

Paper Type: Original Article

Hybrid Evolutionary Approach with the Jaya Algorithm for the Multiobjective Patrolling Problem

Fumito Kudo¹, Muneaki Ohshima², Takeru Yamanaka³, Hiroaki Tohyama⁴, Masaki Tomisawa^{4,*} 

¹Department of Life Science and Informatics, Graduate School of Engineering, Maebashi Institute of Technology, Maebashi, Gunma, Japan; tomisawa@maebashi-it.ac.jp.

²Department of Communication, Ikuei Junior College, Takasaki, Gunma, Japan; ohshima@ikuei-g.ac.jp.

³Department of Life Science and Informatics, Graduate School of Engineering, Maebashi Institute of Technology, Maebashi, Gunma, Japan; tomisawa@maebashi-it.ac.jp.

⁴Department of Life Engineering, Faculty of Engineering, Maebashi Institute of Technology, Maebashi, Gunma, Japan; tohyama@maebashi-it.ac.jp; tomisawa@maebashi-it.ac.jp.

Citation:

Received: 17 January 2026

Revised: 13 November 2025

Accepted: 10 September 2025

Kudo, F., Ohshima, M., Yamanaka, T., Tohyama, H., & Tomisawa, M. (2026). Hybrid evolutionary approach with the Jaya algorithm for the multiobjective patrolling problem. *International Journal of Operations Research and Artificial Intelligence*, 2(2), 108-132.

Abstract

Routing optimization is crucial for many real-world applications, such as urban security, infrastructure inspection, and logistics. Many of these problems can be formulated as Arc Routing Problems (ARPs) on complex transportation networks. Building on our previous studies, we consider the Police Officer Patrol Problem (POPP) and its extension, the Generalized POPP (GPOPP), a bi-objective optimization problem defined on mixed graphs. The GPOPP aims to minimize patrol route length while maximizing the coverage of the guarded area, where patrol officers can visually confirm adjacent streets without physically traversing them.

However, solving the GPOPP efficiently remains challenging for two main reasons. First, conventional route operators developed for node-routing problems often produce infeasible solutions when applied to arc-routing contexts. Second, existing hybrid evolutionary approaches typically require repeated route modifications, resulting in high computational overhead and exhibit sensitivity to parameter settings.

To address these issues, this study proposes an improved hybrid evolutionary framework that integrates evolutionary computation with the Jaya algorithm. The proposed method introduces three key components: 1) a Connectivity-Based Mutation (CBM) operator designed to maintain feasible route structures in mixed graphs while generating diverse solutions, 2) a decomposed Jaya algorithm that improves search efficiency by focusing on a single objective at a time, and 3) a Fuzzy Logic Controller (FLC) that dynamically adjusts mutation and crossover probabilities.

Extensive numerical experiments using benchmark problems derived from the Mixed Rural Postman Problem (MRPP) demonstrate that the proposed method consistently outperforms both the existing hybrid method and NSGA-II in terms of solution quality and computational efficiency.

Keywords: Arc routing problem, Police officer patrolling problem, Evolutionary algorithms, MoEA-HSS, Jaya algorithm, Fuzzy logic control.

 Corresponding Author: tomisawa@maebashi-it.ac.jp

 <https://doi.org/10.48314/ijorai.v2i2.93>

 Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

1|Introduction

Routing optimization is essential for a wide range of real-world applications, such as urban security, infrastructure inspection, and logistics. These applications are often modeled as routing problems on complex transportation networks. In particular, Arc Routing Problems (ARPs) have attracted significant attention because they naturally represent tasks where network edges must be serviced or monitored.

Classical routing formulations such as the traveling salesman problem (TSP), vehicle routing problem (VRP), rural postman problem, and arc routing problem are **NP**-hard, making exact algorithms impractical for large-scale cases. Therefore, metaheuristic approaches, including genetic algorithms (GAs), ant colony optimization (ACO), and tabu search, have become standard tools for obtaining high-quality approximate solutions. Comprehensive surveys confirm that such heuristic approaches remain essential for complex routing problems [5].

Recent research has emphasized hybrid and multiobjective metaheuristics as promising directions for addressing trade-offs among travel distance, time, cost, and environmental impact. For example, memetic algorithms have demonstrated strong performance in airport ground operation routing under temporal constraints [1]. Deep reinforcement learning has been combined with the nondominated sorting genetic algorithm II (NSGA-II) to improve search efficiency in multiobjective VRPs with time windows [2]. Adaptive multiobjective GAs have also been applied to green logistics to balance economic and environmental objectives [3], while NSGA-II combined with the VIKOR method has been adopted for sustainable decision-making in green VRPs [4].

In routing optimization, various route improvement heuristics such as 2-opt, 3-opt, relocate, swap, cross-exchange, and Or-opt have been widely used as local search operators to refine route structures and improve solution quality [6, 7, 8]. These operators are commonly incorporated into metaheuristic and evolutionary algorithms to enhance search performance and maintain population diversity.

Preserving useful route structures and maintaining diversity are widely recognized as key factors for robust search performance [10, 11, 5]. However, most existing operators are designed for node-routing problems, and directly applying them to arc-routing formulations often leads to infeasible solutions.

To address this issue, several hybrid or problem-specific operators have been proposed, including edge-based local search in ant colony optimization [9], knowledge-guided operators in memetic algorithms [12], and segment-preserving crossover for truck-drone routing systems [13]. These developments reflect a broader trend toward domain-aware hybridization to balance exploration, exploitation, and feasibility.

Another promising direction in metaheuristic optimization is the Jaya algorithm proposed by Rao [14]. Jaya is a parameter-free optimization algorithm that iteratively moves candidate solutions toward the best solution while avoiding the worst solution in the population. Because it does not rely on algorithm-specific parameters, it has demonstrated robustness and conceptual simplicity across many optimization problems. These characteristics make Jaya particularly attractive for hybrid evolutionary frameworks, as it can strengthen exploitation without introducing additional algorithm-specific parameters.

Since its introduction, the Jaya algorithm has been extended to a variety of routing and transportation problems. Examples include traffic signal optimization [15], discrete adaptations for the TSP [16], shortest-path problems [17], and hybrid frameworks integrating reinforcement learning or simulated annealing [18]. More recent studies have applied Jaya to uncertain VRPs [19] and multiobjective routing formulations considering carbon emissions [20]. These studies demonstrate that the Jaya algorithm is adaptable not only to continuous optimization but also to discrete and multiobjective routing problems.

In genetic algorithms, parameter control — particularly for crossover and mutation rates — plays a crucial role in search performance [21]. Various strategies have been explored, including statistical parameter tuning via design of experiments [22, 23], adaptive control based on population diversity [24], deterministic parameter scheduling [25], and fuzzy logic control for dynamic adjustment [26, 27, 28].

In our previous work, we introduced the Police Officer Patrol Problem (POPP), a variant of the arc routing problem that has been proven to be **NP**-complete [29, 30]. The POPP was later extended to the Generalized POPP (GPOPP) [31], a bi-objective routing problem defined on mixed graphs. The GPOPP seeks optimal patrol routes for police officers in a security area represented by a mixed graph consisting of directed arcs and undirected edges. In this formulation, patrol officers may visually confirm adjacent streets without physically traversing them. The objectives are to minimize patrol route length and minimize the noncoverage ratio.

Although the hybrid evolutionary method proposed in our previous work achieved promising results, several challenges remain. In particular, repeated route modifications in mutation and local search operations lead to high computational cost, and the performance of the evolutionary search remains sensitive to parameter settings.

To address these challenges, this study develops a more efficient solution method for the GPOPP without redefining the underlying problem formulation.

The main contribution of this study is an improved hybrid evolutionary framework for solving the GPOPP. The proposed framework integrates two complementary mechanisms: a connectivity-based mutation (CBM) operator designed for arc-routing structures and a decomposed Jaya algorithm that focuses on one objective at a time.

These mechanisms play complementary roles in the evolutionary search process. The CBM operator enhances exploration by generating diverse route structures, while the decomposed Jaya algorithm strengthens exploitation by guiding solutions toward improved objective values. By integrating these mechanisms, the proposed framework improves both search efficiency and solution quality.

Extensive numerical experiments demonstrate that the proposed method significantly improves both computational efficiency and solution quality compared with the previous hybrid approach and NSGA-II.

The remainder of this paper is organized as follows. Section 2 reviews the formulation of the GPOPP. Section 3 describes the proposed hybrid evolutionary framework. Section 4 presents the experimental results. Finally, Section 5 concludes the paper and discusses future research directions.

2|Formulation

In this section, we briefly review the formulation of the GPOPP introduced in our previous work [31], which is used as the target problem in this study.

We introduced a bi-objective optimization problem based on POPP, an **NP**-complete edge routing decision problem, and proposed a heuristic algorithm for solving it in our previous work [31]. One police officer (or security guard or robot) is assigned to a security area, which they patrol. Each street a police officer traverses during a patrol is considered guarded. Furthermore, except for streets with essential facilities, such as schools, shopping markets, and banks, police officers are allowed to visually confirm each street adjacent to an intersection without traversing it.

The GPOPP is a bi-objective optimization problem with the following two objectives. The first goal is to find the patrolling route with the shortest length, while the second is to find the route with the largest guarded area. Notably, all high-security streets must be traversed.

Throughout this paper, let $\mathbb{N} = \{1, 2, 3, \dots\}$ be the set of all natural numbers. Moreover, let $I_k = \{0, 1, 2, \dots, k-1\}$ for each $k \in \mathbb{N}$.

Let $G = (V, E, A)$ be a connected simple mixed graph, where V is the set of vertices, E is the set of undirected edges and A is the set of arcs.

$\{u, v\}$ denotes an undirected edge and (u, v) indicates an arc. The term “edge” refers to either an undirected edge or an arc, denoted by $\langle u, v \rangle$; thus, $\langle u, v \rangle = \langle v, u \rangle$ if it is an undirected edge, $\langle v, u \rangle \notin A$ if it is an arc.

Let v be a vertex. Then the subsets U_v^k ($k \geq 0$) of V are defined inductively as follows:

$$U_v^0 = \{v\},$$

$$: U_v^k = \{u \mid \exists v' \in U_v^{k-1}, \langle v', u \rangle \in E \cup A\} \setminus \bigcup_{i \in I_k} U_v^i \quad (k \geq 1).$$

That is, if a vertex u is in U_v^k , then a path exists from v that reaches u via exactly k vertices, but it is not possible to reach u via fewer than $k-1$ vertices. Notably, the starting point v is not counted as a via vertex, but the final vertex, u , is.

Let m_∞ be a sufficiently large positive integer. The parameter $d_{u,v}$ is then given as follows:

- I. $d_{u,v} = d_{v,u}$ ($\forall u, v \in V$),
- II. $\langle u, v \rangle \in E \cup A$ or $\langle v, u \rangle \in E \cup A \implies d_{u,v} < m_\infty$ ($\forall u, v \in V$),
- III. $\langle u, v \rangle, \langle v, u \rangle \notin E \cup A \implies d_{u,v} = m_\infty$ ($\forall u, v \in V$),
- IV. $\langle u, v \rangle, \langle u, w \rangle, \langle w, v \rangle \in E \cup A \implies d_{u,v} \leq d_{u,w} + d_{w,v}$ ($\forall u, v, w \in V$).

Here, $d_{u,v}$ denotes the distance between u and v if an edge $\langle u, v \rangle$ (or $\langle v, u \rangle$) exists. By condition 4, G satisfies the triangle inequality. For convenience, $d_{u,v} = m_\infty$ when no edge exists between u and v .

A set H of edges on which there exist critical facilities is given as the subset of $E \cup A$. Each edge in H is considered a high-security edge and must be traversed. A sequence of vertices $s: v_1, v_2, \dots, v_k$ is considered a patrolling route on G if the following conditions hold:

- I. The sequence s is a walk; that is, $\langle v_i, v_{i+1} \rangle \in E \cup A$ for each $i \in I_k$.
- II. All edges in H are on s ; that is, if $\{u, v\} \in H$, $i \in I_k$ exists, satisfying $u = v_i$ and $v = v_{i+1}$ or $u = v_{i+1}$ and $v = v_i$. If $(u, v) \in H$, $i \in I_k$ exists, satisfying $u = v_i$ and $v = v_{i+1}$.
- III. $v_k = v_1$ that is, the walk s is closed.

The length $L(s)$ of a patrolling route s is the total sum of the distances of all edges on s , calculated as follows:

$$L(s) = \sum_{i=1}^{k-1} d_{v_i, v_{i+1}}.$$

V. $d_{u,v}$: distance between u and v .

$d_{u,v} = m_\infty$ if no edge exists between u and v . Otherwise $d_{u,v} < m_\infty$.

$$\text{VI. } c'_{u,v} = \begin{cases} 1, & \text{if } c_{u,v} + c_{v,u} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $c'_{u,v} = 1$ when an edge exists with endpoints u and v ; $c'_{u,v} = 0$ when no edge exists between them in the mixed graph G . $c'_{u,v}$ is immediately obtained from $c_{u,v}$.

The decision variable $(x_{u,v})$ for the GPOPP represents the number of times the edge $\langle u, v \rangle$ is traversed from u to v . If there exists no edge between u and v , $x_{u,v} = 0$. The auxiliary variable $y_{u,v}$ is expressed as follows:

$$\text{VII. } y_{u,v} = \begin{cases} \frac{1}{2}, & \text{if } \sum_{k=1}^n x_{u,k} > 0 \text{ and } \sum_{k=1}^n x_{v,k} > 0, \\ 1, & \text{if } \sum_{k=1}^n x_{u,k} > 0 \text{ and } \sum_{k=1}^n x_{v,k} = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Notably, $\sum_{k=1}^n x_{u,k}$ is strictly positive if and only if the vertex u is in V_s ; thus, $y_{u,v} = y_{v,u} = \frac{1}{2}$ when both u and v are on s . $y_{u,v} = 1$ and $y_{v,u} = 0$ when u is on s even though v is not on s . Note that $y_{u,v} = y_{v,u} = \frac{1}{2}$ does not mean that $\langle u, v \rangle$ or $\langle v, u \rangle$ are in E_s . Then, $\sum_{u=1}^n \sum_{v=1}^n c'_{u,v} d_{u,v} y_{u,v}$ denotes the total length of edges guarded by the patrolling route s .

Let $\mathbf{x} = (x_{1,1}, x_{1,2}, \dots, x_{n,n})$ be an n^2 -tuple of nonnegative integers. The mathematical model for the GPOPP is formulated as follows:

$$\begin{aligned} \text{minimize } f_1(\mathbf{x}) &= \sum_{u=1}^n \sum_{v=1}^n d_{u,v} x_{u,v}, \\ \text{minimize } f_2(\mathbf{x}) &= 1 - \left(\sum_{u=1}^n \sum_{v=1}^n c'_{u,v} d_{u,v} y_{u,v} \right) \left(\sum_{u=1}^{n-1} \sum_{v=u+1}^n c'_{u,v} d_{u,v} \right)^{-1}. \end{aligned}$$

The objective function f_1 is the function that minimizes the total length of the patrolling route, and f_2 is the function that minimizes the noncovered ratio.

The following constraints must be satisfied:

$$x_{u,v} \geq 0 \quad (\forall u, v \in V), \quad (1)$$

$$c_{u,v} = 0 \implies x_{u,v} = 0 \quad (\forall u, v \in V), \quad (2)$$

$$h_{u,v} = 1 \implies x_{u,v} + x_{v,u} > 0 \quad (\forall u, v \in V), \quad (3)$$

$$\sum_{v=1}^n x_{v,u} = \sum_{v=1}^n x_{u,v} \quad (\forall u \in V), \quad (4)$$

$$W \neq \phi \wedge W \neq V_{\mathbf{x}} \implies \sum_{u \in W} \sum_{v \notin W} x_{u,v} > 0 \quad (\forall W \subseteq V_{\mathbf{x}}), \quad (5)$$

where $V_{\mathbf{x}} = \left\{ u \in V \mid \sum_{v=1}^n x_{u,v} > 0 \right\}$ is the same as V_s defined in the previous section. **Constraint (1)**: The number of times to directly traverse from u to v is nonnegative. **Constraint (2)**: If no edge exists between u and v or even if there exists an arc from v to u , it is not possible to directly traverse from u to v . **Constraint (3)**:

The edge $\langle u, v \rangle$ (or $\langle v, u \rangle$) must be traversed if it is a high-security edge. Constraint (4): For any vertex u , the number of times traversed from other vertices to u is the same as the number of times traversed from u to other vertices.

Suppose that $\mathbf{x}$ does not satisfy Constraint (5). Then, a nonempty proper subset W of $V_{\mathbf{x}}$ exists that satisfies $\sum_{u \in W} \sum_{v \notin W} x_{u,v} = 0$. This means that any edge $\langle u, v \rangle$ incident to $u \in W$ and $v \in V_{\mathbf{x}} \setminus W$ is not traversed; therefore,

$\mathbf{x}$ represents two or more separate walks. In other words, based on *Constraint (5)*, the patrolling route is a continuous closed walk.

3|Proposed Method

3.1|Framework

Let $s: v_0, v_1, v_2, \dots, v_k$ be a legal patrolling route. Then, the method proposed in [31] (referred to as the “existing method” below), and the k -tuple $\mathbf{p} = (v_0, v_1, v_2, \dots, v_{k-1})$ as the chromosome representation corresponding to s was adopted. The same representation is used in the proposed method in this paper.

Let $x \pmod k$ denote the least nonnegative remainder when x is divided by k . When discussing a chromosome $\mathbf{p} = (v_0, v_1, v_2, \dots, v_{k-1})$ of length k , for simplicity of notation, we identify the integer z with $z \pmod k$ in the following. For instance, v_{15} represents the gene v_5 when $k = 10$. Let $V^{\mathbf{p}} = \{v_i \mid 0 \leq i < k\}$ be the set of genes contained in chromosome $\mathbf{p}$.

Many bi-objective optimization problems require the simultaneous optimization of two opposing objectives; therefore, it is necessary to generate a large number of Pareto-optimal solutions with incomparable properties for the decision maker. The existing method is a hybrid heuristic method that combines a hybrid sampling strategy-based multiobjective evolutionary algorithm (MoEA-HSS) with an improved Jaya algorithm for the GPOPP. The MoEA-HSS [32] is based on a hybrid sampling strategy that combines the vector evaluated genetic algorithm (VEGA) [33] with a sampling strategy based on the Pareto dominating and dominated relationship fitness function (PDDR-FF). VEGA is a sampling strategy that prioritizes exploring the edge regions of the Pareto front. In contrast, PDDR-FF is a sampling strategy that tends to converge to the central region of the Pareto front. Combining these two mechanisms maintains both convergence speed and variance performance. The Jaya algorithm modifies a given individual to move it closer to the best solution and away from the worst solution based on the best and worst candidates in the population. Applying the Jaya algorithm can increase the speed at which dominated solutions approach Pareto-optimal solutions in one generation. The existing method for GPOPP is effective through numerical experiments; however, the following problems remain.

CPU time : The mutation generates a new patrolling route by repeating the operation, deleting only one vertex on the patrolling route s or adding only one neighborhood vertex that is not on s by repeatedly (*Fig. 2.a*). The improved Jaya algorithm also repeats the same operations as the mutation to bring a randomly selected solution closer to a hypothetical solution (*Fig. 2.b*). Both operations achieve their performance gains by repeatedly executing a specific local search process, which increases computational time. The challenge remained to improve efficiency by reducing CPU time while maintaining high solution quality.

Parameter setting : The parameter combinations used by the existing method were determined from a small number of preliminary experiments with small-scale benchmark problems. It is necessary to introduce a procedure to find the optimal parameter combinations.

To solve these two problems, this paper introduces a new method based on the framework of existing methods, incorporating newly improved mutation and Jaya algorithms, along with a fuzzy logic controller (FLC) for parameter adjustment. *Fig. 3* presents the framework of the proposed method. The remaining sections show

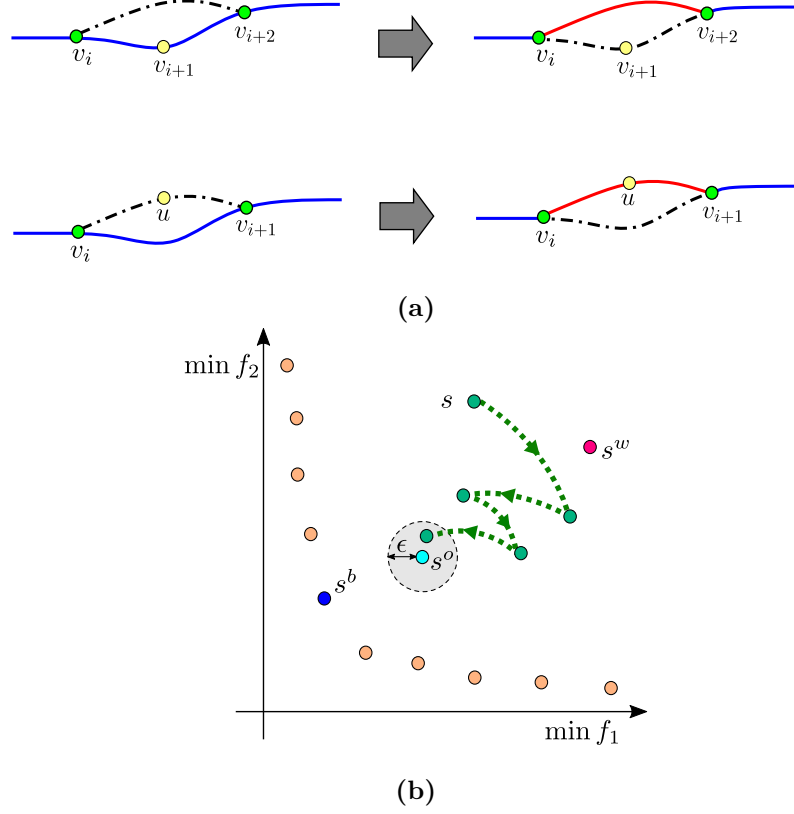


Fig. 2. a. Local Search used in the mutation, b. An image of the local search iterations in the improved Jaya.

A new mutation operation, a new Jaya algorithm, and the FLC. For the crossover operation, we use the same one as the existing method.

3.2|Connectivity-Based Mutation (CBM)

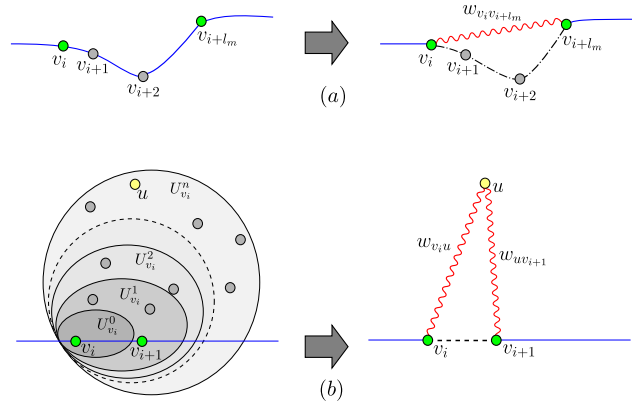
Let l_m be a fixed positive integer and $s: v_0, v_1, v_2, \dots, v_{k-1}, v_k$ be a patrolling route. In the proposed CBM, we randomly select one vertex v_i on s and one of the following processes is performed:

- I. Remove the subroute $v_i, v_{i+1}, v_{i+2}, \dots$, and v_{i+l_m} from s and replace it with the shortest route $w_{v_i v_{i+l_m}}$ from v_i to v_{i+l_m} (Fig. 4.a).
- II. Randomly select a vertex u in $U_{v_i}^{l_m}$ that is reachable from v_i via exactly l_m vertices. Let $w_{v_i u}$ and $w_{uv_{i+1}}$ be the shortest routes from v_i to u and from u to v_{i+1} , respectively. Then remove the subroute v_i, v_{i+1} from s and replace it with the shortest route $w_{v_i u v_{i+1}}$ via u , where $w_{v_i u v_{i+1}}$ is generated by combining $w_{v_i u}$ with $w_{uv_{i+1}}$ (Fig. 4.b).

Fig. 5 presents the pseudocode for CBM. By setting a sufficiently large l_m , CBM maintains the diversity of the population. It is also expected to reduce computational time by eliminating the iterative local search process that occurs in mutations in existing methods.

3.3|Decomposed Jaya Algorithm

Let ϵ be a sufficiently small positive real number and $\mathbf{p}^b$ and $\mathbf{p}^w$ be the best and worst chromosomes in the current population, respectively; s^b and s^w are the corresponding patrolling routes. The improved Jaya algorithm used in existing methods generates a virtual solution s^o from s^b and s^w , operating to bring a dominant solution

Procedure: Proposed Method**Input:** data set of problem and parameters used by evolutionary algorithm**Output:** Pareto-optimal solutions E **begin** $t := 1;$ set crossover rate $\alpha(t)$ and mutation rate $\beta(t);$ set archive $A(t)$ to be empty and randomly create $P(t)$ by Initial Population;calculate the objective functions $f_1(\mathbf{p})$ and $f_2(\mathbf{p})$ for each individual $\mathbf{p}$ in $P(t);$ calculate the fitness $eval(\mathbf{p})$ for each individual $\mathbf{p}$ in $P(t);$ create Pareto $E(P(t))$ and keep the best Pareto solution;**while** ($t \leq \text{MaxGen}$) **do**create mating pools $M_1(t)$ and $M_2(t)$ from $P(t)$ by VEGA;combine mating pools $M_1(t)$, $M_2(t)$ and the archive $A(t)$ as integrated mating pool $M(t);$ create $P(t+1)$ from $M(t)$ by Crossover, Connectivity-Based Mutation, and Decomposed Jaya;calculate objective functions $f_1(\mathbf{p})$ and $f_2(\mathbf{p})$ for each $\mathbf{p}$ in $A(t)$ and $P(t+1);$ update crossover rate $\alpha(t+1)$ and mutation rate $\beta(t+1)$ by Fuzzy Logic Controller;calculate fitness $eval(\mathbf{p})$ for each $\mathbf{p}$ in $A(t)$ and $P(t+1);$ update archive $A(t+1)$ from $P(t+1)$ and $A(t);$ update Pareto $E(P(t+1))$ and update the best Pareto solution; $t := t + 1;$ **return** Pareto-optimal solutions $E(P(t))$ **end;****Fig. 3. Framework of the proposed method.****Fig. 4. Connectivity-based mutation.**

s closer to s° . Because a trade-off exists between the two objective functions of GPOPP, s approaches s° along an inefficient zigzag trajectory; thus, the number of iterations of deleting a vertex on s and adding a nearby vertex not on s tends to increase.

The proposed method, decomposed Jaya algorithm, focuses on only one objective function and ignores the other, linearly approaching s to a set reference value. In other words, an objective function f_g ($g \in \{1, 2\}$) is randomly selected, a reference value o_g is set, and a procedure similar to CBM is used on s to bring the value of f_g closer to o_g . The other objective function tends to increase; however, the reduction in iterative procedures is expected

Procedure: Connectivity-Based Mutation

Input: individual $\mathbf{p} = (v_0, v_1, v_2, \dots, v_{k-1})$, fixed positive integer l_m

Output: individual $\mathbf{q}$

begin

randomly select one gene $v_i \in V^p$;

randomly generate $r \in [0, 1]$;

if $r \leq 0.5$ **then**

get the shortest path $w_{v_i v_{i+l_m}}$ from Dijkstra's algorithm;

generate $\mathbf{q} = (v_0, v_1, \dots, v_{i-1}, w_{v_i v_{i+l_m}}, v_{i+l_m+1}, \dots, v_{k-1})$ from $\mathbf{p}$;

else

randomly select $u \in U_{v_i}^{l_m}$;

get the shortest paths $w_{v_i u}$ and $w_{u v_{i+1}}$ from Dijkstra's algorithm;

generate $\mathbf{q} = (v_0, v_1, \dots, v_{i-1}, w_{v_i u v_{i+1}}, v_{i+2}, \dots, v_{k-1})$ from $\mathbf{p}$;

return $\mathbf{q}$

end;

Fig. 5. Pseudocode for connectivity-based mutation operation.

to reduce execution time. Figure (6) presents images of the decomposed Jaya algorithm's behavior and Figure (7) presents the pseudocode for the decomposed Jaya algorithm.

3.4|Fuzzy Logic Controller (FLC)

The proposed method introduces an FLC [26] to tune two parameters: the mutation probability α and the crossover probability β (Figure (3)).

Let $t (1 \leq t \leq \text{MaxGen})$ be the current generation number; $\alpha(t)$ and $\beta(t)$ are the mutation probability and crossover probability in the t -th generation, respectively. $\alpha(1)$ and $\beta(1)$ are given as parameters. Let $S_{\text{pop}}(t)$ be the set of nondominated solutions in the population just before beginning the processing of the t -th generation. After the completing processing of the t -th generation, the mutation probability $\alpha(t+1)$ and the crossover probability $\beta(t+1)$ in the $(t+1)$ -th generation are calculated as follows:

- (1) Let $S_{\text{off}}(t)$ be the set of nondominated solutions in the offspring-only population generated by the t -th generation processing. Then, for each $x \in S_{\text{off}}(t)$, let $\bar{x}$ be a solution of $S_{\text{pop}}(t)$ that satisfies the following condition:

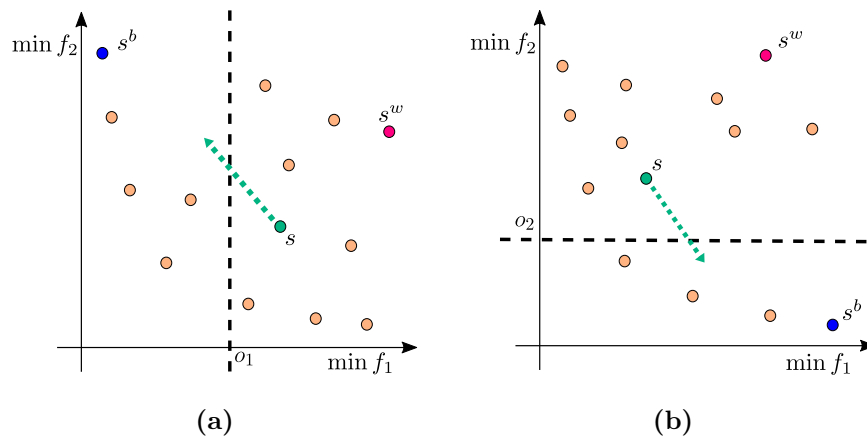


Fig. 6. Images of decomposed jaya algorithm's behavior.

Procedure: Decomposed Jaya Algorithm

Input: individual $\mathbf{p} = (v_0, v_1, v_2, \dots, v_{k-1})$ and $(f_1^b, f_2^b), (f_1^w, f_2^w) \in \mathbb{N}^2$, fixed positive integer l_j

Output: one individual $\mathbf{q}$

begin

$t := 0$;

$\mathbf{q} := \mathbf{p}$;

// let s be a patrolling route corresponding to $\mathbf{q}$

$f_1 := L(s)$;

$f_2 := \overline{cov}(s)$;

randomly generate $r, r_{1,1}, r_{1,2}, r_{2,1}, r_{2,2} \in [0, 1]$;

if $r > 0.5$ **then**

$o_1 := \max \{0, f_1 + r_{1,1} (f_1^b - f_1) - r_{1,2} (f_1^w - f_1)\}$;

while $f_1 - o_1 > \epsilon$ and $t < T$ **do**

randomly select one gene $v_i \in V^q$;

get the shortest path $w_{v_i v_{i+l_j}}$ from Dijkstra's algorithm;

generate $\mathbf{q}' = (v_0, v_1, \dots, v_{i-1}, w_{v_i v_{i+l_j}}, v_{i+l_j+1}, \dots, v_{k-1})$ from $\mathbf{q}$;

$\mathbf{q} := \mathbf{q}'$;

// Let the new chromosome $\mathbf{q}$ be regarded as the sequence $(v_0, v_1, v_2, \dots, v_{k-1})$

// and let s be a patrolling route corresponding to it.

$f_1 := L(s)$;

$t := t + 1$;

else

$o_2 := \max \{0, f_2 + r_{2,1} (f_2^b - f_2) - r_{2,2} (f_2^w - f_2)\}$;

while $f_2 - o_2 > \epsilon$ and $t < T$ **do**

randomly select one gene $v_i \in V^q$;

randomly select $u \in U_{v_i}^{l_j}$;

get the shortest paths $w_{v_i u}$ and $w_{u v_{i+1}}$ from Dijkstra's algorithm;

generate $\mathbf{q}' = (v_0, v_1, \dots, v_{i-1}, w_{v_i u}, v_{i+1}, w_{u v_{i+1}}, v_{i+2}, \dots, v_{k-1})$ from $\mathbf{q}$;

$\mathbf{q} := \mathbf{q}'$;

// Let the new chromosome $\mathbf{q}$ be regarded as the sequence $(v_0, v_1, v_2, \dots, v_{k-1})$

// and let s be a patrolling route corresponding to it.

$f_2 := \overline{cov}(s)$;

$t := t + 1$;

return $\mathbf{q}$

end;

Fig. 7. Pseudocode for the Decomposed Jaya Algorithm.

$$\forall \bar{y} \in S_{\text{pop}}(t), d_{x\bar{x}} \leq d_{x\bar{y}}.$$

Here $d_{z_1 z_2} = \sqrt{(f_1(z_2) - f_1(z_1))^2 + (f_2(z_2) - f_2(z_1))^2}$ is the distance between the solutions z_1 and z_2 .

Then, we calculate the average distance $D(t)$ between $S_{\text{off}}(t)$ and $S_{\text{pop}}(t)$ in the t -th generation:

$$D(t) = \frac{1}{|S_{\text{off}}|} \sum_{x \in S_{\text{off}}} \hat{d}_{x\bar{x}}.$$

Here,

$$\hat{d}_{x\bar{x}} = \begin{cases} d_{x\bar{x}}, & x \text{ is the nondominated solution in } S_{\text{pop}}(t) \cup S_{\text{off}}(t), \\ -d_{x\bar{x}}, & \text{otherwise.} \end{cases}$$

- (2) Using the fuzzy decision table shown in figure (8. a), the control action policy for the change in crossover probability $\Delta c(t)$ and mutation probability $\Delta m(t)$ are determined from the average distance $D(t-1)$ and $D(t)$ as follows:
- (a) The probabilities of the next generation are increased if $\epsilon \leq D(t-1) \leq \gamma$ and $\epsilon \leq D(t) \leq \gamma$.
 - (b) The probabilities of the next generation are decreased if $-\gamma \leq D(t-1) \leq -\epsilon$ and $-\gamma \leq D(t) \leq -\epsilon$.
 - (c) The probabilities of the next generation are increased if $-\epsilon \leq D(t-1) \leq \epsilon$ and $-\epsilon \leq D(t) \leq \epsilon$.

We calculate as $D(0) = 0$ when $t = 1$.

Figure (8.b) presents the membership function used in the proposed method. Here, $\epsilon = 0.002$ is a sufficiently small real number, and $\gamma = 0.02$ is the maximum value of the membership function ($-\gamma$ is the minimum value of the membership function).

- (3) The normalized value, $Z'(i, j)$, of the control action is obtained using the look-up table in figure (8.c). The changes in the crossover probability $\Delta c(t)$ and mutation probability $\Delta m(t)$ are then calculated, respectively.

$$\begin{aligned}\Delta c(t) &= r_1 \hat{u} Z'(i, j), \\ \Delta m(t) &= r_2 \hat{u} Z'(i, j).\end{aligned}$$

Here, we set the adaptation coefficients to $r_1 = 0.005$ and $r_2 = 0.001$.

- (4) The crossover probability $\alpha(t+1)$ and mutation probability $\beta(t+1)$ of the $(t+1)$ -th generation are calculated as follows:

$$\begin{aligned}\alpha(t+1) &= \alpha(t) + \Delta c(t), \\ \beta(t+1) &= \beta(t) + \Delta m(t).\end{aligned}$$

4|Numerical Experiments

This section presents the results of numerical experiments conducted to verify the effectiveness of the proposed method (referred to as pMH2 below) for GPOPP. In these experiments, we compare pMH2 with the existing method (referred to as pMH below) introduced in [31] and the method using NSGA-II (referred to as NSGA-II below) proposed by [34].

Section 4.1 presents the results of numerical experiments confirming that pMH2 is superior to pMH in terms of solution accuracy and execution time. Section 4.2 presents the results of verification experiments on the effects of CBM and decomposed Jaya algorithm, as introduced in pMH2, and on the effect of applying FLC to pMH2. In previous numerical experiments [31], pMH was compared with NSGA-II and found to have superior solution accuracy; however, NSGA-II had a significant advantage in terms of execution time. Section 4.3 shows the results of experiments to confirm how much pMH2 improved performance compared to NSGA-II at this point.

Evaluating the performance of multiobjective optimization methods is challenging because it is difficult to compare the Pareto-optimal solutions (also called nondominated solutions) obtained by different methods. Therefore, we compared each method using three representative evaluation criteria.

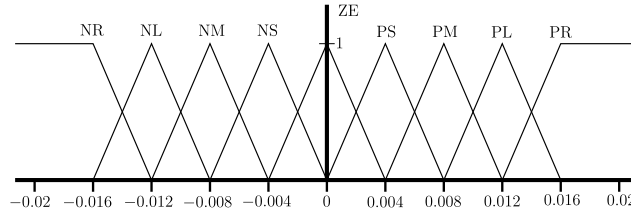
Let N_{ex} be the number of numerical experiments for each method. Suppose we conduct a comparative experiment using two methods, μ and ν . Let $\mathcal{S}_\mu^{[i]}$ and $\mathcal{S}_\nu^{[i]}$ be the nondominated solution sets obtained in the i -th numerical experiment for each method, respectively.

Let $\mathcal{S}_\mu$ and $\mathcal{S}_\nu$ be the nondominated solution sets obtained in $\bigcup_{i=1}^{N_{ex}} \mathcal{S}_\mu^{[i]}$ and $\bigcup_{i=1}^{N_{ex}} \mathcal{S}_\nu^{[i]}$, respectively.

$Z(D(t-1), D(t))$		$D(t)$									
		NR	NL	NM	NS	ZE	PS	PM	PL	PR	
$D(t-1)$	NR	NR	NL	NL	NM	NM	NS	NS	ZE	ZE	
	NL	NL	NL	NM	NM	NS	NS	ZE	ZE	PS	
	NM	NL	NM	NM	NS	NS	ZE	ZE	PS	PS	
	NS	NM	NM	NS	NS	ZE	ZE	PS	PS	PM	
	ZE	NM	NS	NS	ZE	PM	PS	PS	PM	PM	
	PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PL	
	PM	NS	ZE	ZE	PS	PS	PM	PM	PL	PL	
	PL	ZE	ZE	PS	PS	PM	PM	PL	PL	PR	
	PR	ZE	PS	PS	PM	PM	PL	PL	PR	PR	

Note: NR = negative larger; NL = negative large; NM = negative medium; NS = negative small; ZE = zero; PS = positive small; PM = positive medium; PL = positive large; PR = positive larger.

a. Fuzzy Decision table.



b. Membership function.

$Z'(i, j)$		j									
		-4	-3	-2	-1	0	1	2	3	4	
i	-4	-4	-3	-3	-2	-2	-1	-1	0	0	
	-3	-3	-3	-2	-2	-1	-1	0	0	1	
	-2	-3	-2	-2	-1	-1	0	0	1	1	
	-1	-2	-2	-1	-1	0	0	1	1	2	
	0	-2	-1	-1	0	2	1	1	2	2	
	1	-1	-1	0	0	1	1	2	2	3	
	2	-1	0	0	1	1	2	2	3	3	
	3	0	0	1	1	2	2	3	3	4	
	4	0	1	1	2	2	3	3	4	4	

c. Look-up table

Fig. 8. Fuzzy Logic Controller.

The reference solution set $\mathcal{S}_{\mu, \nu}^*$ is defined as the set of nondominated solutions in $\bigcup_{i=1}^{N_{ex}} (\mathcal{S}_{\mu}^{[i]} \cup \mathcal{S}_{\nu}^{[i]})$.

The three evaluation criteria used in the experiment are as follows:

I. Coverage $C(\mathcal{S}_{\mu}^{[i]}, \mathcal{S}_{\nu}^{[j]})$ [35]: $C(\mathcal{S}_{\mu}^{[i]}, \mathcal{S}_{\nu}^{[j]})$ is the ratio of individuals in $\mathcal{S}_{\nu}^{[j]}$ dominated by $\mathcal{S}_{\mu}^{[i]}$. The larger this C value (closer to 1), the better $\mathcal{S}_{\mu}^{[i]}$ is than $\mathcal{S}_{\nu}^{[j]}$.

II. Hypervolume $HV(\mathcal{S}_{\mu}, \mathcal{S}_{\nu})$ [35]: $HV(\mathcal{S}_{\mu}, \mathcal{S}_{\nu})$ is the space size dominated by the set $\mathcal{S}_{\mu}$ of nondominated solutions obtained by the method μ . A larger HV value means better convergence performance. The reference point for computing HV is obtained from $\mathcal{S}_{\mu} \cup \mathcal{S}_{\nu}$.

III. Inverted generational distance $IGD(\mathcal{S}_{\mu}, \mathcal{S}_{\nu})$ [36]: $IGD(\mathcal{S}_{\mu}, \mathcal{S}_{\nu})$ is the average distance from each reference solution in $\mathcal{S}_{\mu, \nu}^*$ to the nearest solution in $\mathcal{S}_{\mu}$. The smaller the IGD value, the better the convergence and distribution performances are considered to be.

The benchmark problems used in the experiments were generated from well-known Mixed Rural Postman Problem (MRPP) benchmark problems RB422 ($|V| = 357$, $|E| = 674$, $|A| = 210$), RB452 ($|V| = 465$, $|E| = 796$, $|A| = 259$), RB472 ($|V| = 498$, $|E| = 849$, $|A| = 265$), RB522 ($|V| = 388$, $|E| = 828$, $|A| = 278$), RB552 ($|V| = 488$, $|E| = 987$, $|A| = 331$) and RB572 ($|V| = 498$, $|E| = 986$, $|A| = 340$) which are publicly available in the Arc Routing Problems: Data Instances repository (<https://www.uv.es/corberan/instancias.htm>).

Since no publicly available benchmark datasets currently exist for the GPOPP with visual confirmation, we generated experimental problems by modifying these MRPP benchmarks. In the original MRPP problems, the number of required edges is relatively large, which is not suitable for evaluating the effect of visual confirmation in the GPOPP. Therefore, in our experiments, the required edges in each MRPP problem were converted into nonrequired edges. We then defined a subset of edges as high-security edges. For each benchmark problem, four experimental patterns were created by setting the number of high-security edges to 10, 20, 30, and 40. High-security edges were randomly selected from the set of edges in each benchmark problem. For each problems, the selection was performed once and the same set was used for all algorithms to ensure a fair comparison. Furthermore, we set the ratio of high-security undirected edges to high-security arcs to 4 : 1.

The experiments were conducted on a PC with Windows 11, an Intel(R) Core(TM) i5-12400 CPU, 2.50 GHz, and 8.0 GB RAM. The parameters were set as follows.

The initial population size: $n(0) = 300$; the initial crossover rate: $\alpha(1) = 0.3$; the initial mutation rate: $\beta(1) = 0.05$; the terminating condition for the decomposed Jaya algorithm: $T = 50$; and the maximum generations: $\text{MaxGen} = 300$. As denoted in Section 3.4, parameters used by FLC operation are set to $\gamma = 0.02$, $\epsilon = 0.002$, $r_1 = 0.005$, and $r_2 = 0.001$. Numerical experiments for each method were performed $N_{ex} = 10$ times.

This section denotes the proposed method as pMH2, and the existing method [31] is denoted as pMH. The proposed method is labeled with a four-digit code representing its configuration. From left to right, the digits correspond to the following configuration: l_m is the input of CBM, l_j is the input of decomposed Jaya algorithm, the activation of decomposed Jaya algorithm, and the activation of FLC. In the numerical experiments, l_m is set to 1 or 6, and l_j is set to 0 or 2. For example, pMH2-1000 represents the proposed method with $l_m = 1$, $l_j = 0$, decomposed Jaya algorithm disabled, and FLC disabled. The configuration that achieved the best performance, pMH2-6211, corresponds to $l_m = 6$, $l_j = 2$, decomposed Jaya algorithm enabled, and FLC enabled.

4.1|Performance Comparison of pMH2-6211 with pMH

We compared the performance of pMH and pMH2-6211 using the benchmark problem RB422_10. pMH2-6211 represents the proposed method with the best-performing configuration.

The pMH achieved good results in terms of evaluation criteria but required a relatively long execution time. Accordingly, one main objective of the pMH2 was to enhance computational efficiency.

The two methods differ in their frameworks; thus, a direct comparison of execution times was not possible. Instead, we evaluated how many generations pMH2-6211 could execute within the execution time required for pMH to complete 25 generations. The results showed that pMH2-6211 could run up to 300 generations within the same execution time, indicating that it was approximately about 12 times faster than pMH.

The evaluation criteria are shown in *Fig. 9* and *Table 1*. (*Fig. 9.a*) shows that the HV values indicate that pMH2-6211 clearly outperforms pMH, demonstrating its superior convergence performance. The IGD results presented in (*Fig. 9.b*) show, that pMH2-6211 achieves better average values, implying that it can generate nondominated solutions closer to the reference solution set than pMH. The C values presented in (*Fig.*

9.c) show that pMH2-6211 also performs better overall than pMH. This indicates that a higher proportion of solutions obtained by pMH2-6211 dominate those of pMH, indicating higher accuracy of the nondominated

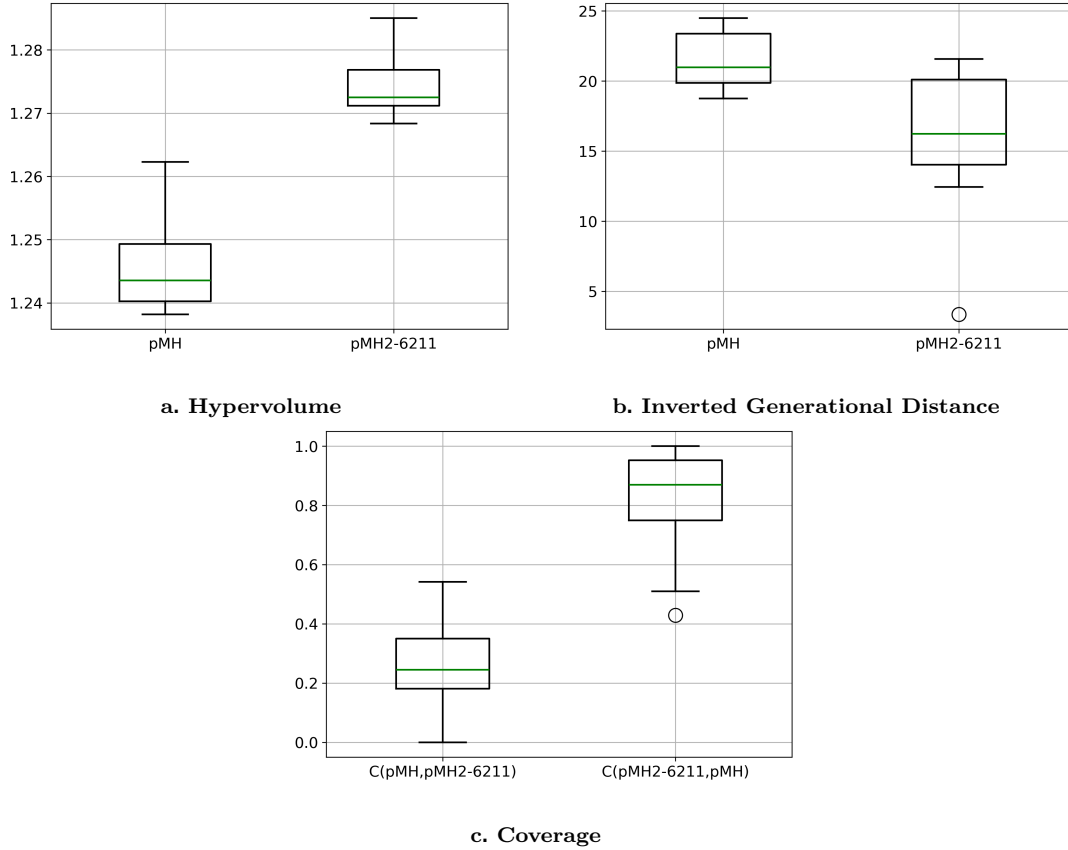


Fig. 9. Boxplots of HV, IGD, and C by pMH and pMH2-6211.

Table 1. Comparison of pMH2-6211 with pMH (Benchmark Problem: RB422_10).

(a) Hypervolume and Inverted Generational Distance

Method	$HV(S_\mu^{[i]})$				$IGD(S_\mu^{[i]})$			
	min	max	ave	std	min	max	ave	std
pMH	1.238	1.262	1.246	0.008	18.757	24.483	21.397	2.190
pMH2-6211	1.268	1.285	1.274	0.005	3.368	21.571	15.872	5.428

(b) Coverage of two sets

Method		$C(S_\mu^{[i]}, S_\nu^{[j]})$			
μ	ν	min	max	ave	std
pMH	pMH2-6211	0.00	0.54	0.26	0.13
pMH2-6211	pMH2	0.43	1.00	0.84	0.13

solutions. Therefore, pMH2-6211 achieves shorter execution time while maintaining high solution quality and can be considered a superior approach compared with pMH.

4.2|Experimental Evaluation of CBM, Decomposed Jaya Algorithm, and FLC

The configuration pMH2-1000 disables both decomposed Jaya algorithm and FLC, and because $l_m = 1$, the mutation operator targets only the next vertex. This configuration represents the least effective version of the proposed method. Using pMH2-1000 as a baseline, two numerical experiments were conducted to examine the

effects of decomposed Jaya algorithm and FLC. In both experiments, the benchmark problem RB422_10 was executed for 100 generations, and the results were compared.

First, to evaluate the contribution of decomposed Jaya algorithm, a comparative experiment was conducted between pMH2-1000 and pMH2-6210, in which decomposed Jaya algorithm was enabled. As shown in figure (10) and table (2), pMH2-6210 consistently outperformed pMH2-1000 across all three evaluation metrics, demonstrating that decomposed Jaya algorithm provides a substantial improvement in performance.

Next, to examine the effect of FLC, a comparative experiment was conducted between pMH2-6210 and pMH2-6211, both of which have FLC enabled. The results are shown in figure (11) and table (3). These results demonstrate the effectiveness of FLC. The difference is not as pronounced as that observed with decomposed Jaya algorithm; however, pMH2-6211 shows better minimum, maximum, and mean values than pMH2-6210 across all three evaluation metrics. These findings confirm the effectiveness of FLC.

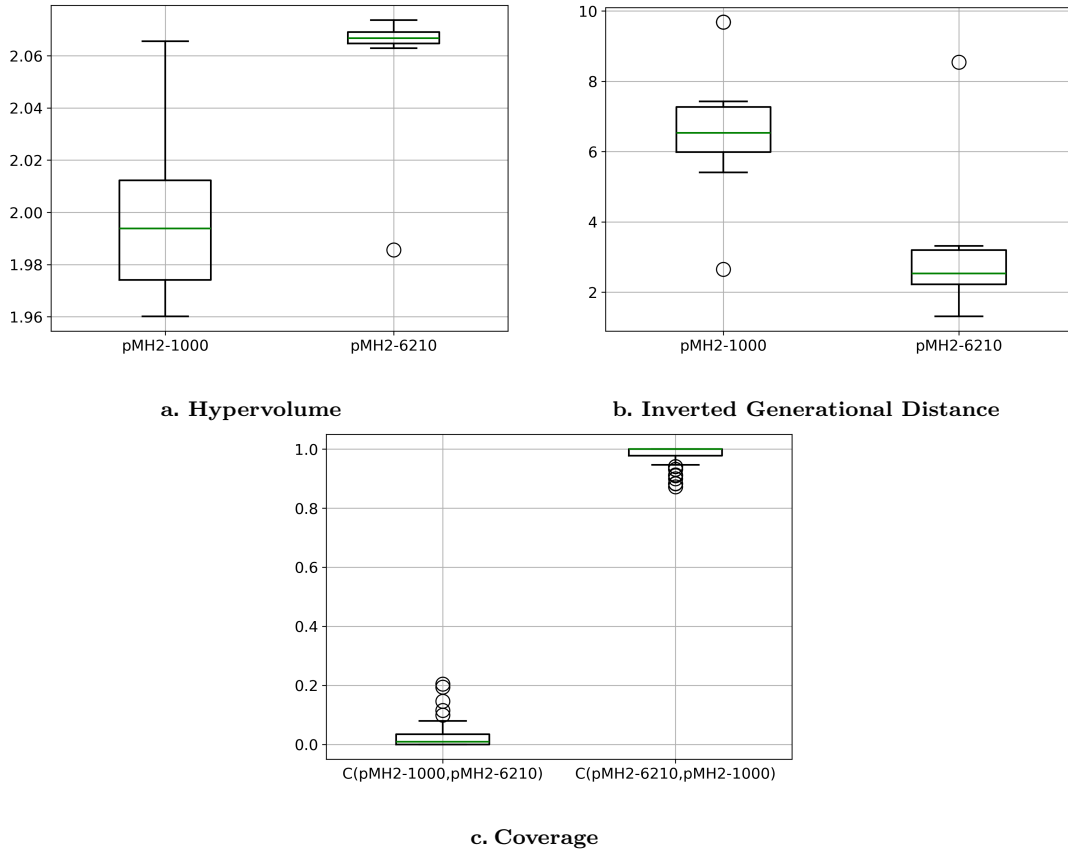


Fig. 10. Boxplots of HV, IGD, and C by pMH2-1000 and pMH2-6210.

Table 2. Comparison of pMH2-6210 with pMH2-1000 (Benchmark Problem: RB422_10, gen = 100).

(a) Hypervolume, Inverted Generational Distance, and CPU Time

Method	$HV(S_\mu^{[i]})$				$IGD(S_\mu^{[i]})$				CPU time [sec] ave
	min	max	ave	std	min	max	ave	std	
pMH2-1000	1.960	2.066	1.998	0.031	2.653	9.681	6.669	2.040	54.062
pMH2-6210	1.986	2.074	2.060	0.026	1.312	8.542	3.043	2.048	57.248

(b) Coverage of two sets

Method		$C(S_\mu^{[i]}, S_\nu^{[j]})$			
μ	ν	min	max	ave	std
pMH2-1000	pMH2-6210	0.00	0.20	0.02	0.04
pMH2-6210	pMH2-1000	0.87	1.00	0.98	0.03

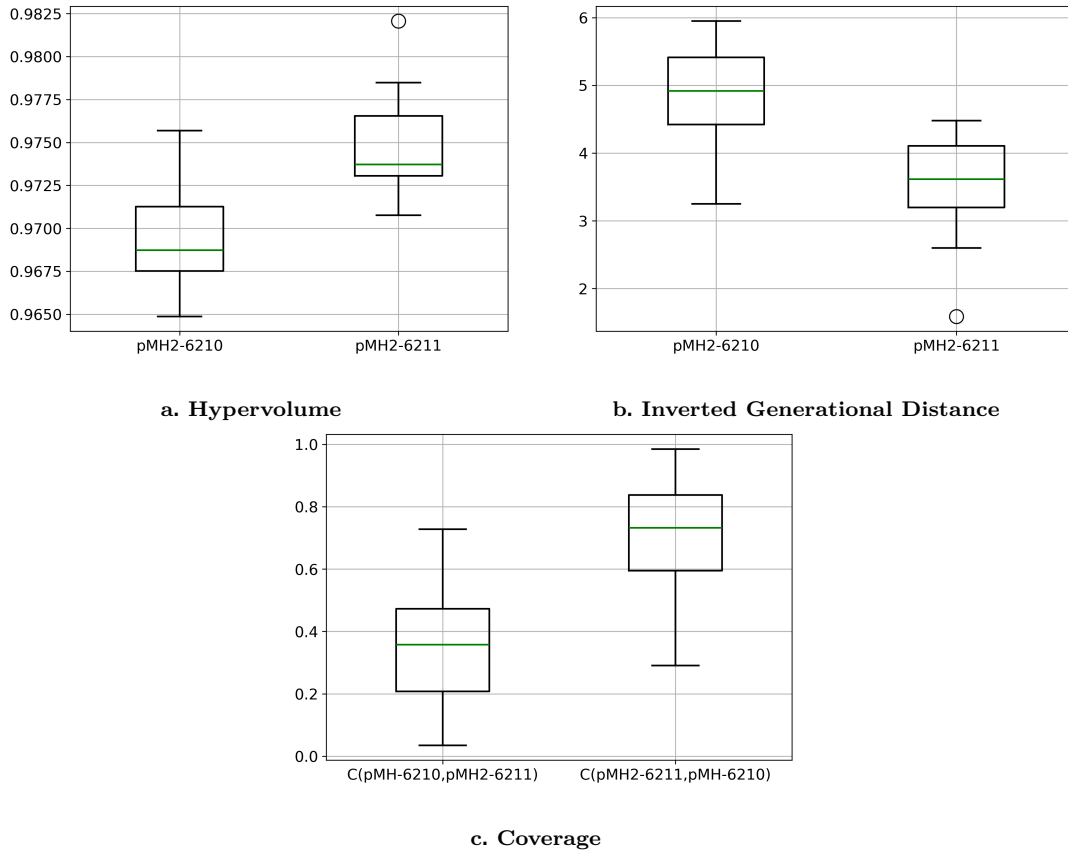


Fig. 11. Boxplots of HV, IGD, and C by pMH2-6210 and pMH2-6211.

Table 3. Comparison of pMH2-6211 with pMH2-6210 (Benchmark Problem: RB422_10, gen=100).

(a) Hypervolume, Inverted Generational Distance, and CPU time									
Method	$HV(\mathcal{S}_\mu^{[i]})$				$IGD(\mathcal{S}_\mu^{[i]})$				CPU time [sec]
	min	max	ave	std	min	max	ave	std	ave
pMH2-6210	0.965	0.976	0.970	0.003	3.248	5.948	4.788	0.867	48.931
pMH2-6211	0.971	0.982	0.975	0.003	1.589	4.477	3.493	0.903	58.659

(b) Coverage of two sets							
Method				$C(\mathcal{S}_\mu^{[i]}, \mathcal{S}_\nu^{[j]})$			
μ	ν			min	max	ave	std
pMH2-6210	pMH2-6211			0.03	0.73	0.35	0.17
pMH2-6211	pMH2-6210			0.29	0.98	0.71	0.17

4.3|Performance Comparison of pMH2-6211 with NSGA-II-6211

A performance comparison was conducted between pMH2-6211 and NSGA-II-6211. Since both methods share the configuration code 6211, they represent the best-performing variants within their respective frameworks. The experimental results are presented in figure (12) and tables (4) and (5).

From the tables, it can be observed that pMH2-6211 does not consistently achieve the best performance in terms of the minimum and maximum values across all benchmark problems. In contrast, when comparing the average values of the three evaluation metrics (HV, IGD, and C), it is evident that pMH2-6211 generally outperforms NSGA-II-6211.

For the HV metric, the minimal standard deviation indicates high stability, and the comparison of mean values confirms that pMH2-6211 performs better in terms of HV. For the IGD metric, the standard deviation of NSGA-II-6211 is larger than that of pMH2-6211. NSGA-II-6211 occasionally achieves better minimum values; however, it often exhibits worse maximum values, suggesting that a smaller mean IGD value corresponds to a more stable and higher-quality nondominated solution set. Therefore, the comparison of mean values demonstrates that pMH2-6211 outperforms NSGA-II-6211 in the IGD metric. For the C metric, both methods frequently produced cases where the maximum value was 1 and the minimum value was 0. Moreover, no significant difference was observed in the standard deviations between the two methods. Hence, the comparison of mean values indicates that pMH2-6211 outperforms NSGA-II-6211 in terms of the C metric.

Overall, these results suggest that the nondominated solution set obtained by pMH2-6211 is statistically superior to that produced by NSGA-II-6211. Consequently, we can conclude that the proposed CBM, decomposed Jaya algorithm, and FLC exhibit high effectiveness compared to pMH and demonstrate strong performance against NSGA-II.

Although the execution time of pMH2-6211 is slower than that of NSGA-II-6211, a significant performance improvement over pMH was confirmed. Since both methods use the proposed operations, this difference in execution speed is due to NSGA-II and MoEA-HSS, not the proposed method.

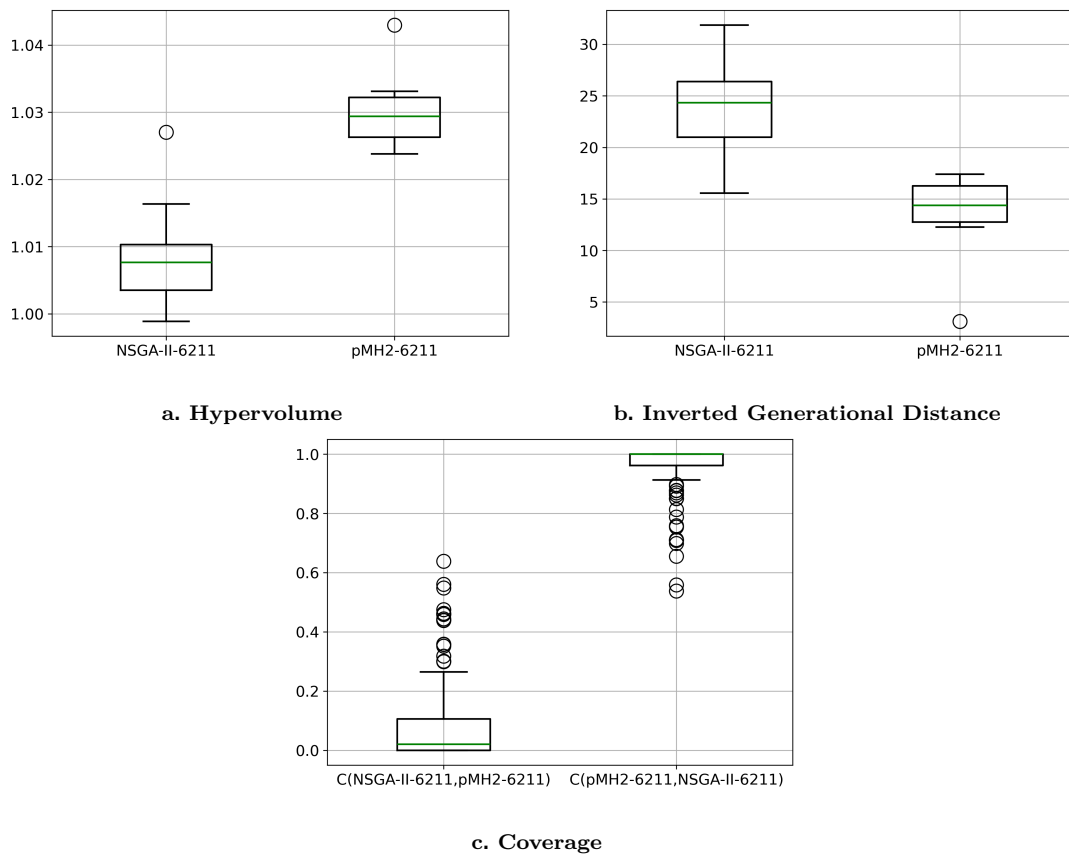


Fig.12. Boxplots of HV, IGD, and C by NSGA-II-6211 and pMH2-6211 (Benchmark Problem: RB422_10).

Table 4. Comparison of pMH2-6211 with NSGA-II-6211 by the Hypervolume, IGD, and CPU time (gen = 300).

Problem	Method	$HV_{\mu}^{[t]}$				$IGD(S_{\mu}^{[t]})$				CPU time [sec] ave
		min	max	ave	std	min	max	ave	std	
RB422_10	NSGA-II-6211	0.999	1.027	1.009	0.008	15.565	31.855	24.125	4.886	83.825
	pMH2-6211	1.024	1.043	1.030	0.005	3.135	17.404	13.635	4.114	138.238
RB422_20	NSGA-II-6211	0.310	0.362	0.332	0.021	2.017	44.568	25.472	16.383	194.137
	pMH2-6211	0.328	0.360	0.346	0.013	4.907	30.258	15.329	10.733	262.374
RB422_30	NSGA-II-6211	0.128	0.178	0.169	0.015	1.623	69.109	14.240	19.955	284.223
	pMH2-6211	0.165	0.178	0.170	0.004	6.810	19.255	12.847	4.395	394.950
RB422_40	NSGA-II-6211	0.134	0.182	0.163	0.019	1.023	35.266	13.495	14.679	362.264
	pMH2-6211	0.164	0.181	0.171	0.005	1.805	13.341	6.876	3.605	487.629
RB452_10	NSGA-II-6211	1.406	1.457	1.419	0.014	4.320	51.561	41.263	13.558	158.289
	pMH2-6211	1.420	1.452	1.430	0.010	12.657	46.585	34.797	11.056	208.143
RB452_20	NSGA-II-6211	0.286	0.340	0.321	0.020	1.661	63.309	26.308	25.188	334.900
	pMH2-6211	0.317	0.339	0.330	0.007	7.266	33.575	15.253	8.009	694.117
RB452_30	NSGA-II-6211	0.195	0.261	0.248	0.020	1.961	38.181	8.742	10.788	485.125
	pMH2-6211	0.244	0.267	0.255	0.006	0.501	11.233	6.064	3.424	863.171
RB452_40	NSGA-II-6211	0.127	0.185	0.171	0.017	1.058	71.069	17.769	21.013	599.043
	pMH2-6211	0.159	0.178	0.171	0.006	8.858	34.587	17.981	7.421	788.062
RB472_10	NSGA-II-6211	2.351	2.477	2.408	0.047	5.869	103.487	67.297	34.140	194.594
	pMH2-6211	2.421	2.489	2.467	0.022	6.766	76.483	30.606	22.169	306.318
RB472_20	NSGA-II-6211	0.388	0.474	0.426	0.028	1.010	64.432	35.705	19.937	555.196
	pMH2-6211	0.383	0.476	0.435	0.028	2.289	68.769	30.013	20.772	655.423
RB472_30	NSGA-II-6211	0.501	0.634	0.550	0.047	0.003	94.236	58.623	31.761	850.782
	pMH2-6211	0.536	0.612	0.561	0.028	13.383	74.127	52.109	20.847	1030.154
RB472_40	NSGA-II-6211	0.376	0.460	0.437	0.028	1.252	86.315	23.445	28.211	1169.335
	pMH2-6211	0.430	0.458	0.444	0.008	3.515	35.241	17.624	9.178	1639.330
RB522_10	NSGA-II-6211	0.792	0.835	0.811	0.015	1.226	61.729	33.705	21.935	127.196
	pMH2-6211	0.805	0.826	0.814	0.008	13.957	49.420	31.864	12.671	177.485
RB522_20	NSGA-II-6211	0.283	0.323	0.304	0.013	0.989	55.557	26.237	18.701	194.191
	pMH2-6211	0.300	0.319	0.310	0.005	7.450	34.991	18.568	7.698	270.528
RB522_30	NSGA-II-6211	0.151	0.188	0.176	0.014	0.500	45.371	14.584	16.563	234.343
	pMH2-6211	0.157	0.185	0.179	0.008	4.644	38.633	10.882	10.051	339.206
RB522_40	NSGA-II-6211	0.095	0.131	0.121	0.013	1.806	29.349	8.824	9.697	344.774
	pMH2-6211	0.118	0.129	0.123	0.003	2.469	10.832	6.891	2.623	493.563
RB552_10	NSGA-II-6211	1.076	1.110	1.088	0.011	7.315	21.488	17.030	4.406	165.783
	pMH2-6211	1.083	1.117	1.099	0.010	2.038	19.951	12.972	5.449	242.913
RB552_20	NSGA-II-6211	0.311	0.361	0.341	0.014	3.412	41.223	17.191	11.473	299.309
	pMH2-6211	0.331	0.365	0.343	0.010	1.900	26.404	14.287	6.583	617.155
RB552_30	NSGA-II-6211	0.194	0.242	0.222	0.020	3.508	45.784	21.513	17.320	474.002
	pMH2-6211	0.228	0.242	0.234	0.004	4.667	17.051	10.925	4.411	856.795
RB552_40	NSGA-II-6211	0.121	0.160	0.149	0.011	0.566	28.156	7.563	7.698	489.442
	pMH2-6211	0.144	0.160	0.152	0.005	0.656	11.852	6.012	3.682	732.059
RB572_10	NSGA-II-6211	0.893	0.937	0.909	0.015	1.062	34.812	24.167	10.691	163.842
	pMH2-6211	0.903	0.934	0.918	0.010	7.035	30.694	18.815	7.992	230.342
RB572_20	NSGA-II-6211	0.293	0.339	0.318	0.015	2.238	35.477	16.996	10.617	289.218
	pMH2-6211	0.299	0.340	0.322	0.012	2.477	30.460	14.228	8.472	536.533
RB572_30	NSGA-II-6211	0.224	0.282	0.263	0.020	0.504	43.966	13.546	14.633	353.588
	pMH2-6211	0.269	0.276	0.273	0.002	4.878	9.942	6.591	1.442	727.968
RB572_40	NSGA-II-6211	0.082	0.095	0.090	0.005	3.785	17.841	9.310	4.700	614.397
	pMH2-6211	0.085	0.100	0.091	0.005	1.335	13.846	8.274	3.958	800.045

Table 5. Comparison of pMH2-6211 with NSGA-II-6211 by Coverage (gen = 300).

Problem	Methods		$C(\mathcal{S}_\mu^{[i]}, \mathcal{S}_\nu^{[j]})$			
	μ	ν	min	max	ave	std
RB422_10	NSGA-II-6211	pMH2-6211	0.00	0.64	0.10	0.16
	pMH2-6211	NSGA-II-6211	0.54	1.00	0.95	0.10
RB422_20	NSGA-II-6211	pMH2-6211	0.00	1.00	0.34	0.43
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.69	0.41
RB422_30	NSGA-II-6211	pMH2-6211	0.00	1.00	0.67	0.38
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.37	0.38
RB422_40	NSGA-II-6211	pMH2-6211	0.00	1.00	0.45	0.42
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.56	0.40
RB452_10	NSGA-II-6211	pMH2-6211	0.00	1.00	0.38	0.27
	pMH2-6211	NSGA-II-6211	0.02	1.00	0.73	0.26
RB452_20	NSGA-II-6211	pMH2-6211	0.00	1.00	0.49	0.43
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.54	0.42
RB452_30	NSGA-II-6211	pMH2-6211	0.00	1.00	0.54	0.40
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.51	0.39
RB452_40	NSGA-II-6211	pMH2-6211	0.00	1.00	0.37	0.42
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.67	0.41
RB472_10	NSGA-II-6211	pMH2-6211	0.00	1.00	0.23	0.33
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.82	0.31
RB472_20	NSGA-II-6211	pMH2-6211	0.00	1.00	0.42	0.42
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.63	0.42
RB472_30	NSGA-II-6211	pMH2-6211	0.00	1.00	0.39	0.46
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.62	0.46
RB472_40	NSGA-II-6211	pMH2-6211	0.00	1.00	0.55	0.42
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.48	0.42
RB522_10	NSGA-II-6211	pMH2-6211	0.00	1.00	0.38	0.36
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.68	0.35
RB522_20	NSGA-II-6211	pMH2-6211	0.00	1.00	0.49	0.38
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.56	0.37
RB522_30	NSGA-II-6211	pMH2-6211	0.00	1.00	0.55	0.44
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.48	0.43
RB522_40	NSGA-II-6211	pMH2-6211	0.00	1.00	0.38	0.38
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.66	0.36
RB552_10	NSGA-II-6211	pMH2-6211	0.00	1.00	0.42	0.35
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.65	0.35
RB552_20	NSGA-II-6211	pMH2-6211	0.00	1.00	0.50	0.38
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.55	0.40
RB552_30	NSGA-II-6211	pMH2-6211	0.00	1.00	0.44	0.43
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.58	0.42
RB552_40	NSGA-II-6211	pMH2-6211	0.00	1.00	0.48	0.40
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.55	0.41
RB572_10	NSGA-II-6211	pMH2-6211	0.00	1.00	0.38	0.33
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.67	0.34
RB572_20	NSGA-II-6211	pMH2-6211	0.00	1.00	0.45	0.41
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.61	0.41
RB572_30	NSGA-II-6211	pMH2-6211	0.00	1.00	0.44	0.40
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.58	0.38
RB572_40	NSGA-II-6211	pMH2-6211	0.00	1.00	0.53	0.39
	pMH2-6211	NSGA-II-6211	0.00	1.00	0.52	0.38

5|Conclusions

This study presented an enhanced hybrid evolutionary framework for the GPOPP, a bi-objective routing optimization model defined on mixed graphs. The proposed method integrates three synergistic mechanisms: a CBM that preserves population diversity while reducing local search iterations, the decomposed Jaya algorithm that performs focusing on only one objective function, and an FLC that adaptively tunes crossover and mutation parameters according to population dynamics. Extensive numerical experiments demonstrate that the proposed framework (pMH2) achieves improved runtime compared to the existing hybrid heuristic method (pMH) while maintaining or improving solution quality. The analysis confirmed that CBM and decomposed Jaya algorithm effectively enhanced exploration and convergence efficiency, In contrast, FLC provided stable and adaptive parameter control throughout evolution. Compared with NSGA-II, pMH2 exhibited superior performance in hypervolume, inverted generational distance, and coverage metrics, indicating strong robustness and scalability for large-scale mixed-graph routing problems. Future studies can extend this framework to cooperative multi-agent patrolling and dynamic environments, incorporating learning-based adaptive control to improve real-time optimization performance.

Acknowledgments

The authors would like to express their sincere gratitude to the editors and anonymous reviewers for their invaluable comments and constructive feedback, which significantly contributed to the enhancement of this paper.

Author Contribution

All authors contributed equally to this manuscript, including implementing the algorithms, performing experiments, analyzing the results, drafting the manuscript, and reviewing the final version.

Funding

The authors declare that no external funding or support was received for the research presented in this paper, including administrative, technical, or in-kind contributions. Data Availability All data supporting the reported findings in this research paper are provided within the manuscript.

Conflicts of Interest

The authors declare that there is no conflict of interest concerning the reported research findings. Funders played no role in the study's design, in the collection, analysis, or interpretation of the data, in the writing of the manuscript, or in the decision to publish the results.

References

- [1] Beke, L., Uribe, L., Lara, A., Artemio Coello Coello, C., Weisz, M., Burke, E. K., & Chen, J. (2024). Routing and scheduling in multigraphs with time constraints—a memetic approach for airport ground movement. *IEEE Transactions on evolutionary computation*, 28(2), 474–488. <https://doi.org/10.1109/TEVC.2023.3262743>
- [2] Wu, R., Wang, R., Hao, Jie., Wu, Q., Wang, P., & Niyato, D. (2025). Multiobjective vehicle routing optimization with time windows: A hybrid approach using deep reinforcement learning and NSGA-II. *IEEE transactions on intelligent transportation systems*, 26(3), 4032–4047. <https://doi.org/10.1109/TITS.2024.3515997>
- [3] Zhao, W., Bian, X., & Mei, X. (2024). An adaptive multiobjective genetic algorithm for solving heterogeneous green city vehicle routing problem. *Multidisciplinary digital publishing institute*, 14(15), 6594. <https://doi.org/10.3390/app14156594>
- [4] Dutta, J., Barma, P. S., Mukherjee, A., Kar, S., De, T., Pamučar, D., ... & Garbinčius, G. (2021). Multi-objective green mixed vehicle routing problem under rough environment. *Transport*, 37(1), 51-63.. <https://doi.org/10.3846/transport.2021.14464>
- [5] Liu, Fei., Lu, C., Gui, L., Zhang, Q., Tong, X., & Yuan, M. (2023). Heuristics for vehicle routing problem: a survey and recent advances. *arXiv:2303.04147*. <https://doi.org/10.48550/arXiv.2303.04147>
- [6] Potvin, J.-Y. (2009). State-of-the art review—evolutionary algorithms for vehicle routing. *INFORMS Journal on Computing*, 21(4), 518–548. <https://doi.org/10.1287/ijoc.1080.0312>

- [7] Gendreau, M., Potvin, J. Y., Bräumlaysy, O., Hasle, G., & Løkketangen, A. (2008). Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In *The vehicle routing problem: latest advances and new challenges* (pp. 143-169). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-77778-8_7
- [8] Ochelska-Mierzejewska, J., Poniszewska-Marańda, A., & Marańda, W. (2021). Selected genetic algorithms for vehicle routing problem solving. *Electronics*, 10(24), 3147. <https://doi.org/10.3390/electronics10243147>
- [9] Lacomme, P., Prins, C., & Tanguy, A. (2004, September). First competitive ant colony scheme for the CARP. In *international workshop on ant colony optimization and swarm intelligence* (pp. 426-427). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-28646-2_48
- [10] Pavai, G., & Geetha, T. V. (2016). A survey on crossover operators *ACM computing surveys (CSUR)*, 49(4), 1-43. <https://doi.org/10.1145/3009966>
- [11] Alorf, A. (2023). A survey of recently developed metaheuristics and their comparative analysis. *Engineering applications of artificial intelligence*, 117, 105622. <https://doi.org/10.1016/j.engappai.2022.105622>
- [12] Li, Q., Liu, S., Chen, W., Zou, J., Tang, K., & Yao, X. (2025). Knowledge-Guided memetic algorithm for capacitated arc routing problems with time-dependent service costs. *arXiv preprint arXiv:2507.21740*. <https://doi.org/10.48550/arXiv.2507.21740>
- [13] Sobhanan, A., Charkhgard, H., & Kwon, C. (2025). Arc Routing Problems with Multiple Trucks and Drones: A Hybrid Genetic Algorithm. *arXiv preprint arXiv:2508.18105*. <https://doi.org/10.48550/arXiv.2508.18105>
- [14] Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International journal of industrial engineering computations*, 7(1), 19-34. <https://doi.org/10.5267/j.ijiec.2015.8.004>
- [15] Gao, K., Zhang, Y., Sadollah, A., & Su, R. (2016, November). Jaya algorithm for solving urban traffic signal control problem. In *2016 14th international conference on control, automation, robotics and vision (ICARCV)* (pp. 1-6). IEEE. <https://doi.org/10.1109/ICARCV.2016.7838661>
- [16] Gunduz, M., & Aslan, M. (2021). DJAYA: A discrete Jaya algorithm for solving traveling salesman problem. *Applied soft computing*, 105, 107275. <https://doi.org/10.1016/j.asoc.2021.107275>
- [17] Wang, R., Zhou, M., Wang, J., & Gao, K. (2023). An improved discrete Jaya algorithm for shortest path problems in transportation-related processes. *Processes*, 11(8), 2447. <https://doi.org/10.3390/pr11082447>
- [18] Xu, J., Hu, W., Gu, W., & Yu, Y. (2023). A discrete jaya algorithm based on reinforcement learning and simulated annealing for the traveling salesman problem. *Mathematics*, 11(14), 3221. <https://doi.org/10.3390/math11143221>
- [19] Zhang, J., Ye, J. X., Lin, J., & Song, H. B. (2024). A discrete Jaya algorithm for vehicle routing problems with uncertain demands. *Systems science & control engineering*, 12(1), 2350165. <https://doi.org/10.1080/21642583.2024.2350165>
- [20] Bajaj, A., & Dhodiya, J. (2025). Multi-Route Multi-Objective TSP: mathematical model and its solution by reference point and aspiration level-based MOJO Jaya algorithm. *Neural computing and applications*, 37(16), 10243-10285. <https://doi.org/10.1007/s00521-025-11029-4>
- [21] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia tools and applications*, 80(5), 8091-8126. <https://doi.org/10.1007/s11042-020-10139-6>
- [22] Mosayebi, M., & Sodhi, M. (2020, July). Tuning genetic algorithm parameters using design of experiments. In *Proceedings of the 2020 genetic and evolutionary computation conference companion* (pp. 1937-1944). <https://doi.org/10.1145/3377929.3398136>
- [23] Mills, K. L., Filliben, J. J., & Haines, A. L. (2015). Determining relative importance and effective settings for genetic algorithm control parameters. *Evolutionary computation*, 23(2), 309-342. https://doi.org/10.1162/EVCO_a_00137
- [24] Lin, W. Y., Lee, W. Y., & Hong, T. P. (2003). Adapting crossover and mutation rates in genetic algorithms. *Journal of information science and engineering*, 19(5), 889-903. <https://doi.org/10.1688/JISE.2003.19.5.10>
- [25] Hassanat, A., Almohammadi, K., Alkafaween, E. A., Abunawas, E., Hammouri, A., & Prasath, V. S. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*, 10(12), 390. <https://doi.org/10.3390/info10120390>
- [26] Altıparmak, F., Gen, M., Dengiz, B., & Smith, A. E. (2004). A genetic algorithm with fuzzy logic controller for design of communication networks. *IEEE transactions on electronics, information and systems*, 124(10), 1979-1985. <https://doi.org/10.1541/ieejieiss.124.1979>
- [27] Vannucci, M., Colla, V., & Dettori, S. (2016). Fuzzy adaptive genetic algorithm for improving the solution of industrial optimization problems. *IFAC-PapersOnLine*, 49(12), 1128-1133. <https://doi.org/10.1016/j.ifacol.2016.07.650>
- [28] Pytel, K. (2025). Fuzzy logic applied to tuning mutation size in evolutionary algorithms. *Scientific reports*, 15(1), 1937. <https://doi.org/10.1038/s41598-025-86349-5>
- [29] Tohyama, H., & Tomisawa, M. (2022). Complexity of the police officer patrol problem. *Journal of information processing*, 30, 307-314. <https://doi.org/10.2197/ipsjip.30.307>

-
- [30] Tomisawa, M., & Tohyama, H. (2024). Computational complexity of the police officer patrol problem on weighted digraphs. *Electronic Journal of graph theory & applications*, 12(2). 297–313. <https://doi.org/10.5614/ejgta.2024.12.2.10>
 - [31] Kudo, F., Tohyama, H., & Tomisawa, M. (2025). Hybrid heuristic approach for generalized police officer patrolling problem. *Frontiers in industrial engineering*, 3, 1620422. <https://doi.org/10.3389/fieng.2025.1620422>
 - [32] Zhang, W., Gen, M., & Jo, J. (2014). Hybrid sampling strategy-based multiobjective evolutionary algorithm for process planning and scheduling problem. *Journal of intelligent manufacturing*, 25(5), 881-897. <https://doi.org/10.1007/s10845-013-0814-2>
 - [33] Schaffer, J. D. (2014, January). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the first international conference on genetic algorithms and their applications* (pp. 93-100). Psychology Press. <https://www.taylorfrancis.com/chapters/edit/10.4324/9781315799674-9/multiple-objective-optimization-vector-evaluated-genetic-algorithms-david-schaffer>
 - [34] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197. <https://doi.org/10.1109/4235.996017>
 - [35] Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4), 257-271. <https://doi.org/10.1109/4235.797969>
 - [36] Coello Coello, C. A., & Reyes Sierra, M. (2004, April). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican international conference on artificial intelligence* (pp. 688-697). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24694-7_71