} e_{ij}^{m}q_{ij}^{m} \\ & + w_p \sum_{(i,j)\in A}\sum_{m\in M} p_{ij}^{m}q_{ij}^{m} + w_s \sum_{(i,j)\in A}\sum_{m\in M} s_{ij}^{m}x_{ij}^{m} \end{aligned} \tag{5}$$

Constraints

- $\sum_{j:(i,j)\in A}\sum_{m\in M} q_{ij}^{m} - \sum_{j:(j,i)\in A}\sum_{m\in M} q_{ij}^{m} = b_i,\ \forall\ i \in N$ (Flow conservation) (6)
- $0 \leq q_{ij}^{m} \leq u_{ij}^{m}x_{ij}^{m} \forall\ (i,j) \in A, m \in M$ (Capacity and linking) (7)
- $\sum_{m\in M} x_{ij}^{m} \leq 1,\ \forall\ (i,j) \in A$ (At most one mode per arc (if required)) (8)

- $x_{ij}^m \le z_m \forall\ (i,j) \in A,\ m \in M,\ \sum_{m\in M} z_m \ge 2$ (Network wide "use at least two modes") (9)
- $\sum_{(i,j)\in A}\sum_{m\in M} t_{ij}^m x_{ij}^m \le T_{max}$ (Aggregate time limit) (10)
- $\sum_{(i,j)\in A}\sum_{m\in M} e_{ij}^m q_{ij}^m \le E_{max}$ (Aggregate emission limit) (11)
- $\sum_{(i,j)\in A}\sum_{m\in M} p_{ij}^m q_{ij}^m \le P_{max}$ (Aggregate energy limit) (12)
- $\sum_{(i,j)\in A}\sum_{m\in M} s_{ij}^m x_{ij}^m \le S_{max}$ (Aggregate safety limit) (13)
- $x_{ij}^m \in \{0,1\},\ z_m \in \{0,1\},\ q_{ij}^m \ge 0$ (domains)

2.3. Mathematical formulation of firefly algorithm

The goal is to optimize an objective function $f(x)$, where x is a candidate solution in the search space. i.e. $min/max\ f(x),\ x \in R^n$, here $x = \{x_1,\ x_2,\ x_3, \dots\dots\dots, x_n\}$ a solution vector in the $n-$ dimensional search space.

The brightness I_i of a firefly is directly proportional to the fitness of its solution: $I_i \propto f(x_i)$. For a minimization problem, brightness is inversely proportional to the objective function value: $I_i = \frac{1}{f(x_i)}$.

The attractiveness of a firefly decreases with the square of the distance (r_{ij}) between two fireflies: $\beta = \beta_0 e^{-\gamma r_{ij}^2}$, here β_0 is initial attractiveness, γ is light absorption coefficient and r_{ij} is Euclidean distance between firefly i and firefly j (Eq. 14):

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{i,k} - x_{j,k}\right)^2} \tag{14}$$

A firefly i is attracted to another brighter firefly j, and its movement is governed by Eq. 15:

$$x_i = x_i + \beta.\left(x_j - x_i\right) + \alpha.(rand - 0.5) \tag{15}$$

Here, x_i is current position of firefly i. x_j is position of the brighter firefly j. β is attractiveness between fireflies i and j. α is randomization parameter controlling the step size. *rand* is random number uniformly distributed in $[0,1]$.

The term $\alpha.(rand - 0.5)$ adds randomness to the movement to ensure exploration of the search space.

The flowchart of FA is shown in Fig. 2 (Algorithm 1).

3. Proposed mathematical formulation of the model

3.1. Mathematical formulation of the model in context of firefly algorithm

Let x_{ij} denotes the transportation mode used for the i^{th} firefly on the j^{th} leg of the journey, where $i = 1, 2, \dots\dots\dots, n_{fireflies}$ and $j = 1, 2, \dots\dots\dots, n_{legs}$.

Minimize the total cost of transportation across all fireflies and all legs of the journey (Eq. 16):

$$Minimize\ Z = \sum_{i=1}^{n_{fireflies}} \sum_{j=1}^{n_{legs}} c_{x_{ij}} \tag{16}$$

here $c_{x_{ij}}$ is the total cost of the transport mode used by the i^{th} firefly on the j^{th} leg of the journey.

Constraints:

- Time constraint

The total time of TP for each firefly must not exceed the time limit T (Eq. 17):

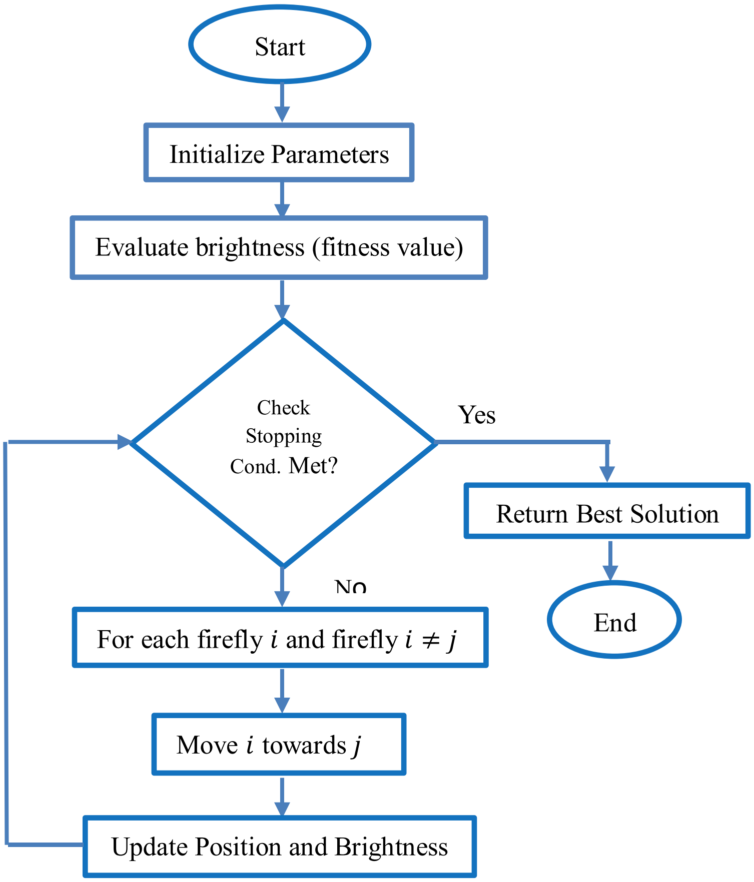

Fig. 2. Flowchart of the firefly algorithm.

Algorithm 1

Firefly algorithm (FA).

```
1. Initialize the required parameters such as $\alpha$ (randomness parameter), $\beta_0$ (initial attractiveness), $\gamma$ (light absorption coefficient), population size $N$, and the maximum number of iterations $T$. Randomly initialize the positions of all fireflies in the search space.
2. Evaluate brightness (fitness value) of each firefly using the objective function $f(x)$.
3. do
  • for each firefly $i$:
    • for each firefly $j \neq i$:
      ○ If $f(x_j) < f(x_i)$:
        ■ Move firefly $i$ towards firefly $j$ using the equation:
            $x_i = x_i + \beta.(x_j - x_i) + \alpha.(rand - 0.5)$
          Ensure $x_i$ remains within the search space:
            $x_i = \max(x_{min}, \min(x_{max}, x_i))$
  • Update brightness of each firefly based on its position.
4. while the stopping criterion (e.g., maximum iterations) is not satisfied.
5. return the position of the brightest firefly (best solution)
```

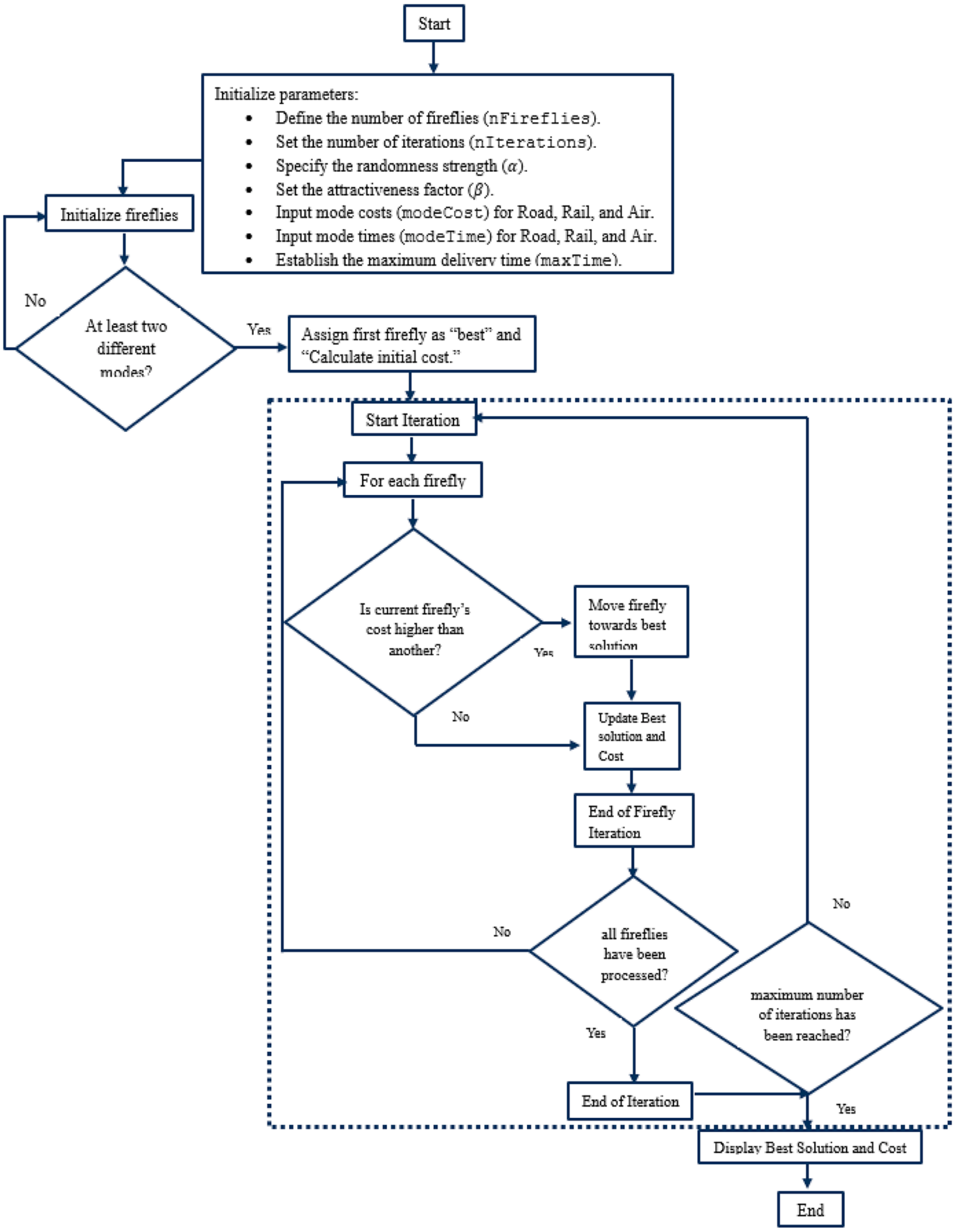

Fig. 3. Flow process of the model.

$$\sum_{j=1}^{n_{legs}} t_{x_{ij}} \leq T\ \forall\ i \tag{17}$$

here $t_{x_{ij}}$ is the time required for the i^{th} firefly to travel the j^{th} leg of the journey using mode x_{ij}.

- Mode diversity constraint:

  Each firefly must use at least two different modes of TP (Eq. 18):

$$\left|unique\left(\left\{x_{ij} : i = 1, 2, \ldots\ldots\ldots n_{legs}\right\}\right|\geq 2\ \forall i \tag{18}$$

- Penalty for exceeding time:

  A large penalty is added if the time constraint is violated.
  If $\sum_{j=1}^{n_{legs}} t_{x_{ij}} > T$ then add penalty to Z
  Firefly algorithm components:

- Brightness (Objective function related) (Eq. 19)

$$I_i = -Cost(x_i) \tag{19}$$

  Where I_i is the brightness of the i^{th} firefly.

- Attractiveness given by Eq. 20:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \tag{20}$$

here r_{ij} is the distance between firefly i and j and β_0 is the attractiveness at $r = 0$.

- Distance between solutions (Eq. 21):

$$r_{ij} = \sqrt{\sum_{j=1}^{n_{legs}} \left(x_{ij} - x_{jj}\right)^2} \tag{21}$$

- Movement of the firefly i towards firefly j (Eq. 22):

$$x_{ij}^{new} = x_{ij} + \beta(r_{ij}).(x_{jj} - x_{ij}) + \alpha(rand(.) - 0.5) \tag{22}$$

here α is the randomization parameter and $rand()$ is a random number between 0 and 1.

The flowchart of proposed FA for MTS is shown in Fig. 3.

3.2. *Extended mathematical formulation of the model in context of firefly algorithm*

Let $x_{i,d}$ denote the transportation mode used by the i^{th} firefly on the d^{th} leg of the journey, where $i = 1, 2, \ldots.., n$ (fireflies) and $d = 1, 2, \ldots\ldots, D$ (legs). For each mode $m \in \{Road, Rail, Sea, Air\}$, let c_m, t_m, e_m, p_m and s_m be the cost, time, emissions, energy and safety. Define for firefly i

$$C_i = \sum_{d=1}^{D} c_{x_{i,d}},\ T_i = \sum_{d=1}^{D} t_{x_{i,d}},\ E_i = \sum_{d=1}^{D} e_{x_{i,d}},\ P_i = \sum_{d=1}^{D} p_{x_{i,d}},\ S_i = \sum_{d=1}^{D} s_{x_{i,d}}$$

Minimize the weighted total across all fireflies and legs (Eq. 23)

$$\min Z = \sum_{i=1}^{n} \left(w_c C_i + w_t T_i + w_e E_i + w_p P_i + w_s S_i\right) \tag{23}$$

where $w_c + w_t + w_e + w_p + w_s = 1$

Constraints:

- Time limit

$$T_i \leq T_{max},\ i = 1, \ldots, n$$

- Emissions limit

$$E_i \leq E_{max},\ i = 1, \ldots, n$$

- Energy limit

$$P_i \leq P_{max},\ i = 1, \ldots, n$$

- Safety limit

$$S_i \leq S_{max},\ i = 1, \ldots, n$$

- Mode diversity constraint: each firefly must use at least two different modes over D legs, $|\{x_{i,1}, \ldots\ldots, x_{i,D}\}| \geq 2, i = 1, \ldots\ldots., n$

$$T_i \leq T_{max},\ i = 1, \ldots, n$$

- Penalty note:

If any constraint is violated for firefly i, add a large penalty to its contribution inside (Eq. 23): $Z \leftarrow Z + \rho\ (\rho \gg 0)$

Firefly algorithm components:

- Brightness (Objective- related): for a minimization task, brightness of firefly i is inversely proportional to its weighted value (Eq. 24):

$$B_i = \frac{1}{1 + \left(w_c C_i + w_t T_i + w_e E_i + w_p P_i + w_s S_i\right)} \tag{24}$$

- Attractiveness (decreases with squared distance) (Eq. 25)

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2},\ \beta_0 > 0, \gamma > 0 \tag{25}$$

- Distance between solutions on the continuous FA space (Eq. 26)

$$r_{ij} {=} \| y_i - y_j \text{AptCommand2016;}_2 = \sqrt{\sum_{d=1}^{D} \left(y_{i,d} - y_{j,d}\right)^2} \tag{26}$$

- Movement of firefly i towards brighter firefly j (Eq. 27):

$$y_i \leftarrow y_i + \beta(r_{ij})\left(y_j - y_i\right) + \alpha\left(u - \frac{1}{2}\right),\ u \sim U[0,1]^D \tag{27}$$

3.3. *Proposed algorithm*

4. Numerical computations

Problem 1. To evaluate the performance of the FA, we benchmark it on ten multimodal test functions proposed by Qu et al. [72], namely ESF1–ESF5, ESF6, ESF8, and CF1–CF3. These functions represent diverse landscapes, including trap-type, unequal and decreasing minima, pairwise interactions, logarithmic oscillations and complex

Algorithm 2

Firefly algorithm for transport mode optimization.

```
1. Initialize required parameters such as $n$ (total number of fireflies), $T$ (number of iterations), $\alpha$ (coefficient of randomness), $\beta$ (set of attractiveness coefficients), $C$ (cost vector for transport modes: Road, Rail, Sea, Air), $M$ (time vector for transport modes: Road, Rail, Sea, Air), $M_{max}$(maximum allowable delivery time), $p_m$ (Penalty for modes constraint), $p_t$ (Penalty for time exceeding).
2. Initialize transport modes $x_i$ for each firefly $i$, ensuring at least two distinct modes for three and four transport segments.
3. do
  • For each attractiveness coefficient $\beta_k$ in $\beta$:
      a. Set $BestCost_k$ to a high value (e.g., infinity).
  2. Update the population $P(t+1)$ by selecting the finest results from the current population and offspring. For each firefly $i$:
      ■ For each firefly $j$:
        If $Cost(x_j) < Cost(x_i)$:
        Move $x_i$ towards $x_j$ using:
          $x_i = round(x_i + \beta_k \cdot (x_j - x_i) + \alpha \cdot (rand(1, no.\ of\ modes) - 0.5))$
        Ensure $x_i$ remains valid:
              $x_i = max(1, min(no.\ of\ modes, x_i))$
        ■ Update $BestCost_k$ if $Cost(x_i) < BestCost_k$
4. Display Results:
  • For each $\beta_k$, print $\beta_k$ and $BestCost_k$
  • Plot $\beta$ versus $BestCost$
5. Cost Calculation Function:
• Input: Solution vector $x$ indicating transport modes.
• Calculate total cost:
      $TotalCost = \sum_{i=1}^{no.\ of\ modes} C_{x_i}$
• Calculate total time:
      $TotalTime = \sum_{i=1}^{no.\ of\ modes} M_{x_i}$
• Apply penalties:
    ○ If not all transport modes are used:
      $TotalCost = TotalCost + p_m$ (two modes penality)
    ○ If $TotalTime > M_{max}$:
        $TotalCost = TotalCost + p_t$ (time exceeding penality)
• Return TotalCost.
  Return the $BestCost_k$ for each $\beta_k$.
```

Algorithm 3

Firefly algorithm for extended transportation model.

```
  • Input:
      • Number of fireflies $n$, maximum iterations $I$.
      • Firefly parameters: attractiveness $\beta_0$, absorption $\gamma$, randomness $\alpha$.
      • Model data: cost $c_{d,m}$, time $t_{d,m}$, emissions $e_{d,m}$, energy $p_{d,m}$, safety $s_{d,m}$; limits ($T_{max}$, $E_{max}$, $P_{max}$, $S_{max}$); weights $(w_c, w_t, w_e, w_p, w_s)$.
  • Output:
      • Best feasible solution $x^*$(mode assignment per leg) with objective value $Z^*$.
Begin
  1. Initialize population:
      • For Generate positions $y_i \in [1, |M|]^D$ uniformly at random.
      • Plot Discretize each $y_i$ to integer modes $x_i$.
  2. Evaluate brightness:
    For each firefly $i = 1, \ldots, n$
    • Compute totals:
      $C_i = \sum_{d=1}^{D} c_{x_{i,d}}$, $T_i = \sum_{d=1}^{D} t_{x_{i,d}}$, $E_i = \sum_{d=1}^{D} e_{x_{i,d}}$, $P_i = \sum_{d=1}^{D} p_{x_{i,d}}$, $S_i = \sum_{d=1}^{D} s_{x_{i,d}}$
    • Apply penalties if $T_i > T_{max}$, $E_i > E_{max}$, $P_i > P_{max}$, $S_i > S_{max}$or fewer than two modes used.
    • Compute fitness
        $Z_i = w_c C_i + w_t T_i + w_e E_i + w_p P_i + w_s S_i + \prod x_i$
    • Brightness: $B_i = \frac{1}{(1 + Z_i)}$.
  3. Iterative update:
    For $t = 1$ to $I$:
    • C For each pair of fireflies $(i, j)$ with $B_j > B_i$:
      • Compute Euclidean distance: $r_{ij} = \| y_i - y_j$AptCommand2016;$_2$.
      • Attractiveness: $\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2}$.
      • Move firefly $i$ towards $j$:
        $y_i \leftarrow y_i + \beta(r_{ij})\left(y_j - y_i\right) + \alpha(u - 0.5)$, $u \sim U[0, 1]^D$
      • Apply boundary control: $y_i \in [1, |M|]^D$.
      • Discretize to $x_i$, evaluate $Z_i$ and brightness $B_i$.
    • Update best so far solution $x^*$.
  4. Return $x^*$ with objective value $Z^*$.
  End
```

composition functions. The objective is to assess FA's convergence ability and robustness against local optima across challenging optimization scenarios.

In our study, we rigorously examine and build upon Problems 1 and 2 as delineated in the seminal work of Brar et al. [67].

Problem 2. Consider a scenario where a shipping company needs to transport a batch of goods from a warehouse to a delivery centre. They can choose from three transportation modes: road, rail, or air. Each option involves specific costs and transit times, detailed as follows: **Road Transport**: This mode costs $500 and takes 8 h to complete the delivery. **Rail Transport**: Opting for rail incurs a cost of $700 and requires 5 h for the shipment to reach its destination. **Air Transport**: The fastest option is air transport, which takes only 3 h but comes at a higher cost of $1000. The aim is to transport goods in less than 18 h, and the cost must be minimized.

Extended Problem 2. In order to integrate environmental and safety considerations, the Problem 1 is reformulated. A shipment must traverse **three consecutive legs**, and each leg can be operated by **Road, Rail**, or **Air**. The parameters of each mode include cost, time, carbon emissions, energy consumption and safety risk, as presented in Table 3.

The system-wide restrictions are: maximum delivery time $T_{max} = 18$ h, carbon emission limit $E_{max} = 500$ kg, energy limit $P_{max} = 300$, and safety index limit $S_{max} = 0.12$. The optimization problem is to minimize the weighted objective function (Eq. 5), subject to constraints (Eqs. (6)–(13)), ensuring that at least two transportation modes are employed across the three legs.

Solution: We have used MATLAB to handle this problem. Below is the problem's output:

4.1. Output (Problem 2)

Best Solution:
Road, Rail, Rail
Best Cost: 1900

Problem 3. Consider the transportation of a shipment of products from a warehouse to a delivery center, where four distinct modes of transportation are available: road, rail, air, and sea. Each mode comes with its unique costs and delivery durations: **Road Transport**: This mode costs $700 and requires 10 h for delivery. **Shipping**: Using this option costs $1905 and takes 12 h to complete the shipment. **Rail Transport**: Choosing rail involves a cost of $850 and a transit time of 7 h. **Air Transport**: The quickest mode is air transport, which takes only 2 h but has the highest cost at $2550. The aim is to supply the products in less than 36 h and keep costs down.

Extended Problem 3. The second problem is similarly extended. Here, a shipment consists of **four consecutive legs**, and each leg may be assigned to **Road, Sea, Rail**, or **Air**. Table 4 reports the multimodal parameters including cost, time, emissions, energy, and safety.

The system-wide restrictions are: maximum delivery time $T_{max} = 36$ h, carbon emission limit E_{max}=900 kg, energy limit $P_{max} = 500$, and safety index limit $S_{max} = 0.18$. The optimization problem is to minimize the weighted objective function (Eq. 5), subject to constraints (Eqs. (6)–(13)), ensuring that at least two transportation modes are employed across the three legs.

Solution: We have used MATLAB to handle this problem. Below is the problem's output:

4.2. Output (Problem 3)

Best Solution:
Rail, Road, Road, Rail
Best Cost: 3100

4.3. Explanatory notes for Figs. 4–9 (Problem 2)

Fig. 4: Best costs when $\beta = 0.0001$.

→ At very low β values, the algorithm shows unstable convergence with sharp fluctuations. This reflects high sensitivity, where exploration dominates and prevents the algorithm from quickly refining solutions.

Fig. 5: Best costs when $\beta = 0.02$.

→ Slightly higher β values improve performance but still show variability. The algorithm begins to transition toward more stable behaviour, though exploration remains strong.

Fig. 6: Best costs when $\beta = 0.025$.

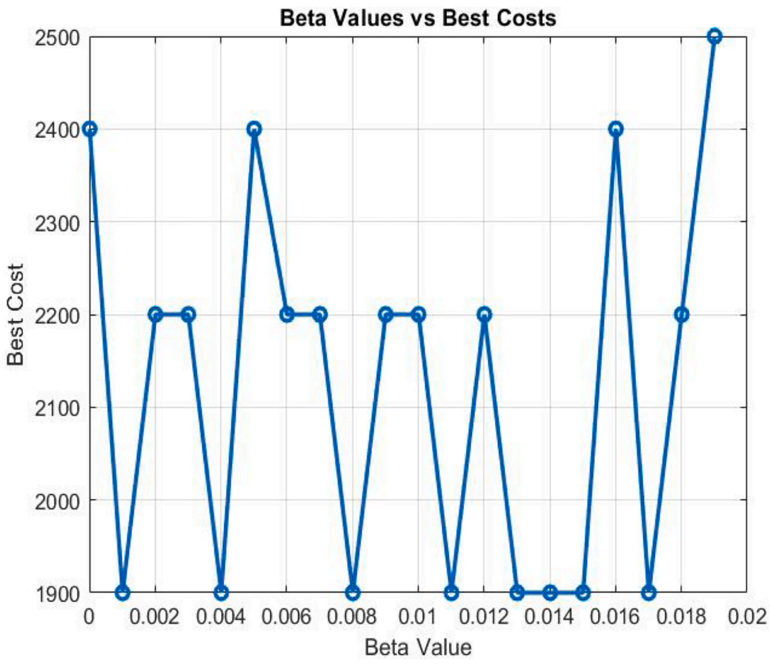

Fig. 4. Best costs when $0.00001 \leq \beta \leq 0.02$.

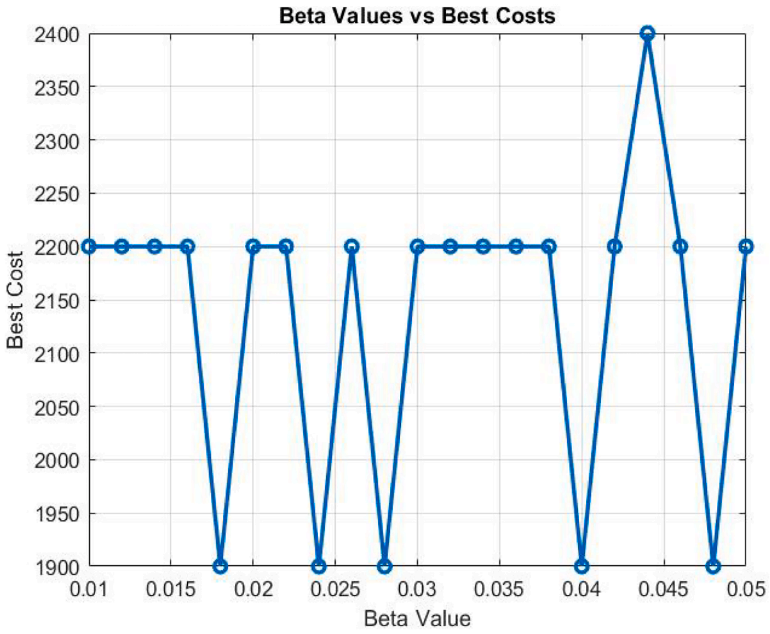

Fig. 5. Best costs when $0.01 \leq \beta \leq 0.05$.

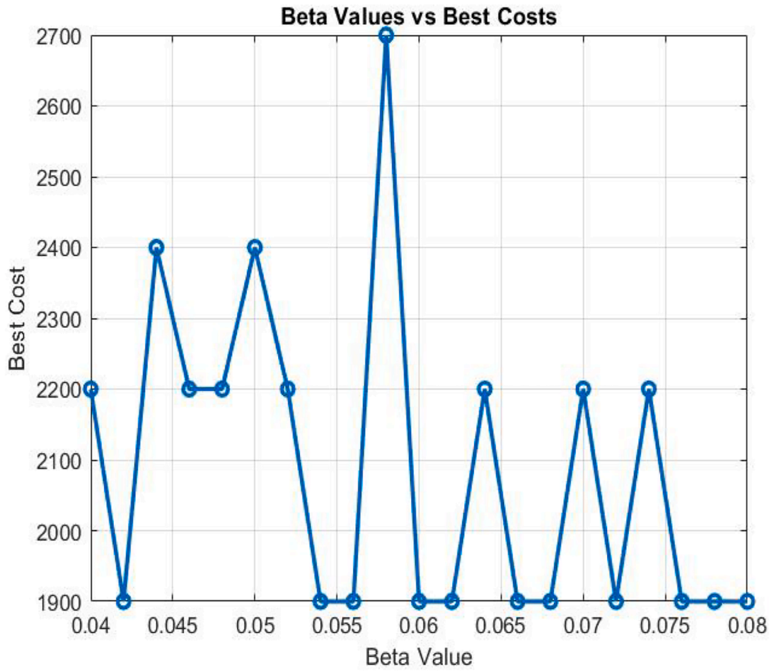

Fig. 6. Best costs when $0.04 \leq \beta \leq 0.08$.

→ Convergence stabilizes compared to lower β. The algorithm achieves more consistent cost reductions, indicating a better balance between exploration and exploitation.

Fig. 7: Best costs when $\beta = 0.05$.

→ At this intermediate level, the FA converges reliably to low costs, showing fewer spikes. This suggests an effective trade-off between global exploration and local refinement.

Fig. 8: Best costs when $\beta = 0.055$.

→ Near this range, FA exhibits strong exploitation with smoother convergence patterns, consistently approaching optimal solutions. Variability is minimized.

Fig. 9: Best costs when $\beta = 0.1$.

→ At higher β values, convergence is fast and stable. The algorithm predominantly exploits promising solutions while still maintaining enough exploration to avoid local minima.

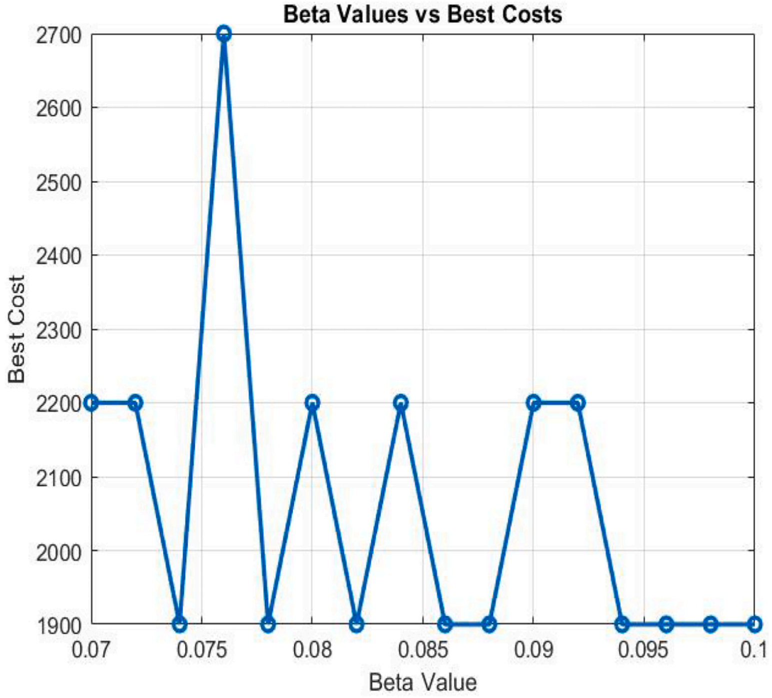

Fig. 7. Best costs when $0.07 \le \beta \le 0.1$.

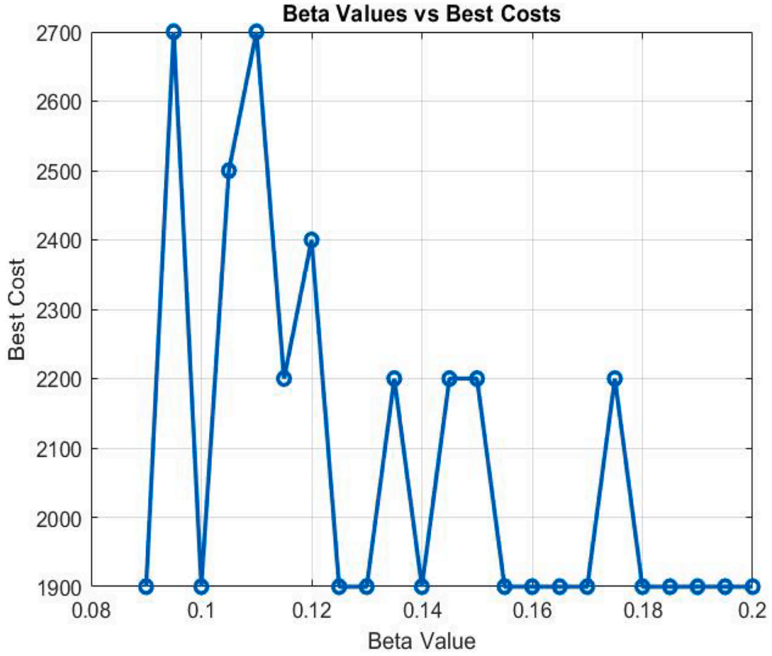

Fig. 8. Best costs when $0.09 \le \beta \le 0.2$.

4.4. Explanatory notes for Figs. 10–15 (Problem 3)

Fig. 10: Best costs when $\beta = 0.0001$.

→ Similar to Problem 1, low β values cause high variability. The search remains largely exploratory, delaying convergence to optimal costs.

Fig. 11: Best costs when $\beta = 0.02$.

→ Slight improvement over the lowest β, but convergence is still unstable, showing the algorithm's cautious exploration tendency.

Fig. 12: Best costs when $\beta = 0.025$.

→ At this stage, FA begins to converge more consistently. Costs drop steadily, suggesting that the algorithm is escaping local optima more effectively.

Fig. 13: Best costs when $\beta = 0.05$.

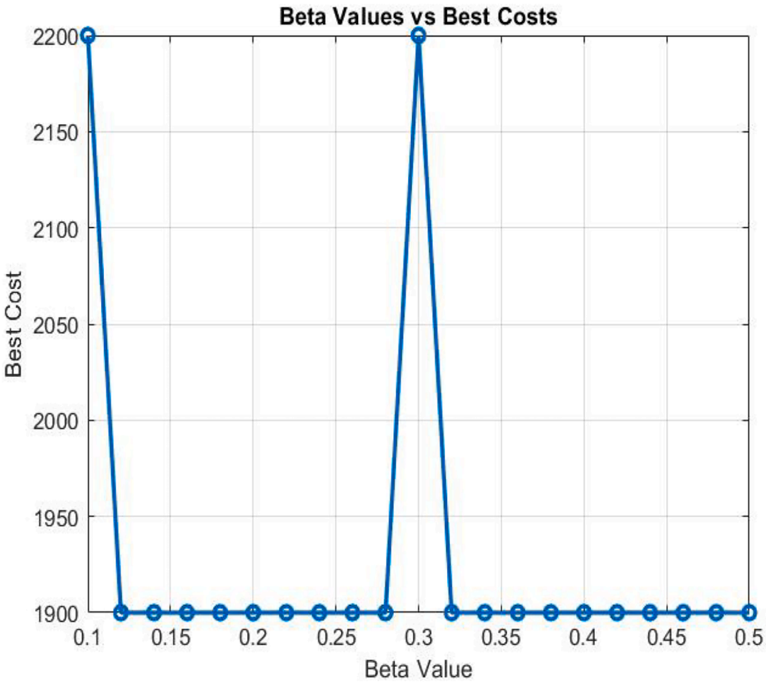

Fig. 9. Best costs when $0.1 \le \beta \le 0.5$.

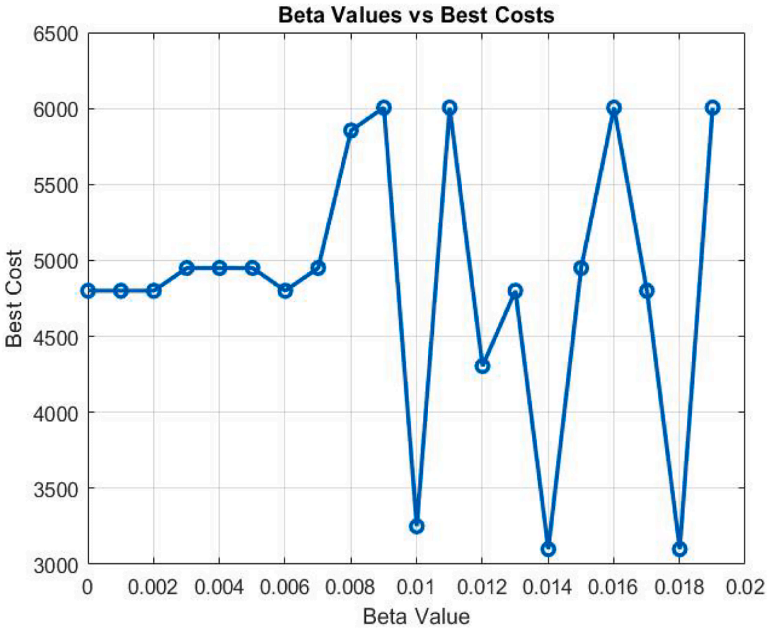

Fig. 10. Best costs when $0.00001 \le \beta \le 0.02$.

→ Optimal balance observed. The algorithm refines solutions while still exploring alternative paths, yielding consistently low costs.

Fig. 14: Best costs when $\beta = 0.15$.

→ Higher β values drive stronger exploitation. FA reaches lower-cost solutions quickly, though minor spikes indicate occasional over-exploration.

Fig. 15: Best costs when $\beta = 0.20$.

→ Very high β values allow extensive exploration but also increase variability. While low costs are still obtained, the stability of convergence is somewhat reduced.

5. Results and discussion

5.1. *Benchmark multimodal test functions (Problem 1)*

To assess the robustness of the Firefly Algorithm (FA), Problem 1 utilized ten multimodal benchmark functions proposed by Qu et al. [72] (Table 2). These cover trap-type, uneven minima, oscillatory and composite landscapes.

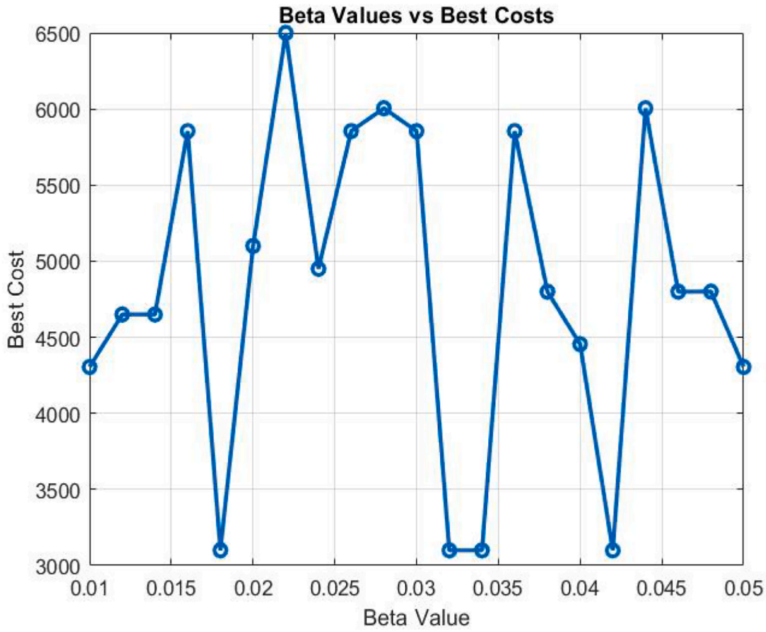

Fig. 11. Best costs when $0.01 \leq \beta \leq 0.05$.

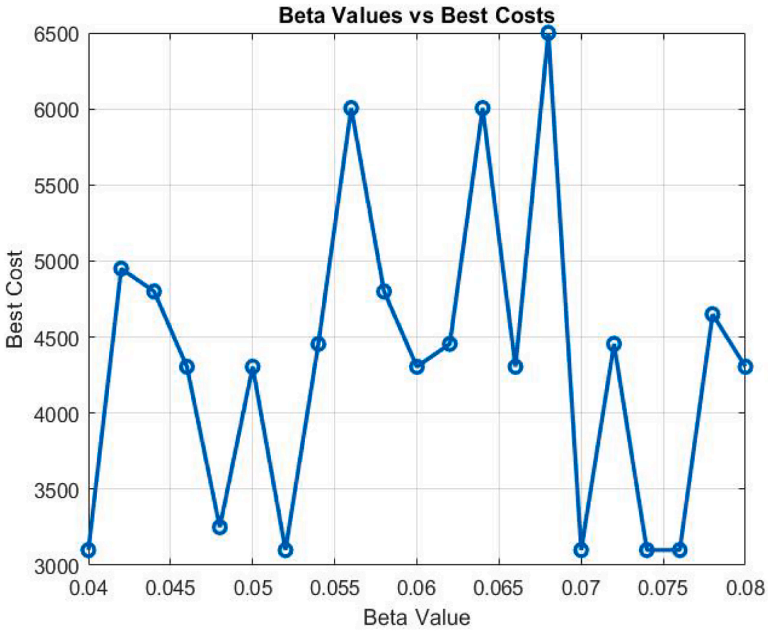

Fig. 12. Best costs when $0.04 \leq \beta \leq 0.08$.

5.2. *MTS (Problem 2)*

5.2.1. *Standard Problem 2*

In the three-mode system (road, rail, air) with an 18 h time constraint, FA (Algorithm 2) produced stable and competitive solutions. Figs. 10–15 reveal that costs gradually decreased as β increased, with optimal stability occurring in the intermediate–high range. The best multimodal plan was Rail–Road–Road–Rail, yielding a total cost of $3100. Figs. 4–9 show the convergence patterns of FA across different attractiveness parameters (β). At low β (Figs. 4 and 5), results were unstable with strong fluctuations, highlighting FA's sensitivity to parameter tuning. Intermediate β (Figs. 6 and 7) led to steadier convergence, while higher β (Figs. 8 and 9) showed consistent cost minimization with reduced variance, demonstrating FA's ability to escape local minima and exploit promising regions.

5.2.2. *Extended Problem 2*

The extended scenario incorporated environmental and safety dimensions (Table 3): carbon emissions, energy usage, and safety risk indices. FA (Algorithm 3) effectively integrated these multiple objectives, penalizing infeasible solutions while ensuring mode diversity. Comparative outcomes (Table 6; Fig. 16) show that all algorithms converged to similar weighted objectives (∼833.56), but FA maintained greater variability control and avoided premature convergence.

Unlike Grey Wolf and Electron Radar, which occasionally overshoot feasible regions, FA balanced exploration and exploitation, yielding consistent solutions under stricter sustainability constraints. This underscores FA's potential for green and safe multimodal logistics

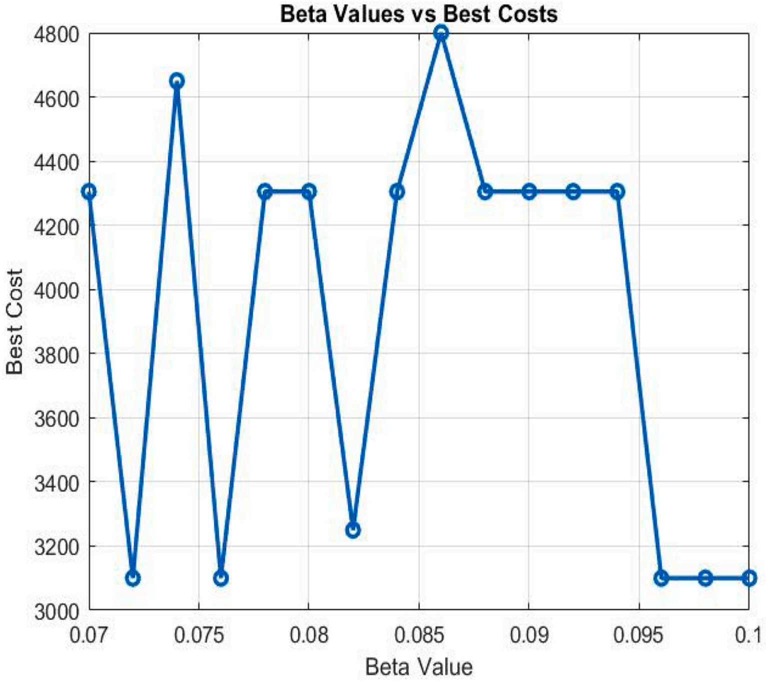

Fig. 13. Best costs when $0.07 \le \beta \le 0.1$.

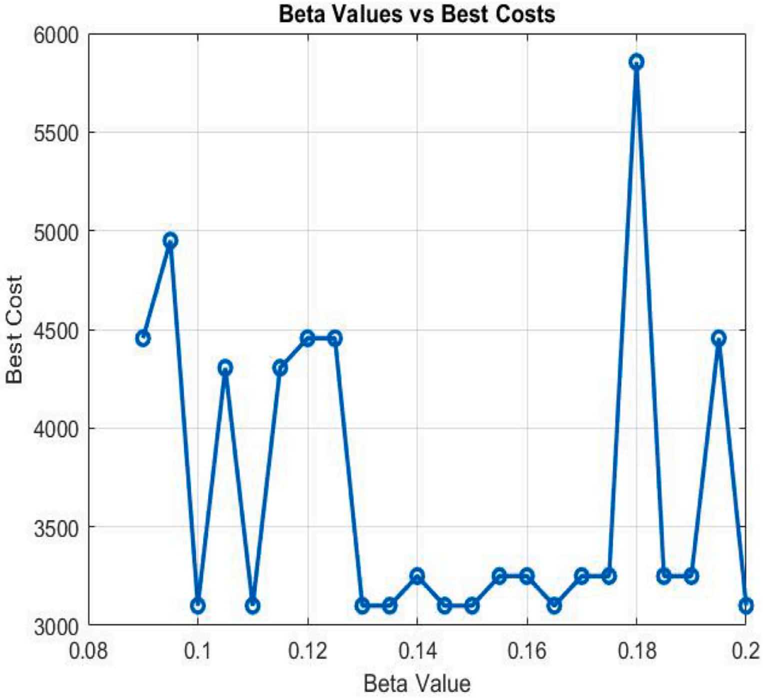

Fig. 14. Best costs when $0.09 \le \beta \le 0.2$.

planning, where minimizing not only cost and time but also emissions and safety risks is critical.

5.3. *Multimodal transportation problem (Problem 3)*

5.3.1. *Standard Problem 3*

In the four-mode network (road, rail, sea, air) with a 36 h time window, FA successfully handled the increased complexity of four consecutive legs. Figs. 17–18 and Table 7 show that FA's best configuration again produced a cost of $3100, outperforming algorithms such as GA and Electron Radar, which exhibited unstable convergence and higher-cost outliers.

5.3.2. *Extended Problem 3*

When extended with environmental and safety considerations (Table 4), Problem 3 tested scalability further. The optimization included carbon limits (900 kg), energy consumption, and safety indices. Results (Table 8; Fig. 18) demonstrate that FA maintained

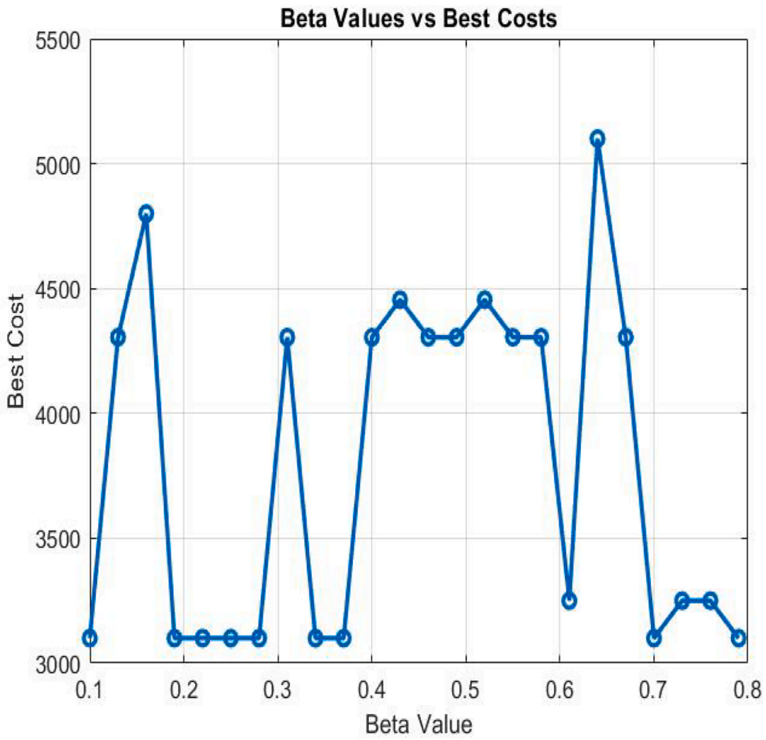

Fig. 15. Best costs when $0.1 \le \beta \le 0.5$.

Table 2
Multimodal benchmark function’s test.

ID	Function Name	Mathematical Form	Mean	Median	Best	Worst	Std
ESF1	Expanded Two-Peak Trap	$f(x) = \sum_{i=1}^{D} g_1(x_i) + 200D$ where $g_1(y) = \begin{cases} -160 + y^2, & y < 0 \\ \frac{160(y-15)}{15}, & 0 \le y \le 15 \\ \frac{200(15-y)}{5}, & 15 \le y \le 20 \\ -200 + (y-20)^2, & y > 20 \end{cases}$	1.9845e+06	2.078e+06	1.2824e+06	2.4141e+06	3.701e+05
ESF2	Expanded Five-Uneven-Peak Trap	$f(x) = \sum_{i=1}^{D} g_2(x_i) + 200D$ defined piecewise by 9 linear/quadratic segments (uneven peaks at 2.5, 5, 7.5,…,27.5)	2.2385e+06	2.2807e+06	1.5703e+06	2.9557e+06	4.7186e+05
ESF3	Expanded Equal Minima	$f(x) = \sum_{i=1}^{D} g_3(x_i) + D$, where $g_3(y) = \begin{cases} y^2, & y\langle 0 \ or \ y\rangle 1 \\ -sin^6(5\pi y), & 0 \le y \le 1 \end{cases}$	5916.2	6336.4	3241.1	7194.6	1374.7
ESF4	Expanded Decreasing Minima	$f(x) = \sum_{i=1}^{D} g_4(x_i) + D$ where $g_4(y) = \begin{cases} y^2, & y\langle 0 \ or \ y\rangle 1 \\ -\exp\left(-2ln2.\left(\frac{y-0.1}{0.8}\right)^2\right) sin^6(5\pi y), & 0 \le y \le 1 \end{cases}$	5360.6	5267.9	4230.4	6851.7	906.35
ESF5	Expanded Uneven Minima	$f(x) = \sum_{i=1}^{D} g_5(x_i) - D$, where $g_5(y) = \begin{cases} y^2, & y\langle 0 \ or \ y\rangle 1 \\ -sin^6\left(5\pi\left(y^{\frac{3}{4}} - 0.05\right)\right), & 0 \le y \le 1 \end{cases}$	5136.1	5173.2	3518.1	6274.9	892.55
ESF6	Expanded Himmelblau (pairwise)	$f(x) = \sum_{i=1}^{D/2} \left[\left(x_{2i-1}^2 + x_{2i} - 11\right)^2 + \left(x_{2i-1} + x_{2i}^2 - 7\right)^2\right]$	8.2876e+08	8.0324e+08	2.6683e+08	1.2753e+09	3.4633e+08
ESF7	Modified Vincent	$f(x) = \frac{1}{D}\sum_{i=1}^{D} g_8(x_i)$ where $g_8(y) = \begin{cases} \sin(10lny) + 1, & 0.25 \le y \le 10 \\ (0.25-y)^2 + \sin(10ln2.5) + 1, & y < 0.25 \\ (y-10)^2 + \sin(10ln10) + 1, & y > 10 \end{cases}$	2715.5	2708	1980.7	3333.9	416.08
CF1	Composition Function 1	$f(x) = \sum_{i=1}^{10} w_i(\lambda_i g_i(M_i(x - o_i) + bias_i) + 900$. Base set: Sphere, Elliptic, Bent Cigar, Discus, Different Powers (2 each)	9.6741e+05	8.7641e+05	3.9959e+05	1.5248e+06	3.9279e+05
CF2	Composition Function 2	$f(x) = \sum_{i=1}^{10} w_i(\lambda_i g_i(M_i(x - o_i) + bias_i) + 1000$. Base set: Elliptic, Different Powers, Rosenbrock, Discus, Sphere.	1.8943e+06	1.8538e+06	1.2753e+06	3.0242e+06	4.9616e+05
CF3	Composition Function 3	$f(x) = \sum_{i=1}^{10} w_i(\lambda_i g_i(M_i(x - o_i) + bias_i) + 1100$. Base set: Rosenbrock, Rastrigin, HappyCat, Scaffer’s F6, Schwefel (two of each).	2439.2	1671.7	908.06	10,407	2812.4

Table 3
Extended Problem 2.

Mode	Cost ($)	Time (h)	Emissions (kg CO_2)	Energy (L/ kWh)	Safety index
Road	500	8	120	45	0.035
Rail	700	5	80	30	0.020
Air	1000	3	360	140	0.060

competitive weighted objectives (~1368.42), comparable to TLBO and PSO, but superior in balancing exploration and avoiding wide cost fluctuations. Fig. 19

The presence of additional constraints slightly increased stochastic variation; however, FA consistently found feasible, near-optimal multimodal assignments without excessive parameter sensitivity. This confirms its applicability for large-scale, constraint-rich multimodal systems, where multiple objectives must be simultaneously satisfied.

Note: The algorithm's source code is included in the Appendix.

5.4. *Comparative insights across problems*

The consolidated performance of algorithms is summarized in Tables 5–9. While TLBO is simple and parameter-free, GA and PSO provide stability and adaptability. Brute Force guarantees optimality but lacks scalability. Grey Wolf and Electron Radar often over-explore, producing variability and occasional high-cost outliers.

By contrast, FA consistently demonstrated:

- Robustness in benchmark multimodal functions (Problem 1).
- Reliability in balancing cost and time in standard transportation problems (Problems 2–3).
- Adaptability in extended formulations with emissions, energy, and safety constraints.

The computational complexity analysis (Table 9) further highlights FA's moderate scalability (O(N^2D·I)), which is justified by its strong convergence behaviour and solution diversity. In real-world multimodal logistics, FA's ability to integrate multiple criteria while preserving computational feasibility makes it a strong candidate for sustainable, safe, and cost-effective transportation optimization.

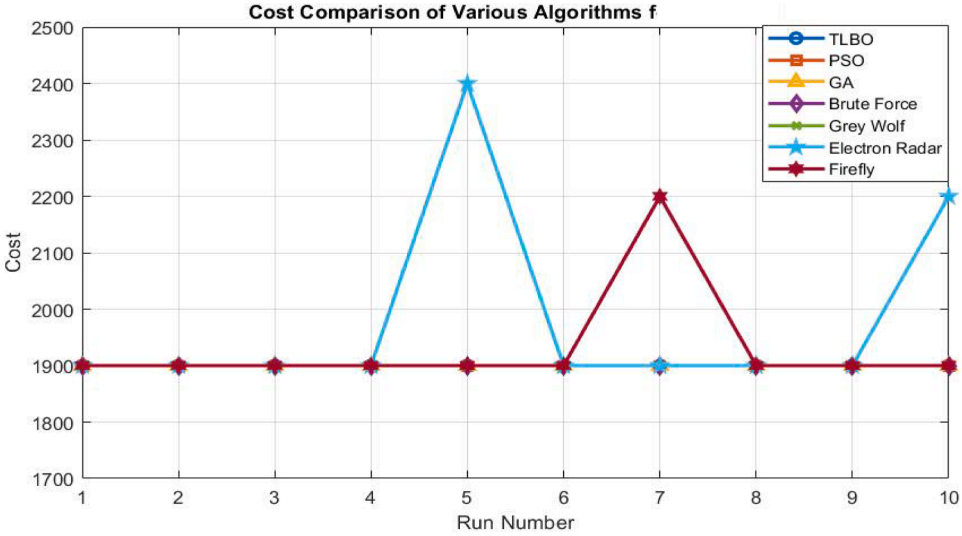

Fig. 16. Cost comparison of algorithms for Problem 2.

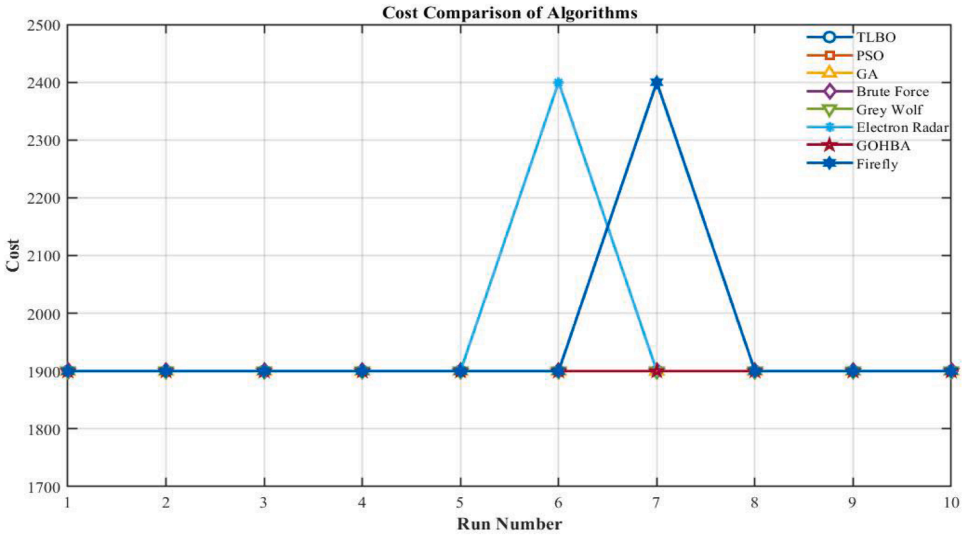

Fig. 17. Cost comparison of algorithms for extended Problem 2.

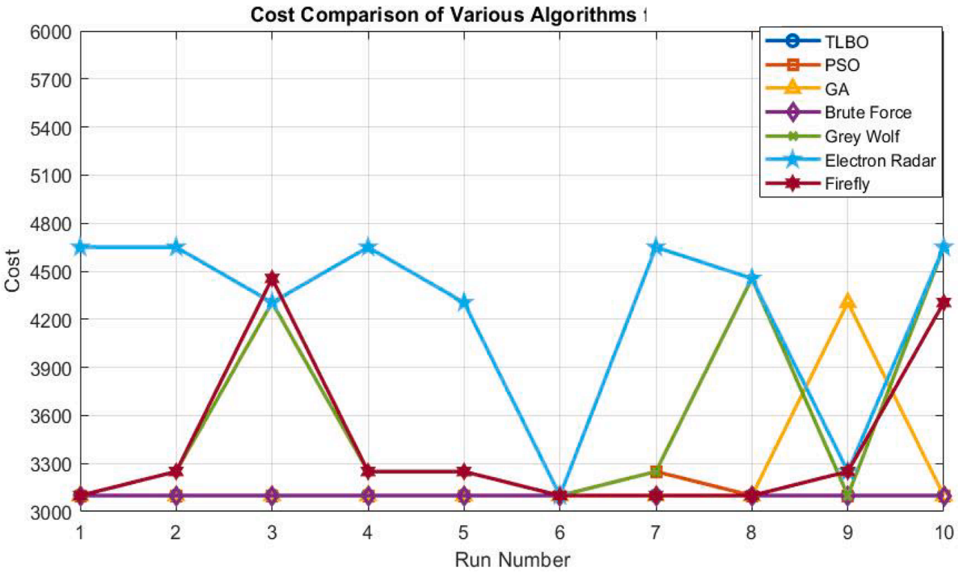

Fig. 18. Cost comparison of algorithms for Problem 3.

Table 4
Extended Problem 3.

Mode	Cost ($)	Time (h)	Emissions (kg CO_2)	Energy (L/ kWh)	Safety index
Road	700	10	150	55	0.040
Sea	1905	12	120	40	0.030
Rail	850	7	95	35	0.022
Air	2550	2	450	180	0.065

5.4.1. *Explanation of notation*

- N = number of agents (particles, fireflies, wolves, learners, chromosomes, etc.)
- D = problem dimension (number of variables; e.g., 3 for Problem 1, 4 for Problem 2)
- I = number of iterations or generations
- M = number of candidate solutions in a local neighbourhood (in ERSA)
- M^D = total number of all possible combinations (for brute force)

6. Conclusion

This study focused on MTS and used FA to optimize cost and time. The MTS is a complicated problem and yet the FA efficiently navigates it, exceeding existing optimization methods.

The system's durability and effectiveness are shown by numerical assessments of the two case studies-road, rail, and air transport in 18 h and marine transport in 36 h from the work of [67]. The FA identifies the most cost-effective and time-efficient techniques in successful optimization scenarios, proving its usefulness for real-world applications. Moreover, in our study we also did a comparative approach by juxtaposing the FA with the GA, Brute-Force, PSO, GWO, ERSA, GOHBA and TLBO [67], revealing its superior performance. This comparison not only validates the FA's efficacy but it also relevance in current and future MTS challenges.

In conclusion, this research contributes a significant piece of work to the field of logistics and supply chain management, showcasing a powerful, adaptable, and efficient tool for the optimization of MTS.

To address scalability and real-time applicability, it is worth noting that the proposed MATLAB-based methodology, while demonstrated on

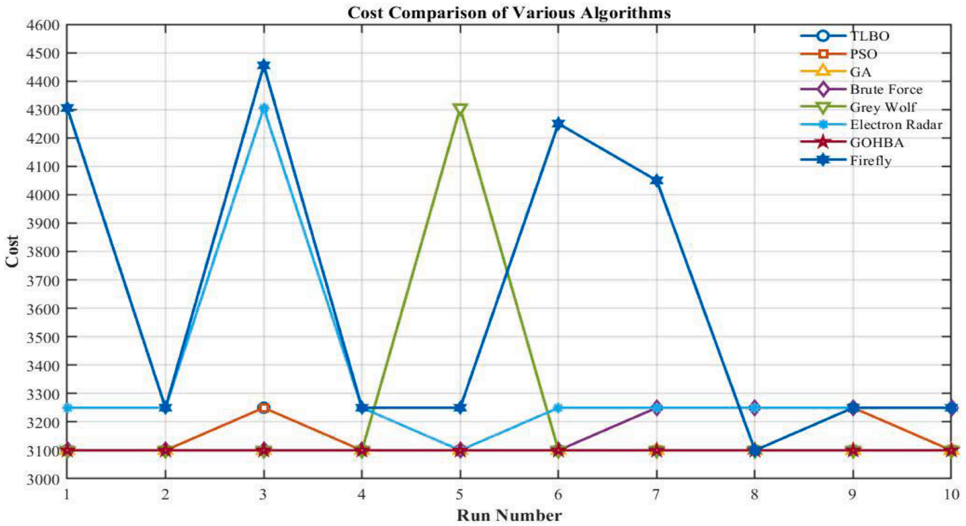

Fig. 19. Cost comparison of algorithms for extended Problem 3.

Table 5
Cost comparison of various algorithms for Problem 2.

Algorithms		No. of algorithms run									
		1	2	3	4	5	6	7	8	9	10
TLBO	No. of learners = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
PSO	No. of particles = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
GA	No. of population = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Brute Force	Based on combinations	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Grey Wolf	No. of wolves = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Electron Radar	No. of streamers = 10 No. of iterations = 10	1900	1900	1900	1900	2400	1900	1900	1900	1900	2200
GOHBA	No. of population = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Firefly	No. of fireflies = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	2200	1900	1900	1900

Table 6
Cost Comparison of various algorithms for extended Problem 2.

Algorithms		No. of algorithms run										
		1	2	3	4	5	6	7	8	9	10	Best weighted objective
TLBO	No. of learners = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	833.561250
PSO	No. of particles = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	833.561250
GA	No. of population = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	833.561250
Brute Force	Based on combinations	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	833.561250
Grey Wolf	No. of wolves = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	833.561250
Electron Radar	No. of streamers = 10 No. of iterations = 10	1900	1900	1900	1900	2400	1900	1900	1900	1900	1900	833.561250
GOHBA	No. of population = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	833.561250
Firefly	No. of fireflies = 10 No. of iterations = 10	1900	1900	1900	1900	1900	1900	1900	2400	1900	1900	833.561250

Table 7
Cost comparison of various algorithms for Problem 2.

Algorithms		No. of algorithms run									
		1	2	3	4	5	6	7	8	9	10
TLBO	No. of learners = 10 No. of iterations = 10	3100	3100	3100	3100	3100	3100	3100	3100	3100	3100
PSO	No. of particles = 10 No. of iterations = 10	3100	3100	3100	3100	3100	3250	3100	3100	3250	3100
GA	No. of population = 10 No. of iterations = 10	3100	3100	3100	3100	3100	3100	3100	3100	4305	3100
Brute Force	Based on combinations	3100	3100	3100	3100	3100	3100	3100	3100	3100	3100
Grey Wolf	No. of wolves = 10 No. of iterations = 10	3100	3250	4305	3250	3250	3100	3250	4455	3100	4650
Electron Radar	No. of streamers = 10 No. of iterations = 10	4650	4650	4305	4650	4305	3100	4650	4455	3250	4650
GOHBA	No. of population = 10 No. of iterations = 10	3100	3100	3100	3100	3100	3100	3100	3100	3100	3100
Firefly	No. of fireflies = 10 No. of iterations = 10	3100	3250	4455	3250	3250	3100	3100	3100	3250	4305

small-scale multimodal transportation problems, can be extended to large-scale networks by appropriately tuning algorithmic parameters such as population size and number of iterations. Additionally, MATLAB's parallel computing capabilities (e.g., parfor, vectorized operations) can significantly enhance computational efficiency for high-dimensional problem instances. For dynamic or real-time transportation environments, the metaheuristic algorithms used (such as Firefly, PSO, and GA) can be adapted using time-based stopping conditions, re-optimization triggers, or sliding window mechanisms to react to changing conditions. Future enhancements may also include the integration of online learning or adaptive heuristics to ensure responsiveness and robustness in ever-evolving logistics scenarios.

6.1. Limitations

Despite the effectiveness of the proposed Firefly Algorithm-based

Table 8
Cost comparison of various algorithms for extended Problem 2.

Algorithms		No. of algorithms run										
		1	2	3	4	5	6	7	8	9	10	Best weighted objective
TLBO	No. of learners = 10 No. of iterations = 10	3100	3100	3250	3100	3100	3100	3100	3100	3100	3100	1368.418600
PSO	No. of particles = 10 No. of iterations = 10	3100	3100	3250	3100	3100	3100	3100	3100	3250	3100	1368.418600
GA	No. of population = 10 No. of iterations = 10	3100	3100	3100	3100	3100	3100	3100	3100	3100	3100	1368.418600
Brute Force	Based on combinations	3100	3100	3100	3100	3100	3100	3250	3250	3250	3250	1368.418600
Grey Wolf	No. of wolves = 10 No. of iterations = 10	3100	3100	3100	3100	4305	3100	3100	3100	3100	3100	1368.418600
Electron Radar	No. of streamers = 10 No. of iterations = 10	3250	3250	4305	3250	3100	3100	3250	3100	3250	3100	1368.418600
GOHBA	No. of population = 10 No. of iterations = 10	3100	3100	3100	3100	3100	3100	3100	3100	3100	3100	1368.418600
Firefly	No. of fireflies = 10 No. of iterations = 10	4305	3250	4455	3250	3250	4305	3250	3100	3250	3250	1368.418600

Table 9
Compariosn of computational complexity, strengths and weakness in context of MTS.

Algorithms	Computational Complexity	Strengths	Weakness
TLBO	$O(N \times D \times I)$	Simple, parameter-free, provides stable and consistent optimal cost	Lacks solution diversity; may converge prematurely to local optima
PSO	$O(N \times D \times I)$	Balances exploration and exploitation, provides variability and good accuracy	Variability is moderate; can stagnate near local optima
GA	$O(N \times D \times I)$	High adaptability, introduces randomness via crossover and mutation	May require tuning; moderate variability can still miss narrow optima
Brute Force	$O(M^D)$ (M = number of modes, D = number of positions)	Guaranteed global optimum; exhaustive search	Not scalable; infeasible for large D or M; no variability
Grey Wolf	$O(N \times D \times I)$	Strong exploitation near best solutions, flexible search behavior	May over-explores; large variability leads to broader search space
Electron Radar	$O(N \times D \times M \times I)$ (M = samples per search sphere)	High exploration power; detects diverse optima; good variability	Search space is very wide; risk of skipping optimal regions; slower convergence
GOHBA	$O(N \times D \times I)$	Provides better exploration–exploitation balance through chaotic initialization, sinusoidal density factor, and golden sine strategy	Requires careful parameter tuning and adds implementation complexity
Firefly	$O(N^2 \times D \times I)$ (due to pairwise comparisons)	Maintains moderate variability and convergence toward optima	Can produce higher cost outliers in some runs due to randomness

approach in optimizing cost and time in multimodal transportation systems, the study has a few noteworthy limitations:

- The current model focuses solely on minimizing transportation cost and time

  without incorporating other important real-world factors such as carbon emissions, fuel consumption, or environmental impact.

- The case studies considered are relatively simplified and do not reflect the geographical diversity, types of transported goods, or emergency logistics scenarios that are common in practical transportation networks.
- The system does not currently account for unexpected disruptions such as route delays, accidents, or sudden changes in vehicle or mode availability, which can significantly affect solution reliability.
- Real-time variations like fluctuating traffic conditions, changing fuel prices, or weather uncertainties are not yet integrated into the model. The current approach assumes static input parameters and fixed conditions.

6.2. Future work

The proposed methodology can be further improved by integrating additional constraints into the MTS problem, such as product security and carbon emissions.

Suggestions:

Carbon Emission Constraint: Add emission values per transport mode and penalize solutions exceeding a predefined emission threshold.

Security Constraint: Assign security levels to each mode and restrict or penalize use of insecure modes for sensitive goods.

Multi-objective Formulation: Combine cost, emission, and security into a single objective using weighted aggregation.

Constraint Handling: Integrate new constraints directly into the cost function using soft penalties or hard feasibility checks.

Other recommendations:

To capitalize on the strengths of the Firefly Algorithm, future work can explore:

- Hybridization with local search methods to further refine final solutions.
- Fuzzy Firefly variants for problems involving linguistic or uncertain constraints.
- Real-time transportation planning, where adaptability to live data is critical.
- Multi-objective extensions, optimizing for time, fuel, and environmental impact along with cost.

List of Shortening

Notations	Full Form
TP	Transportation Problem
MTS	Multimodal Transportation System
GA	Genetic Algorithm
BSA	Backtracking Search Algorithm
PSO	Particle Swarm Optimizer
FA	Firefly Algorithm
GSA	Gravitational Search Algorithm
CSSA	Charged System Search Algorithm
TLBO	Teaching-Learning-Based Optimization Algorithm
BSO	Brain Storm Optimization Algorithm
GWO	Grey Wolf Optimization
ERSA	Electron Radar Search Algorithm
GOHBA	Global-Optimization Honey Badger Algorithm

Ethical approval

This article does not involve any research conducted on human participants or animals by the authors.

CRediT authorship contribution statement

Tarun Kumar: Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Kapil Kumar:** Resources, Funding acquisition, Formal analysis, Data curation. **Kailash Dhanuk:** Visualization, Software, Investigation, Funding acquisition, Formal analysis. **Anirudh Kumar Bhargava:** Validation, Resources, Methodology, Formal analysis, Data curation. **M.K. Sharma:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

All authors confirm that there are no conflicts of interest to declare.

Acknowledgments

The first author expresses gratefulness for the financial support provided by the University Grants Commission (UGC)

Appendix

Source Code

FA

```
%% ————————- User settings ————————-
w = struct(’c’,0.40,’t’,0.10,’e’,0.20,’p’,0.15,’s’,
0.15);% weights (sum to 1)
runs = 10;% independent runs
popSize = 10;% population size
maxIters = 10; % iterations per run
alpha0= 0.25; % randomness (decays)
gamma = 0.5; % light absorption
beta0 = 1.0; % base attractiveness
%% ————————- Problems ————————-
P1 = make_problem1(); % 3 legs, modes: Road/Rail/Air
P2 = make_problem2(); % 4 legs, modes: Road/Sea/Rail/
Air
fprintf(’\n=== Solving Extended Problem 1 ===\n’);
best1 = solve_with_fa(P1, w, runs, popSize, maxIters,
alpha0, gamma, beta0);
print_result(best1, P1);
fprintf(’\n=== Solving Extended Problem 2 ===\n’);
best2 = solve_with_fa(P2, w, runs, popSize, maxIters,
alpha0, gamma, beta0);
print_result(best2, P2);
end
%% ======================================
======================================
function prob = make_problem1()
% Modes: Road, Rail, Air (same per leg), 3 legs
prob.name = ’Extended Problem 1′;
prob.legs = 3;
prob.modes = ["Road","Rail","Air"];
% per-use parameters (same for all legs)
prob.c = [500, 700, 1000]; % cost $
prob.t = [ 8, 5,3]; % time h
prob.e = [120, 80, 360]; % emissions kg CO2
prob.p = [ 45, 30, 140]; % energy L/kWh
prob.s = [0.035,0.020,0.060];% safety index
% Limits
prob.Tmax = 18;
prob.Emax = 500;
prob.Pmax = 300;
prob.Smax = 0.12;
% “use >= 2 modes” across the trip
prob.requireAtLeastTwoModes = true;
end
function prob = make_problem2()
% Modes: Road, Sea, Rail, Air (same per leg), 4 legs
prob.name = ’Extended Problem 2′;
prob.legs = 4;
prob.modes = ["Road","Sea","Rail","Air"];
% per-use parameters (same for all legs)
prob.c = [700, 1905, 850, 2550];
prob.t = [ 10, 12,7,2];
prob.e = [150, 120, 95, 450];
prob.p = [ 55, 40, 35, 180];
prob.s = [0.040,0.030,0.022,0.065];
% Limits
prob.Tmax = 36;
prob.Emax = 900;
prob.Pmax = 500;
prob.Smax = 0.18;
% “use >= 2 modes”
prob.requireAtLeastTwoModes = true;
end
%% ======================================
======================================
function best = solve_with_fa(prob, w, runs, popSize,
maxIters, alpha0, gamma, beta0)
D = prob.legs; % dimensions = number of legs
M = numel(prob.modes); % number of modes
lb = ones(1,D); % each leg: mode index in {1..M}
ub = M*ones(1,D);
best.bestVal = inf;
best.x= [];
best.totals = struct();
best.runCosts = zeros(runs,1); % NEW: cost per run
best.runObjs = zeros(runs,1); % (optional) weighted
objective per run
best.runFeas = false(runs,1); % (optional) feasi-
bility flag per run
for r = 1:runs
rng(r); % reproducibility per run
% Initialize population (continuous -> discretized
for eval)
X = rand(popSize,D).*(ub-lb) + lb; % continuous
F = inf(popSize,1);
```

```
alpha = alpha0;
% Evaluate initial
for i=1:popSize
xi = discretize_mode_vector(X(i,:), lb, ub);
F(i) = fitness(xi, prob, w);
end
[bestVal, idx] = min(F);
bestX = discretize_mode_vector(X(idx,:), lb, ub);
% FA loop
for it = 1:maxIters
for i = 1:popSize
for j = 1:popSize
if F(j) < F(i)
rij = norm(X(i,:)-X(j,:));
beta = beta0 * exp(-gamma * rij^2);
step = alpha*(rand(1,D) - 0.5);
X(i,:) = X(i,:) + beta*(X(j,:) - X(i,:)) + step;
end
end
% repair to [1,M] and discretize for evaluation
X(i,:) = min(max(X(i,:), lb), ub);
xi = discretize_mode_vector(X(i,:), lb, ub);
F(i) = fitness(xi, prob, w);
end
% anneal randomness a bit
alpha = alpha * 0.98;
% track global best of this run
[curVal, idx] = min(F);
xi = discretize_mode_vector(X(idx,:), lb, ub);
if curVal < bestVal
bestVal = curVal; bestX = xi;
end
end
% Store best of this run (cost, objective, feasibility)
[Zrun, totals] = fitness(bestX, prob, w);
best.runCosts(r) = totals.C; %<– cost for this run
best.runObjs(r) = Zrun; % (optional) objective for this run
best.runFeas(r) = (totals.penalty == 0);
% Update global best if needed
if bestVal < best.bestVal
best.bestVal = bestVal;
best.x = bestX;
best.totals = totals;
end
end
end
%% ==================================================================================
function xi = discretize_mode_vector(x, lb, ub)
% Discretize: each leg picks nearest integer in [1..M]
xi = round(x);
xi = min(max(xi, lb), ub);
end
%% ==============================================================================-===
function [Z, totals] = fitness(x_choice, prob, w)
% Fitness = weighted sum + penalties for constraint violations
% x_choice: 1xD integers in 1..M (selected mode per leg)
% Totals
C=0; T=0; E=0; P=0; S=0;
for ell = 1:prob.legs
m = x_choice(ell);
C = C + prob.c(m);
T = T + prob.t(m);
E = E + prob.e(m);
P = P + prob.p(m);
S = S + prob.s(m);
end
% Weighted objective (E1)
Z = w.c*C + w.t*T + w.e*E + w.p*P + w.s*S;
% Penalties (E6–E9 and "use >=2 modes")
pen = 0;
rho = 1e6; % penalty scale (adjust if needed)
if T > prob.Tmax, pen = pen + rho*((T - prob.Tmax)/max(1,prob.Tmax)); end
if E > prob.Emax, pen = pen + rho*((E - prob.Emax)/max(1,prob.Emax)); end
if P > prob.Pmax, pen = pen + rho*((P - prob.Pmax)/max(1,prob.Pmax)); end
if S > prob.Smax, pen = pen + rho*((S - prob.Smax)/max(1,prob.Smax)); end
if isfield(prob,’requireAtLeastTwoModes’) && prob.requireAtLeastTwoModes
if numel(unique(x_choice)) < 2
pen = pen + rho;
end
end
Z = Z + pen;
if nargout > 1
totals = struct(’C’,C,’T’,T,’E’,E,’P’,P,’S’,S,’penalty’,pen);
end
end
%% ==============================================================================
function print_result(best, prob)
% Clean printing; also show cost from every run
mnames_cell = cellstr(prob.modes); % convert string array to cellstr for %s
x = best.x;
fprintf(’%s — Best weighted objective: %.6f\n’, prob.name, best.bestVal);
fprintf(’Chosen modes per leg: ’);
for i=1:numel(x)
fprintf(’%s’, mnames_cell{x(i)});
if i < numel(x), fprintf(’ ’); end
end
fprintf(’\nTotals Cost: %.2f Time: %.2f h Emissions: %.2f kg Energy: %.2f Safety: %.3f\n’, ...
best.totals.C, best.totals.T, best.totals.E, best.totals.P, best.totals.S);
% NEW: show all run-wise costs (row vector)
fprintf(’Costs from each run:\n’);
disp(best.runCosts(:)’);
% (Optional) also show feasibility per run
% fprintf(’Feasible runs (no penalty):\n’);
% disp(best.runFeas(:)’);
if best.totals.penalty > 0
fprintf(’**Note:** penalty applied = %.2e (violated a constraint)\n’, best.totals.penalty);
```

```
else
fprintf('All constraints satisfied (no penalty).\n');
end
end
```

Data availability

For inquiries regarding data availability, please contact the authors directly.

References

[1] S. Nama, S. Sharma, A.K. Saha, A.H. Gandomi, A quantum mutation-based backtracking search algorithm, Artif. Intell. Rev. 55 (2022) 3019–3073, https://doi.org/10.1007/s10462-021-10078-0.

[2] S. Chakraborty, A.K. Saha, S. Sharma, S. Mirjalili, R. Chakraborty, A novel enhanced whale optimization algorithm for global optimization, Comput. Ind. Eng. 153 (2021) 107086.

[3] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence, MIT Press, 1992.

[4] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.

[5] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the ICNN'95 – International Conference Neural on Networks 4, 1995, pp. 1942–1948.

[6] X.S. Yang, Firefly algorithms for multimodal optimization, in: Proceedings of the International Symposium on Stochastic Algorithms, Springer, Berlin, Heidelberg, 2009, pp. 169–178.

[7] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.

[8] A. Kaveh, T. Bakhshpoori, Charged system search algorithm. Metaheuristics: Outlines, MATLAB Codes and Examples, Springer, Cham, 2019, https://doi.org/10.1007/978-3-030-04067-3_8.

[9] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.

[10] Y. Shi, Brain storm optimization algorithm, in: Y. Tan, Y. Shi, Y. Chai, G. Wang (Eds.), Advances in Swarm Intelligence, ICSI 2011, Lecture Notes in Computer Science, Advances in Swarm Intelligence, ICSI 2011, Lecture Notes in Computer Science, 6728, Springer, Berlin, Heidelberg, 2011, pp. 303–311, https://doi.org/10.1007/978-3-642-21515-5_36.

[11] S. Chakraborty, A.K. Saha, A.E. Ezugwu, R. Chakraborty, A. Saha, Horizontal crossover and co-operative hunting-based whale optimization algorithm for feature selection, Knowl. Based Syst. 282 (2023) 111108.

[12] S.K. Sahoo, E.H. Houssein, M. Premkumar, A.K. Saha, M.M. Emam, Self-adaptive moth flame optimizer combined with crossover operator and Fibonacci search strategy for COVID-19 CT image segmentation, Expert Syst. Appl. 227 (2023) 120367.

[13] S. Chakraborty, A.K. Saha, A. Chhabra, Improving whale optimization algorithm with elite strategy and its application to engineering-design and cloud task scheduling problems, Cogn. Comput. (2023) 1–29.

[14] S. Sharma, A.K. Saha, G. Lohar, Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm, Eng. Comput. 38 (4) (2022) 2897–2923.

[15] S. Chakraborty, S. Sharma, A.K. Saha, A. Saha, A novel improved whale optimization algorithm to solve numerical optimization and real-world applications, Artif. Intell. Rev. (2022) 1–112.

[16] S. Sharma, A.K. Saha, A. Majumder, S. Nama, MPBOA-a novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation, Multimed. Tools Appl. 80 (2021) 12035–12076.

[17] S. Nama, A.K. Saha, S. Ghosh, Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c-Φ backfill, Appl. Soft Comput. 52 (2017) 885–897.

[18] F.S. Gharehchopogh, B. Abdollahzadeh, S. Barshandeh, B. Arasteh, A multi-objective mutation-based dynamic Harris Hawks optimization for botnet detection in IoT, Internet Things 24 (2023) 100952.

[19] O.N. Oyelade, A.E. Ezugwu, Immunity-based Ebola optimization search algorithm for minimization of feature extraction with reduction in digital mammography using CNN models, Sci. Rep. 12 (1) (2022) 17916.

[20] A.E. Ezugwu, Advanced discrete firefly algorithm with adaptive mutation-based neighborhood search for scheduling unrelated parallel machines with sequence-dependent setup times, Int. J. Intell. Syst. 37 (8) (2022) 4612–4653.

[21] M. Curtin, Brute Force, Ebook ISBN 978-0-387-27160-6, Springer, Copernicus, New York, NY, 2007, https://doi.org/10.1007/b138699.

[22] R.H. Pratt, T.J. Lomax, Performance measures for multimodal transportation systems, Transp. Res. Rec. 1518 (1) (1996) 85–93, https://doi.org/10.1177/0361198196151800115.

[23] D. Rondinelli, M. Berry, Multimodal transportation, logistics, and the environment: managing interactions in a global economy, Eur. Manag. J. 18 (4) (2000) 398–410, https://doi.org/10.1016/S0263-2373(00)00029-3.

[24] L. Castelli, R. Pesenti, W. Ukovich, Scheduling multimodal transportation systems, Eur. J. Oper. Res. 155 (3) (2004) 603–615, https://doi.org/10.1016/j.ejor.2003.02.002.

[25] K.G. Zografos, K.N. Androutsopoulos, Algorithms for itinerary planning in multimodal transportation networks, IEEE Trans. Intell. Transp. Syst. 9 (1) (2008) 175–184, https://doi.org/10.1109/TITS.2008.915650.

[26] X.S. Yang, Nature-inspired Metaheuristic Algorithms, Luniver Press, 2010.

[27] A. Kengpol, W. Meethom, M. Tuominen, The development of a decision support system in multimodal transportation routing within Greater Mekong sub-region countries, Int. J. Prod. Econ. 140 (2) (2012) 691–701, https://doi.org/10.1016/j.ijpe.2011.02.024.

[28] I. Fister, I. Fister Jr, X.S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm. Evol. Comput. 13 (2013) 34–46, https://doi.org/10.1016/j.swevo.2013.06.001.

[29] G.K. Jati, R. Manurung, Discrete firefly algorithm for traveling salesman problem: a new movement scheme. Swarm Intelligence Bio-Inspired Computation, Elsevier, 2013, pp. 295–312, https://doi.org/10.1016/B978-0-12-405163-8.00013-2.

[30] X.S. Yang, Multiobjective firefly algorithm for continuous optimization, Eng. Comput. 29 (2013) 175–184, https://doi.org/10.1007/s00366-012-0254-1.

[31] J.C. Peters, E.P. Han, S. Peeta, D. DeLaurentis, Analyzing the potential for high-speed rail as part of the multimodal transportation system in the United States' Midwest corridor, Int. J. Transp. Sci. Technol. 3 (2) (2014) 129–148, https://doi.org/10.1260/2046-0430.3.2.129.

[32] A. Rahmani, S.A. MirHassani, A hybrid firefly-genetic algorithm for the capacitated facility location problem, Inf. Sci. 283 (2014) 70–78, https://doi.org/10.1016/j.ins.2014.06.002.

[33] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.

[34] N. Nekouie, M. Yaghoobi, A new method in multimodal optimization based on firefly algorithm, Artif. Intell. Rev. 46 (2016) 267–287, https://doi.org/10.1007/s10462-016-9463-0.

[35] C. Hao, Y. Yue, Optimization on combination of transport routes and modes on dynamic programming for a container multimodal transport system, Procedia Eng. 137 (2016) 382–390, https://doi.org/10.1016/j.proeng.2016.01.272.

[36] E. Osaba, R. Carballedo, X.S. Yang, F. Diaz, An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows. Nature-Inspired Computation in Engineering, Springer, 2016, pp. 21–41, https://doi.org/10.1007/978-3-319-30235-5_2.

[37] E. Osaba, X.S. Yang, F. Diaz, E. Onieva, A.D. Masegosa, A. Perallos, A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy, Soft Comput. 21 (2017) 5295–5308, https://doi.org/10.1007/s00500-016-2114-1.

[38] R. Goel, R. Maini, A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems, J. Comput. Sci. 25 (2018) 28–37, https://doi.org/10.1016/j.jocs.2017.12.012.

[39] D. Aggarwal, V. Kumar, An improved firefly algorithm for the vehicle routing problem with time windows, in: Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2018, pp. 222–229, https://doi.org/10.1109/ICACCI.2018.8554555.

[40] T.B. Chandrawati, R.F. Sari, A review of firefly algorithms for path planning, vehicle routing and traveling salesman problems, in: Proceedings of the 2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI), IEEE, 2018, pp. 30–35, https://doi.org/10.1109/ICon-EEI.2018.8784312.

[41] R. Micale, G. Marannano, A. Giallanza, P.P. Miglietta, G.P. Agnusdei, G.La Scalia, Sustainable vehicle routing based on firefly algorithm and TOPSIS methodology, Sustain. Futures 1 (2019) 100001, https://doi.org/10.1016/j.sftr.2019.100001.

[42] D. Chen, Y. Zhang, L. Gao, R.G. Thompson, Optimizing multimodal transportation routes considering container use, Sustainability 11 (19) (2019) 5320, https://doi.org/10.3390/su11195320.

[43] S. Khalifehzadeh, M.B. Fakhrzad, A modified firefly algorithm for optimizing a multi stage supply chain network with stochastic demand and fuzzy production capacity, Comput. Ind. Eng. 133 (2019) 42–56, https://doi.org/10.1016/j.cie.2019.04.048.

[44] P.P. Matthopoulos, S. Sofianopoulou, A firefly algorithm for the heterogeneous fixed fleet vehicle routing problem, Int. J. Ind. Syst. Eng. 33 (2) (2019) 204–224, https://doi.org/10.1504/IJISE.2019.102471.

[45] G. Maity, S.K. Roy, J.L. Verdegay, Analyzing multimodal transportation problem and its application to artificial intelligence, Neural Comput. Appl. 32 (2020) 2243–2256, https://doi.org/10.1007/s00521-019-04393-5.

[46] D. Trachanatzi, M. Rigakis, M. Marinaki, Y. Marinakis, A firefly algorithm for the environmental prize-collecting vehicle routing problem, Swarm Evol. Comput. 57 (2020) 100712, https://doi.org/10.1016/j.swevo.2020.100712.

[47] J. Sun, A.C. Chow, S.M. Madanat, Multimodal transportation system protection against sea level rise, Transp. Res. Part D Transp. Environ. 88 (2020) 102568, https://doi.org/10.1016/j.trd.2020.102568.

[48] M. Elhoseny, Intelligent firefly-based algorithm with Levy distribution (FF-L) for multicast routing in vehicular communications, Expert. Syst. Appl. 140 (2020) 112889, https://doi.org/10.1016/j.eswa.2019.112889.

[49] R.K. Goel, R. Maini, Evolutionary ant colony algorithm using firefly-based transition for solving vehicle routing problems, Int. J. Comput. Sci. Eng. 21 (2) (2020) 281–288, https://doi.org/10.1504/IJCSE.2020.105736.

[50] Rahmanzadeh, M.S. Pishvaee, Electron radar search algorithm: a novel developed meta-heuristic algorithm, Soft Comput. 24 (11) (2020) 8443–8465, https://doi.org/10.1007/s00500-019-04410-8.

[51] A.M. Altabeeb, A.M. Mohsen, L. Abualigah, A. Ghallab, Solving capacitated vehicle routing problem using cooperative firefly algorithm, Appl. Soft Comput. 108 (2021) 107403, https://doi.org/10.1016/j.asoc.2021.107403.

[52] Z. Liu, Z. Wang, Q. Cheng, R. Yin, M. Wang, Estimation of urban network capacity with second-best constraints for multimodal transport systems, Transp. Res. Part B Methodol. 152 (2021) 276–294, https://doi.org/10.1016/j.trb.2021.08.011.

[53] O. Pavlenko, D. Muzylyov, N. Shramenko, D. Cagáňová, V. Ivanov, Mathematical modeling as a tool for selecting a rational logistical route in multimodal transport systems. Industry 4.0 Challenges Smart Cities, Springer, Cham, 2022, pp. 23–37, https://doi.org/10.1007/978-3-030-92968-8_2.

[54] C. Archetti, L. Peirano, M.G. Speranza, Optimization in multimodal freight transportation problems: a survey, Eur. J. Oper. Res. 299 (1) (2022) 1–20, https://doi.org/10.1016/j.ejor.2021.07.031.

[55] F. Wu, C. Lyu, Y. Liu, A personalized recommendation system for multi-modal transportation systems, Multimodal Transp. 1 (2) (2022) 100016, https://doi.org/10.1016/j.multra.2022.100016.

[56] N. Yin, Multiobjective optimization for vehicle routing optimization problem in low-carbon intelligent transportation, IEEE Trans. Intell. Transp. Syst. (2022), https://doi.org/10.1109/TITS.2022.3193679.

[57] M.D. Hina, A. Soukane, A. Ramdane-Cherif, Computational intelligence in intelligent transportation systems: an overview. Innovative Trends in Computational Intelligence, Springer, Cham, 2022, pp. 27–43, https://doi.org/10.1007/978-3-030-78284-9_2.

[58] Z.G. Chen, Z.H. Zhan, S. Kwong, J. Zhang, Evolutionary computation for intelligent transportation in smart cities: a survey, IEEE Comput. Intell. Mag. 17 (2) (2022) 83–102, https://doi.org/10.1109/MCI.2022.3155330.

[59] R. Yesodha, T. Amudha, A bio-inspired approach: firefly algorithm for multi-depot vehicle routing problem with time windows, Comput. Commun. 190 (2022) 48–56, https://doi.org/10.1016/j.comcom.2022.04.005.

[60] A. Utamima, R. Indramawan, Hybrid firefly algorithm for optimizing the garbage transport route, in: Proceedings of the 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), IEEE, 2023, pp. 204–208, https://doi.org/10.1109/3ICT60104.2023.10391780.

[61] M. Vasheghani, M. Abtahi, Strategic planning for multimodal transportation in ports, Marit. Policy Manag. 50 (7) (2023) 957–979, https://doi.org/10.1080/03088839.2022.2061060.

[62] R. Akuh, M. Zhong, A. Raza, Y. Dong, A method for evaluating the balance of land use and multimodal transport system of new towns/cities using an integrated modeling framework, Multimodal Transp. 2 (1) (2023) 100063, https://doi.org/10.1016/j.multra.2022.100063.

[63] X. Feng, R. Song, W. Yin, X. Yin, R. Zhang, Multimodal transportation network with cargo containerization technology: advantages and challenges, Transp. Policy 132 (2023) 128–143, https://doi.org/10.1016/j.tranpol.2022.12.006.

[64] Z. Lv, W. Shang, Impacts of intelligent transportation systems on energy conservation and emission reduction of transport systems: a comprehensive review, Green Technol. Sustain. 1 (1) (2023) 100002, https://doi.org/10.1016/j.grets.2022.100002.

[65] P. Wang, J. Qin, J. Li, M. Wu, S. Zhou, L. Feng, Optimal transshipment route planning method based on deep learning for multimodal transport scenarios, Electron 12 (2) (2023) 417, https://doi.org/10.3390/electronics12020417.

[66] D. Zhang, G. Zhu, Particle swarm optimization based on temporal-difference learning for solving multi-objective optimization problems, Computing 105 (8) (2023) 1795–1820, https://doi.org/10.1007/s00607-023-01166-w.

[67] T. Brar, T. Kumar, M.K. Sharma, Optimizing multimodal transportation systems using the teaching–learning-based algorithm, Int. J. Appl. Comput. Math. 10 (2024) 18, https://doi.org/10.1007/s40819-023-01655-8.

[68] Z. Wang, L. Shen, X. Li, L. Gao, An improved multi-objective firefly algorithm for energy-efficient hybrid flowshop rescheduling problem, J. Clean. Prod. 385 (2023) 135738, https://doi.org/10.1016/j.jclepro.2022.135738.

[69] S.M. Hashemi, A. Sahafi, A.M. Rahmani, M. Bohlouli, A new approach for service activation management in fog computing using Cat Swarm Optimization algorithm, Computing (2024) 1–36, https://doi.org/10.1007/s00607-024-01302-0.

[70] N. Kartli, E. Bostanci, M.S. Guzel, Heuristic algorithm for an optimal solution of fully fuzzy transportation problem, Computing (2024) 1–33, https://doi.org/10.1007/s00607-024-01319-5.

[71] K.R. Pratiba, S. Ridhanya, J. Ridhisha, P. Hemashree, Integrated Q-learning with firefly algorithm for transportation problems, EAI Endorsed Trans. Energy Web 11 (1) (2024), https://doi.org/10.4108/ew.5047.

[72] B.-Y. Qu, J.J. Liang, Z.Y. Wang, Q. Chen, P.N. Suganthan, Novel benchmark functions for continuous multimodal optimization with comparative results, Swarm Evol. Comput. 26 (2016) 23–34, https://doi.org/10.1016/j.swevo.2015.07.003.

[73] Y. Huang, S. Lu, Q. Liu, T. Han, T. Li, GOHBA: improved Honey Badger algorithm for global optimization, Biomimetics 10 (2) (2025) 92, https://doi.org/10.3390/biomimetics10020092.