MALAYSIAN JOURNAL OF SCIENCE
MJS
ISSN 1394 - 3065

5th International Conference on Applied & Industrial Mathematics and Statistics 2025 (ICoAIMS 2025)

Special issue

Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

https://mjs.um.edu.my

Determining the Mixed Metric Dimension of a Graph with the Binary Gray Wolf Algorithm

Hendy[1ab], Mohammad Isa Irawan[2a*], Imam Mukhlash[3a], and Rinurwati[4a]

Abstract: Many computational techniques for extracting insights from data depend on embedding data within a Euclidean space. However, symbolic datasets, particularly biological sequence datasets resulting from high-throughput sequencing processes, often pose challenges for conventional embedding methods, as they tend to yield either excessively high-dimensional or overly simplified representations, thereby limiting their analytical applicability. To address these constraints, the concept of metric dimension (MD) in graph theory provides a rigorous and practical framework for representing points in the Euclidean plane based on their relative distances. In this study, we explore a hybrid concept that combines two distance parameters: MD and edge metric dimension (EMD). The mixed metric dimension (MMD) uniquely characterizes a graph's elements (edges and vertices) based on their distance vectors to a designated set of vertices. We developed a computational intelligence approach to determine the mixed metric dimension of graphs: the Binary Gray Wolf Optimizer (BGWO) algorithm. We integrate a distance-based matrix representation with population-based optimization methods. To improve solution performance, several modifications have been incorporated, including the validation of row uniqueness, the implementation of dynamic stagnation handling techniques, and the adoption of an elitism strategy. Our findings indicate that the BGWO achieves efficient completion times and delivers competitive performance for this problem.

Keywords: Binary gray wolf optimizer algorithm, edge metric dimension (EMD), mixed metric dimension (MMD)

1. Introduction

Unless otherwise stated, all graphs in this study are connected and simple. The distance between two vertices u and v in a graph G is represented as $d_G(u,v)$. A vertex $s \in V(G)$ is said to differentiate (or resolve) two vertices u and v if $d_G(u,s) \neq d_G(v,s)$. A subset $S \subseteq V(G)$, is referred to as a metric generator for G if each pair of vertices u and v in G is differentiated by at least one vertex in S. The smallest cardinality of such a subset is called the MD of G, denoted by $\dim(G)$. The concept originated independently through the contributions of Harrary and Melter (1976) and Slater (1975), who referred to it as resolving sets and locating sets, respectively.

The MD of graphs has a wide range of applications, including mathematical biology (Tillquist & Lladser, 2019). Collections of symbolic representations, especially sequences of biological macromolecules generated by advanced genomic sequencing technologies, often present challenges for conventional embedding methods. These techniques often result in excessively high-dimensional or overly simplistic representations, thereby reducing their usefulness for analytical purposes. To overcome these challenges, the graph-theoretical notion of the MD provides a rigorous and efficient framework for representing data points within the Euclidean plane. Furthermore, the concept of metric dimension is also used in various applications beyond data representation, including network topology analysis (Beerliova et al., 2006), sensor placement optimization (Chartrand et al., 2000), structural graph characterization and pharmaceutical chemistry (Imran et al., 2022), methods for processing patterns and images (Melter & Tomescu, 1984), autonomous robot movement (Khuller et al., 1996), and combinatorial optimization (Seb & Tannier, 2016).

1.1. Edge metric dimension

The metric dimension concept has recently evolved from focusing on vertex identification to edge identification in a graph (Kelenc, Tratnik, & Yero, 2018). For a graph $H = (V, E)$, a vertex $v \in V$ and an edge $e = uw \in E$, the distance between the vertex v and the edge e is defined as $d_G(e,v) = \min\{d_G(u,v), d_G(w,v)\}$. A vertex $x \in V$ distinguishes two edges $e_1, e_2 \in E$ if $d_G(x,e_1) \neq d_G(x,e_2)$. A set S of vertices is an edge metric generator if every pair of distinct edges in G is differentiated through at least one vertex in S. The minimum cardinality of an edge metric generator in G is known as the edge metric dimension, denoted by $dim_e(G)$. An edge metric basis of G is an edge metric generator with a cardinality of exactly $dim_e(G)$. For instance, consider a graph G, as illustrated in Figure 1. $Dim(H) = 2$ with the resolving set $S_1 = \{a_1, a_2\}$. The representations of all vertices: $r(a_1|S_1) = (0,1)$, $r(a_2|S_1) = (1,0)$, $r(a_3|S_1) = (2,1)$, $r(a_4|S_1) = (2,2)$, $r(a_5|S_1) = (1,2)$ are all different. $Dim_e(H) = 3$ with the edge metric generator of G is $S_2 = \{a_1, a_4, a_5\}$. The depictions of all edges: $r(a_1a_2|S_2) = (0,1,1)$, $r(a_1a_5|S_2) = (0,1,0)$, $r(a_2a_3|S_2) = (1,1,1)$, $r(a_2a_4|S_2) = (1,0,1)$, $r(a_2a_5|S_2) = (1,1,0)$, $r(a_3a_4|S_2) = (2,0,1)$, $r(a_4a_5|S_2) = (1,0,0)$ are all different.

Authors information:
[a]Dept. of Mathematics, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. E-mail: 7002211006@student.its.ac.id[1], mii@its.ac.id[2], imamm@matematika.its.ac.id[3], rinur@matematika.its.ac.id[4].
[b]Faculty of Engineering, Universitas Kadiri, Kediri, Indonesia. E-mail: 7002211006@student.its.ac.id[1]

*Corresponding Author: mii@its.ac.id

Received: October 28, 2025
Accepted: May 6, 2026
Published: November 15, 2026

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

Numerous studies were conducted on the EMD of various graph families and the outcomes of specific graph operations (Peterin & Yero, 2020; Zhang & Gao, 2020). Furthermore, research was also performed on the characterization of graphs with particular EMD (Zhu et al., 2019; Knor et al., 2021).

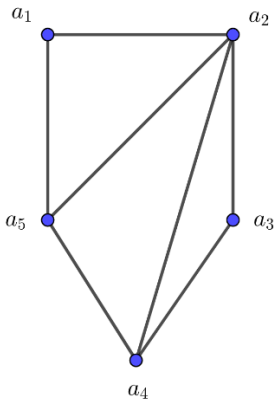


Figure 1. Graph H

1.2. Mixed metric dimension

This study focuses on a combined variation of the two previously mentioned parameters. Specifically, given a graph G, our objective is to distinguish elements of G (edges and vertices) through distance vectors to a predetermined set of vertices in G. A vertex a in a graph G is said to differentiate two elements u and v (which may be vertices or edges) if $d_G(u,a) \neq d_G(v,a)$. A set $S \subseteq G$ is defined as a mixed metric generator if any two distinct elements of G are differentiated by at least one vertex in S. The smallest size of such a set is the mixed metric dimension (MMD) of G, represented by $dim_m(G)$ (Kelenc et al. 2017).

For example, consider the graph H illustrated in Figure 1. It can be demonstrated that there is no mixed metric generator of H with cardinality < 4. The graph H has $dim_m(H) = 4$, where $S_3 = \{a_1, a_3, a_4, a_5\}$ as mixed metric generator. The representations of all elements of H: $r(a_1|S_3) = (0,2,2,1)$, $r(a_2|S_3) = (1,1,1,1)$, $r(a_3|S_3) = (2,0,1,2)$, $r(a_4|S_3) = (2,1,0,1)$, $r(a_5|S_3) = (1,2,1,0)$, $r(a_1a_2|S_3) = (0,1,1,1)$, $r(a_1a_5|S_3) = (0,2,1,0)$, $r(a_2a_3|S_3) = (1,0,1,1)$, $r(a_2a_4|S_3) = (1,1,0,1)$, $r(a_2a_5|S_3) = (1,1,1,0)$, $r(a_3a_4|S_3) = (2,0,0,1)$, $r(a_4a_5|S_3) = (1,1,0,0)$ are all different.

2. Literature review

The MMD possess several combinatorial properties. Every mixed-metric generator functions simultaneously as a metric generator and an edge metric generator. Consequently, the following relationship holds.

Remark 1. (Kelenc et al. 2017). For any graph G, $dim_m(G) \geq \max\{\dim(G), dim_e(G)\}$

Remark 2. (Kelenc et al. 2017). For any graph G of order n, $2 \leq dim_m(G) \leq n$

Several studies have investigated the MMD in different graph families and examined the effects of various graph operations (Koam et al., 2023; Danas, 2021; Abbas et al., 2024; Darmaji & Azahra, 2022; Sedlar & Škrekovski, 2021b). Furthermore, research was conducted on the characterization of graphs with a particular mixed-metric dimension (Sedlar & Škrekovski, 2021a).

Computing the MD of a graph can be modeled as an optimization problem. From a computational perspective, Diaz et al. (2017) and Hauptmann et al. (2012) demonstrated the complexity of the MDP, showing that it is NP-complete. Computational methods for the metric dimension problem have previously been explored by Kratica et al. (2009) on determining the strong MD of various graph classes using genetic algorithms. Kratica et al. (2012) formulated the problem of a strong MD as a linear integer programming model. Chen et al. (2016) designed a strongly polynomial-time exact algorithm to solve the minimum cost doubly resolving set problem for specific graph classes, while Hoffmann et al. (2016) implemented a linear-time algorithm for determining the MD of certain graph classes.

Further advancements occurred when Murdiansyah and Adiwijaya (2017) applied the particle swarm optimization algorithm in the same year to solve the MDP for hypercube graphs. Vietz et al. (2019) proposed a new approach by decomposing graphs and applying an algorithm to determine their metric dimension. Muhammad and Susilowati (2020) developed an algorithm to compute the metric dimension of a graph. Chelladurai (2020) designed an algorithm to determine the MD and basis of simple graphs. Mohamed et al. (2023) proposed an algorithm to address the dominant metric dimension problem by employing the binary Archimedes optimization algorithm.

This paper introduces a computational approach to a problem that extends the concept of metric dimension, known as the MMD. To date, no computational method has been developed to solve the MMDP. The approach adopted in this paper utilizes the binary gray wolf optimizer, a population-based metaheuristic algorithm inspired by the hunting behavior of gray wolves. Research findings show that the GWO is one of the most widely used algorithms for solving various optimization problems. A bibliometric analysis indicates that the gray wolf optimizer is the most frequently referenced metaheuristic in computer science (Hendy et al., 2023).

This paper adopts the binary variant of the Gray Wolf Optimizer, an optimization algorithm initially introduced by Mirjalili et al. (2014). This algorithm falls under the category of population-based swarm intelligence techniques. It is designed based on the hunting behavior and hierarchical structure of gray wolf (Canis lupus) packs. Gray wolves, members of the Canidae family, are classified as apex predators, indicating their dominance at the top of the food chain. These animals typically live in structured packs, with an average pack size of 5 to 12 members.

The pack's leader is called the alpha, who may be either a male or female wolf and holds the highest authority within the gray wolf hierarchy. The alpha is responsible for making key decisions. The next hierarchical level is the beta, a subordinate to the alpha who assists in decision making and performing

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

associated activities. At the lowest level is the omega, which acts as the group's scapegoat and is submissive to all dominant wolves. Wolves that do not belong to the alpha, beta, or omega groups but dominate the omega are classified as delta wolves. Although delta wolves are outranked by alpha and beta wolves, they exert authority over the omega.

The following principles are established in developing a mathematical model based on the social hierarchy of gray wolves: (1) The optimal solution, representing the optimal fitness value, is designated as α. (2) The next best solution is represented by β, while the third-highest ranked solution is denoted as δ. (3) All remaining wolves not belonging to these categories are classified as omega. (4) Within the GWO algorithm, the optimization process, analogous to the hunting process, is directed by the α, β, and δ wolves. In contrast, the ω wolves follow the guidance of these leading wolves.

A pack of gray wolves surrounds their prey during hunting. In mathematical modeling, this process is expressed by the following equation.

$$\vec{D} = \left|\vec{K}.\overrightarrow{Y_p}(i) - \vec{Y}(i)\right| \quad (1)$$

$$\vec{Y}(i+1) = \overrightarrow{Y_p}(t) - \vec{A}.\vec{D} \quad (2)$$

In this context, i represents the current iteration, $\vec{A}$ and $\vec{K}$ are coefficient vectors. $\overrightarrow{Y_p}$ denotes the coordinate vector representing the prey's position, and $\vec{Y}$ denotes the coordinate vector of the gray wolf. Vectors $\vec{A}$ and $\vec{K}$ are determined based on the following formulas:

$$\vec{A} = 2\vec{a}.\overrightarrow{n_1} - \vec{a} \quad (3)$$

$$\vec{K} = 2\,\overrightarrow{n_2} \quad (4)$$

Here, $\vec{a}$ linearly decays from an initial value of 2 to 0 across all iterations, where $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$ represent randomly generated vectors within the range of [0,1].

In mathematically simulating gray wolf hunting behavior, the α (representing the most optimal potential solution), β, and δ possess better information regarding the prospective location of the prey. Therefore, the top three optimal solutions obtained are retained, while the remaining search wolves, including the omega members, update their positions with reference to where the leading wolves are located. This hunting process is modeled through equations (5) to (11) (Mirjalili et al., 2014).

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{K_1}.\overrightarrow{Y_\alpha} - \vec{Y}\right| \quad (5)$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{K_2}.\overrightarrow{Y_\beta} - \vec{Y}\right| \quad (6)$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{K_3}.\overrightarrow{Y_\delta} - \vec{Y}\right| \quad (7)$$

$$\overrightarrow{Y_1} = \overrightarrow{Y_\alpha} - \overrightarrow{A_1}.\left(\overrightarrow{D_\alpha}\right) \quad (8)$$

$$\overrightarrow{Y_2} = \overrightarrow{Y_\beta} - \overrightarrow{A_2}.\left(\overrightarrow{D_\beta}\right) \quad (9)$$

$$\overrightarrow{Y_3} = \vec{Y} - \overrightarrow{A_3}.\left(\overrightarrow{D_\delta}\right) \quad (10)$$

$$\overrightarrow{Y_{(\iota+1)}} = \frac{\overrightarrow{Y_1}+\overrightarrow{Y_2}+\overrightarrow{Y_3}}{3} \quad (11)$$

3. Proposed BGWO algorithm for the mixed metric dimension problem

Modifications and refinements of the BGWO algorithm are crucial for its implementation in solving the MMDP. This necessity arises from the inherent characteristics of the solutions to the mixed metric dimension problem, which are represented as binary ordered, with not all randomly generated outputs producing feasible candidate solutions. This section offers a detailed explanation of the BGWO algorithm, adapted from its application to the feature selection problem, as described in prior work (Emary et al., 2016). This section also discusses various modification processes designed to enhance the algorithm's performance in solving the mixed metric dimension problem. The primary updating equation is presented as shown in equation (12). Refer to Algorithm 1 for further details.

$$X_i^{t+1} = Crossover(x_1, x_2, x_3) \quad (12)$$

where $Crossover(x_1, x_2, x_3)$ signifies an appropriate crossover operation involving solutions x, y, z, and x_1, x_2, x_3 are binary vectors denoting the impact of a wolf's movement towards the α, β, and δ gray wolves sequentially. The x_1, x_2, x_3 are determined using equations (13),(16), and (19) correspondingly.

$$x_1^d = \begin{cases} 1, \text{if } (x_\alpha^d + bstep_\alpha^d) \geq 1 \\ 0, \quad \text{otherwise} \end{cases} \quad (13)$$

x_1^d represents the position vector of the alpha wolf in dimension d, while $bstep_\alpha^d$ signifies a binary step in dimension d, which can be computed as described in equation (14).

$$bstep_\alpha^d = \begin{cases} 1, \text{if } cstep_\alpha^d \geq rand \\ 0, \quad \text{otherwise} \end{cases} \quad (14)$$

$rand$ is a randomly generated number from a uniform distribution within the [0, 1] range. $cstep_\alpha^d$ reflects the continuous-valued step size for dimension d and can be computed using a sigmoidal function as outlined in equation (15).

$$cstep_\alpha^d = \frac{1}{1+e^{-10(A_1^d D_\alpha^d - 0{,}5)}} \quad (15)$$

where A_1^d and D_α^d are determined by equations (3), and (9) in the dimension d.

$$x_2^d = \begin{cases} 1, \text{if } (x_\beta^d + bstep_\beta^d) \geq 1 \\ 0, \quad \text{otherwise} \end{cases} \quad (16)$$

The vector x_β^d represents the position vector of the beta wolf in dimension d, and $bstep_\beta^d$ represents a binary step in dimension d that can be determined as in equation (17).

$$bstep_\beta^d = \begin{cases} 1, \text{if } cstep_\beta^d \geq rand \\ 0, \quad \text{otherwise} \end{cases} \quad (17)$$

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

where the $rand$ denotes a randomly generated number from a uniform distribution within the range ∈ [0, 1], and $cstep_\beta^d$ represents the continuous-valued step size for dimension d which can be computed using a sigmoidal function as outlined in equation (18).

$$cstep_\beta^d = \frac{1}{1+e^{-10(A_1^d D_\beta^d - 0{,}5)}} \quad (18)$$

where A_1^d and D_β^d are computed using equations (3) and (5) in the dimension d.

$$x_3^d = \begin{cases} 1, \text{if } (x_\delta^d + bstep_\delta^d) \geq 1 \\ 0, \quad \text{otherwise} \end{cases} \quad (19)$$

x_δ^d represents the position vector of the delta wolf in dimension d, while $bstep_\delta^d$ denotes a binary step in dimension d, calculated according to equation (20).

$$bstep_\delta^d = \begin{cases} 1, \text{if } cstep_\delta^d \geq rand \\ 0, \quad \text{otherwise} \end{cases} \quad (20)$$

where $rand$ is a random number drawn from the uniform distribution ∈ [0, 1], and $cstep_\delta^d$ is the continuous-valued step size for dimension d and can be calculated using a sigmoidal function as in equation (21).

$$cstep_\delta^d = \frac{1}{1+e^{-10(A_1^d D_\delta^d - 0{,}5)}} \quad (21)$$

As equation (22) demonstrates, a stochastic crossover strategy is implemented per dimension to perform the crossover operation among the solutions a, b, and c.

$$x_d = \begin{cases} a_d, & \text{if } rand < \frac{1}{3} \\ b_d, & \text{if } \frac{1}{3} \leq rand < \frac{2}{3} \\ c_d, & \text{otherwise} \end{cases} \quad (22)$$

where a_d, b_d, c_d represent the binary values for the first, second, and third parameters in dimension d, x_d denotes the crossover output at dimension d, and rand is a randomly generated number from a uniform distribution within the range [0,1].

Algorithm 1. Modified BGWO Algorithm for mixed metric dimension problem

Input: n number of vertices, Matrix C, popsize, MaxIt
Output: x_α optimal gray wolf, mixed metric dimension

1. Initalize a population of a popsize wolves' positions at random ∈ $[0,1]$
2. Find the representation of all vertices and edges with respect to wolves
3. Row uniqueness validation
4. Replace the population based on objective value
5. Find the fitness value
6. Find the α, β, and δ based on fitness while stopping criteria not met do

For each $wolf_i \epsilon\ pack$ do
Calculate x_1, x_2, x_3 using equations (13),(16), and (19)
$x_i^{t+1} \leftarrow$ crossover among x_1, x_2, x_3 using equation (22)
end (for)

I. Update a, A, C
II. Evaluate the positions of individual wolves
III. Update α, β, and δ
IV. Mutation
V. Validation
VI. Stagnation handling and elitism end (while)

The proposed method combines distance-based matrix representations and population-based optimization techniques modeled on the behavior of gray wolves in nature. This section describes the construction of the problem, initialization, iterative process, and enhancements incorporated into the BGWO approach.

I. Graph Representation and Matrix Construction The input graph is represented by an adjacency matrix A, which stores the weights between nodes. The diagonal entries of A are set to zero, indicating that the distance between any node and itself is zero. The distance is set to infinity (inf) if no direct edge exists between two nodes. To evaluate the distances between nodes and edges, the following matrices are constructed:

1. Distance Matrix D, contains the minimum distances along the shortest paths connecting every pair of vertices, computed using the Floyd-Warshall algorithm.
2. Edge-to-Node Distance Matrix B: For each edge in the graph, the minimum distance to all nodes is calculated and stored in Matrix B.
3. Combined Matrix C: The Matrix C merges D and B to provide a complete distance representation of both nodes and edges.

The Matrix C serves as the primary input to the optimization algorithm.

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

II. Binary Gray Wolf Optimization Framework

The steps of the BGWO algorithm are outlined as follows:

Step 1: Population Initialization

A population of popsize binary wolves is arbitrarily generated. Each wolf represents a candidate solution, where a one at position d indicates that vertex d is selected for the solution. If no valid solution is found in the initial population (i.e., all nodes are set to 0), the algorithm regenerates new wolves to ensure at least one vertex is selected.

Step 2: Fitness Evaluation

For each wolf in the population, the following steps are performed:

1. Matrix Representation: The binary vector of a wolf selects specific columns from Matrix C, generating a submatrix.
2. Lexicographic Sorting: The submatrix is sorted lexicographically to facilitate comparison between rows.
3. Row Uniqueness Validation: If any two rows in the sorted Matrix are identical, the wolf is deemed invalid. The wolf is updated by randomly flipping one 0 to a 1, and the validation is repeated until all rows are a unique representation of both vertices and edges.
4. Objective Value Calculation: The algorithm computes the maximum positional difference (j(i)) between consecutive rows of the sorted matrix. This value measures the quality of the solution.
5. Fitness Assignment: The fitness value is defined as 1/(1 + j(i)), where a lower j(i) yields a higher fitness score. This formulation encourages solutions with minimal vertex selections while maintaining row uniqueness.

Step 3: Selection of α, β, and δ Wolves

Based on the fitness values, the wolves are ranked, and the three best wolves are identified as α, β, and δ. These wolves guide the population's exploration and exploitation during the optimization process.

Step 4: Iterative Optimization

The algorithm proceeds iteratively for a maximum number of iterations, applying the following operations at each iteration:

1. Wolf Update: Each wolf updates its position based on the influence of alpha, beta, and delta. The update involves generating random coefficients (A and C), calculating new candidate positions (X1, X2, X3), and performing crossover to determine the final position.
2. Mutation: To maintain population diversity and avoid premature convergence, a mutation step randomly flips bits in the wolf vector with a certain probability.
3. Validation: After updating positions, the wolves undergo validation to ensure that all matrix representations maintain unique rows. Wolves failing this validation are repaired by modifying their binary vectors.
4. Objective Reevaluation: The algorithm recomputes the updated population's objective values and fitness scores.
5. Update α, β, and δ wolves: The α, β, and δ wolves are updated based on the new fitness rankings.

Step 5: Stagnation Handling

The algorithm tracks changes in the best objective value across iterations to address stagnation. If no improvement is observed over multiple iterations, the algorithm performs a partial restart by reinitializing a portion of the population. This strategy helps escape local optima and enhances the solution space exploration.

III. Termination and Result Visualization

The algorithm terminates when it reaches either the maximum iteration count or a convergence criterion. The best solution (alpha) is then extracted, representing the optimal selection of vertices for the MMDP. The script stores the MMD at each iteration. A plot is generated to visualize the progression of the metric dimension over time, demonstrating the algorithm's performance and how it approaches convergence.

4. Results

In this section, the experiments were conducted using MATLAB R2023b, installed on a Windows 10 Pro system equipped with a Core i5 processor and 16 GB of RAM. The results of the BGWO algorithm for the MMD of the cycle graphs, wheel graphs, path graphs, grid graphs, and complete bipartite graphs are presented in Tables 1, 2, 3, 4, and 5, respectively. Also, the upper bounds for the EMD and MMD of the generalized antiprism graph are provided. These theoretical results can then be compared with the outcomes obtained from the BGWO algorithm simulations.

An elitism process is required for path graphs P_n with $n \geq 12$, for ladder graphs L_n with $n \geq 9$, and for grid graphs $P_4 \times P_5$. Elitism ensures that the best solution found so far is never lost during iterations. After evaluating all individuals in the population at each iteration, the algorithm identifies the best-performing solution (alpha) based on the objective function. If this new best solution is better than the previously recorded best solution, it updates the record. Otherwise, the previous best solution is reinserted into the population, replacing the current worst individual. This process guarantees that the best solution persists through generations, preventing degradation due to random mutations or exploration steps. Consequently, elitism helps stabilize the algorithm, reduces fluctuations, and ensures convergence towards an optimal solution over time (see Fig. 2). This simulation shows that the BGWO algorithm produces the same results as those reported in (Kelenc, 2016; Sedlar & Škrekovski, 2021a; Sedlar & Škrekovski, 2021b).

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

Table 1. Results of BGWO Algorithm for MMD on Cycle graphs.

Graph	n	m	popsize	It	dim_m	Mixed metric generator
Cycle graphs	4	4	20	1	3	$\{a_1, a_3, a_4\}$
	5	5	20	1	3	$\{a_1, a, a_4\}$
	6	6	20	1	3	$\{a_1, a_3, a_4\}$
	7	7	20	1	3	$\{a_1, a_4, a_6\}$
	8	8	20	1	3	$\{a_2, a_4, a_8\}$
	9	9	20	1	3	$\{a_1, a_5, a_9\}$
	10	10	20	1	3	$\{a_3, a_5, a_8\}$
	11	11	20	1	3	$\{a_3, a_5, a_9\}$
	12	12	20	1	3	$\{a_2, a_4, a_8\}$
	13	13	20	1	3	$\{a_2, a_8, a_{11}\}$
	14	14	20	1	3	$\{a_6, a_{13}, a_{14}\}$
	15	15	20	1	3	$\{a_5, a_{11}, a_{15}\}$

Table 2. Results of BGWO Algorithm for MMD on Wheel graphs.

Graph	n	m	popsize	It	dim_m	Mixed metric generator
Wheel graph	4	6	20	1	4	$\{w_1, w_2, w_3, w_4\}$
	5	8	20	1	4	$\{w_2, w_3, w_4, w_5\}$
	6	10	20	1	5	$\{w_2, w_3, w_4, w_5, w_6\}$
	7	12	20	1	6	$\{w_2, w_3, w_4, w_5, w_6, w_7\}$
	8	14	20	1	7	$\{w_2, w_3, w_4, w_5, w_6, w_7, w_8\}$
	9	16	20	1	8	$\{w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\}$
	10	18	20	1	9	$\{w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}\}$

Table 3. Results of BGWO Algorithm for MMD on path graphs.

Graph	n	m	popsize	It	dim_m	Mixed metric generator
Path graph	2	1	20	1	2	$\{v_1, v_2\}$
	3	2	20	1	2	$\{v_1, v_3\}$
	4	3	20	1	2	$\{v_1, v_4\}$
	5	4	20	1	2	$\{v_1, v_5\}$
	6	5	20	1	2	$\{v_1, v_6\}$
	7	6	20	1	2	$\{v_1, v_7\}$
	8	7	20	1	2	$\{v_1, v_8\}$
	9	8	20	1	2	$\{v_1, v_{10}\}$
	10	9	20	1	2	$\{v_1, v_{10}\}$
	11	10	20	1	2	$\{v_1, v_{11}\}$
	12	11	20	1	2	$\{v_1, v_{12}\}$
	13	12	20	1	2	$\{v_1, v_{13}\}$
	14	13	20	1	2	$\{v_1, v_{14}\}$
	15	14	20	1	2	$\{v_1, v_{15}\}$

Table 4. Results of BGWO Algorithm for MMD on Ladder and Grid graphs.

Graph	n	m	popsize	It	dim_m	Mixed metric generator
L_3	6	7	20	1	3	$\{l_1, l_3, l_6\}$
L_4	8	10	20	1	3	$\{l_3, l_5, l_8\}$
L_5	10	13	20	1	3	$\{l_1, l_5, l_8\}$
L_6	12	16	20	1	3	$\{l_2, l_7, l_{12}\}$
L_7	14	19	20	1	3	$\{l_2, l_8, l_{14}\}$
L_8	16	22	20	1	3	$\{l_1, l_8, l_{13}\}$
L_9	18	25	20	3	3	$\{l_5, l_{10}, l_{18}\}$
L_{10}	20	28	20	2	3	$\{l_1, l_{10}, l_{14}\}$
$P_3 \times P_3$	9	12	20	1	3	$\{p_1, p_3, p_7\}$
$P_3 \times P_4$	12	17	20	1	3	$\{p_1, p_3, p_{10}\}$
$P_4 \times P_5$	20	31	20	3	3	$\{p_1, p_4, p_{19}\}$
			20			

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

Table 5. Results of BGWO Algorithm for MMD on Complete Bipartite graphs.

Graph	n	m	popsize	It	dim_m	Mixed metric generator
L_3	6	7	20	1	3	$\{l_1, l_3, l_6\}$
L_4	8	10	20	1	3	$\{l_3, l_5, l_8\}$
L_5	10	13	20	1	3	$\{l_1, l_5, l_8\}$
L_6	12	16	20	1	3	$\{l_2, l_7, l_{12}\}$
L_7	14	19	20	1	3	$\{l_2, l_8, l_{14}\}$
L_8	16	22	20	1	3	$\{l_1, l_8, l_{13}\}$
L_9	18	25	20	3	3	$\{l_5, l_{10}, l_{18}\}$
L_{10}	20	28	20	2	3	$\{l_1, l_{10}, l_{14}\}$
$P_3 \times P_3$	9	12	20	1	3	$\{p_1, p_3, p_7\}$
$P_3 \times P_4$	12	17	20	1	3	$\{p_1, p_3, p_{10}\}$
$P_4 \times P_5$	20	31	20	3	3	$\{p_1, p_4, p_{19}\}$
			20			

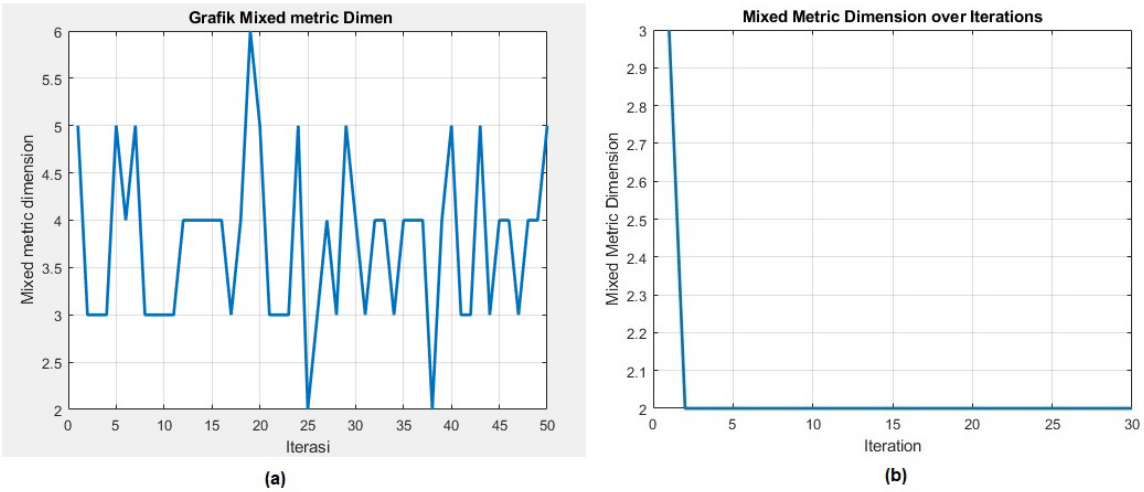


Figure 2. The MMD of P_{12} without (a) and with (b) elitism process

Now, we give the upper bound on the EMD and the upper bound on the MMD of the generalized antiprism graph.

Theorem 1. Edge metric dimension of the generalized antiprism graph A_3^n, $dim(A_3^n) \leq 2n$.

Proof. Suppose $V(A_3^n) = \{a_1^1, a_1^2, a_1^3, a_2^1, a_2^2, a_2^3, a_3^1, a_3^2, a_3^3, \dots, a_n^1, a_n^2, a_n^3\}$ and $S^n = \{a_1^1, a_1^2, a_2^2, a_2^3, a_3^1, a_3^3, \dots, a_n^x, a_n^y\} \subseteq V(A_3^n)$ with $(x, y) = \begin{cases} (1,2) \text{ if } n = 3t-2, \ t \in \{1,2,3,\dots\} \\ (2,3) \text{if } n = 3t-1, \ t \in \{1,2,3,\dots\} \text{ is the set of vertices located} \\ (1,3) \quad \text{otherwise} \end{cases}$
on paths that are opposite to each other, as illustrated in Figure 3. The representation of all edges in A_3^n is as follows:

$$r(a_1^1 a_1^2 | S^n) = (0,0,1,1,2,2, \dots, n-1, n-1)$$
$$r(a_1^2 a_1^3 | S^n) = (1,0,1,1,2,2, \dots, n-1, n-1)$$
$$r(a_1^1 a_1^3 | S^n) = (0,1,1,1,2,2, \dots, n-1, n-1)$$
$$r(a_2^1 a_2^2 | S^n) = (1,1,0,1,1,1, \dots, n-2, n-2)$$
$$r(a_2^2 a_2^3 | S^n) = (1,1,0,0,1,1, \dots, n-2, n-2)$$
$$r(a_2^1 a_2^3 | S^n) = (1,1,1,0,1,1, \dots, n-2, n-2)$$
$$r(a_3^1 a_3^2 | S^n) = (2,2,1,1,0,1, \dots, n-3, n-3)$$
$$r(a_3^2 a_3^3 | S^n) = (2,2,1,1,1,0, \dots, n-3, n-3)$$
$$r(a_3^1 a_3^3 | S^n) = (2,2,1,1,0,0, \dots, n-3, n-3)$$
$$\vdots$$
$$r(a_n^1 a_n^2 | S^n) = (n-1, n-1, n-2, n-2, n-3, n-3, \dots, x_1, y_1)$$
$$r(a_n^2 a_n^3 | S^n) = (n-1, n-1, n-2, n-2, n-3, n-3, \dots, x_2, y_2)$$
$$r(a_n^1 a_n^3 | S^n) = (n-1, n-1, n-2, n-2, n-3, n-3, \dots, x_3, y_3)$$

where:

$$(x_1, y_1) = \begin{cases} (0,0) \text{ if } n = 3t-2, \ t \in N \\ (0,1) \quad \text{otherwise,} \end{cases} \quad (x_2, y_2) = \begin{cases} (1,0) \text{ if } n = 3t-2 \text{ or } n = 3t, \ t \in N \\ (0,0) \quad \text{otherwise} \end{cases}$$

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

$$(x_3, y_3) = \begin{cases} (0,1) \text{ if } n = 3t-2 \text{ or } n = 3t, t \in N \\ (1,0) \text{ if } n = 3t-1, t \in N \\ (0,0) \text{ otherwise} \end{cases}$$

Next:

$$r(a_1^1 a_2^1 | S^n) = (0,1,1,1,1,1,2,2,3,3, \dots, n-2, n-2)$$
$$r(a_1^2 a_2^1 | S^n) = (1,0,1,1,1,1,2,2,3,3, \dots, n-2, n-2)$$
$$r(a_1^2 a_2^2 | S^n) = (1,0,0,1,1,2,2,2,3,3, \dots, n-2, n-2)$$
$$r(a_1^3 a_2^2 | S^n) = (1,1,0,1,1,2,2,2,3,3, \dots, n-2, n-2)$$
$$r(a_1^3 a_2^3 | S^n) = (1,1,1,0,2,1,2,2,3,3, \dots, n-2, n-2)$$
$$r(a_1^1 a_2^3 | S^n) = (0,1,1,0,2,1,2,2,3,3, \dots, n-2, n-2)$$
$$r(a_2^1 a_3^1 | S^n) = (1,1,1,1,0,1,1,2,2,2, \dots, n-3, n-3)$$
$$r(a_2^2 a_3^1 | S^n) = (2,1,0,1,0,1,1,2,2,2, \dots, n-3, n-3)$$
$$r(a_2^2 a_3^2 | S^n) = (2,1,0,1,1,1,1,1,2,2, \dots, n-3, n-3)$$
$$r(a_2^3 a_3^2 | S^n) = (1,2,1,0,1,1,1,1,2,2, \dots, n-3, n-3)$$
$$r(a_2^3 a_3^3 | S^n) = (1,2,1,0,1,0,2,1,2,2, \dots, n-3, n-3)$$
$$r(a_2^1 a_3^3 | S^n) = (1,1,1,1,1,0,2,1,2,2, \dots, n-3, n-3)$$
$$r(a_3^1 a_4^1 | S^n) = (2,2,1,2,0,1,0,1,2,1, \dots, n-4, n-4)$$
$$r(a_3^2 a_4^1 | S^n) = (2,2,1,1,1,1,0,1,2,1, \dots, n-4, n-4)$$
$$r(a_3^2 a_4^2 | S^n) = (2,2,1,1,1,1,1,0,1,2, \dots, n-4, n-4)$$
$$r(a_3^3 a_4^2 | S^n) = (2,2,2,1,1,0,1,0,1,2, \dots, n-4, n-4)$$
$$r(a_3^3 a_4^3 | S^n) = (2,2,2,1,1,0,1,1,1,1, \dots, n-4, n-4)$$
$$r(a_3^1 a_4^3 | S^n) = (2,2,1,2,0,1,1,1,1,1, \dots, n-4, n-4)$$
$$r(a_4^1 a_5^1 | S^n) = (3,3,2,2,1,2,0,1,1,1, \dots, n-5, n-5)$$
$$r(a_4^2 a_5^1 | S^n) = (3,3,2,2,2,1,1,0,1,1, \dots, n-5, n-5)$$
$$r(a_4^2 a_5^2 | S^n) = (3,3,2,2,2,1,1,0,0,1, \dots, n-5, n-5)$$
$$r(a_4^3 a_5^2 | S^n) = (3,3,2,2,1,1,1,1,0,1, \dots, n-5, n-5)$$
$$r(a_4^3 a_5^3 | S^n) = (3,3,2,2,1,1,1,1,1,0, \dots, n-5, n-5)$$
$$r(a_4^1 a_5^3 | S^n) = (3,3,2,2,1,2,0,1,1,0, \dots, n-5, n-5)$$

$$\vdots$$

$$r(a_{(n-2)}^1 a_{(n-1)}^1 | S^n) = (n-3, n-3, n-4, n-4, \dots, p_1, q_1)$$
$$r(a_{(n-2)}^2 a_{(n-1)}^1 | S^n) = (n-3, n-3, n-4, n-4, \dots, p_2, q_2)$$
$$r(a_{(n-2)}^2 a_{(n-1)}^2 | S^n) = (n-3, n-3, n-4, n-4, \dots, p_3, q_3)$$
$$r(a_{(n-2)}^3 a_{(n-1)}^2 | S^n) = (n-3, n-3, n-4, n-4, \dots, p_4, q_4)$$
$$r(a_{(n-2)}^3 a_{(n-1)}^3 | S^n) = (n-3, n-3, n-4, n-4, \dots, p_5, q_5)$$
$$r(a_{(n-2)}^1 a_{(n-1)}^3 | S^n) = (n-3, n-3, n-4, n-4, \dots, p_6, q_6)$$

$$r(a_{(n-1)}^1 a_n^1 | S^n) = (n-2, n-2, n-3, n-3, \dots, 2,2, r_1, s_1, t_1, u_1, v_1, w_1)$$

$$r(a_{(n-1)}^2 a_n^1 | S^n) = (n-2, n-2, n-3, n-3, \dots, 2,2, r_2, s_2, t_2, u_2, v_2, w_2)$$

$$r(a_{(n-1)}^2 a_n^2 | S^n) = (n-2, n-2, n-3, n-3, \dots, 2,2, r_3, s_3, t_3, u_3, v_3, w_3)$$

$$r(a_{(n-1)}^3 a_n^2 | S^n) = (n-2, n-2, n-3, n-3, \dots, 2,2, r_4, s_4, t_4, u_4, v_4, w_4)$$

$$r(a_{(n-1)}^3 a_n^3 | S^n) = (n-2, n-2, n-3, n-3, \dots, 2,2, r_5, s_5, t_5, u_5, v_5, w_5)$$

$$r(a_{(n-1)}^1 a_n^3 | S^n) = (n-2, n-2, n-3, n-3, \dots, 2,2, r_6, s_6, t_6, u_6, v_6, w_6)$$

where:

$$(p_1, q_1) = \begin{cases} (1,1) \text{ if } n = 3t-2, t \in N \\ (1,2) \text{ if } n = 3t-1, t \in N \\ (2,1) \text{ otherwise} \end{cases}, \quad (p_2, q_2) = \begin{cases} (1,1) \text{ if } n = 3t-2, t \in N \\ (1,2) \text{ if } n = 3t-1, t \in N \\ (2,1) \text{ otherwise} \end{cases}$$

$$(p_3, q_3) = \begin{cases} (1,2) \text{ if } n = 3t-2 \text{ or } n = 3t, t \in N \\ (1,1) \text{ otherwise} \end{cases}, \quad (p_4, q_4) = \begin{cases} (1,2) \text{ if } n = 3t-2 \text{ or } n = 3t, t \in N \\ (1,1) \text{ otherwise} \end{cases}$$

$$(p_5, q_5) = \begin{cases} (1,1) \text{ if } n = 3t, t \in N \\ (2,1) \text{ otherwise} \end{cases}, \quad (p_6, q_6) = \begin{cases} (1,1) \text{ if } n = 3t, t \in N \\ (2,1) \text{ otherwise} \end{cases}$$

$$(r_1, s_1) = \begin{cases} (1,1) \text{ if } n = 3t-2, t \in N \\ (1,2) \text{ otherwise} \end{cases}, \quad (r_2, s_2) = \begin{cases} (1,1) \text{ if } n = 3t-1, t \in N \\ (2,1) \text{ otherwise} \end{cases}$$

$$(r_3, s_3) = \begin{cases} (1,1) \text{ if } n = 3t-2, t \in N \\ (2,1) \text{ otherwise} \end{cases}, \quad (r_4, s_4) = \begin{cases} (1,2) \text{ if } n = 3t-2, t \in N \\ (2,1) \text{ if } n = 3t-1, t \in N \\ (1,1) \text{ otherwise} \end{cases}$$

$$(r_5, s_5) = \begin{cases} (1,2) \text{ if } n = 3t-2, t \in N \\ (2,1) \text{ if } n = 3t-1, t \in N \\ (1,1) \text{ otherwise} \end{cases}, \quad (r_6, s_6) = \begin{cases} (1,1) \text{ if } n = 3t-2, t \in N \\ (1,2) \text{ otherwise} \end{cases}$$

$$(t_1, u_1) = \begin{cases} (1,1) \text{ if } n = 3t-2, t \in N \\ (0,1) \text{ otherwise} \end{cases}, \quad (t_2, u_2) = \begin{cases} (0,1) \text{ if } n = 3t-2, t \in N \\ (1,1) \text{ if } n = 3t-1, t \in N \\ (1,0) \text{ otherwise} \end{cases}$$

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

$$(t_3,u_3)=\begin{cases}(0,1)\ \text{if } n=3t-2, t\in N\\(1,1)\ \text{if } n=3t-1, t\in N\\(1,0)\ \text{otherwise}\end{cases},\ (t_4,u_4)=\begin{cases}(1,1)\ \text{if } n=3t, t\in N\\(1,0)\ \text{otherwise}\end{cases}$$

$$(t_5,u_5)=\begin{cases}(1,1)\ \text{if } n=3t, t\in N\\(1,0)\ \text{otherwise}\end{cases},\ (t_6,u_6)=\begin{cases}(1,1)\ \text{if } n=3t-2, t\in N\\(0,1)\ \text{otherwise}\end{cases}$$

$$(v_1,w_1)=\begin{cases}(1,1)\ \text{if } n=3t, t\in N\\(0,1)\ \text{otherwise}\end{cases},\ (v_2,w_2)=\begin{cases}(1,1)\ \text{if } n=3t, t\in N\\(0,1)\ \text{otherwise}\end{cases}$$

$$(v_3,w_3)=\begin{cases}(1,1)\ \text{if } n=3t-2, t\in N\\(1,0)\ \text{if } n=3t-1, t\in N\\(0,1)\ \text{otherwise}\end{cases},\ (v_4,w_4)=\begin{cases}(1,1)\ \text{if } n=3t-2, t\in N\\(1,0)\ \text{if } n=3t-1, t\in N\\(0,1)\ \text{otherwise}\end{cases}$$

$$(v_5,w_5)=\begin{cases}(1,1)\ \text{if } n=3t, t\in N\\(1,0)\ \text{otherwise}\end{cases},\ (v_6,w_6)=\begin{cases}(1,1)\ \text{if } n=3t-1, t\in N\\(0,1)\ \text{otherwise}\end{cases}$$

Because of all the edge representation are distinct, thus S^n is basis of A_3^n and $dim_e(A_3^n)\le 2n$. ■

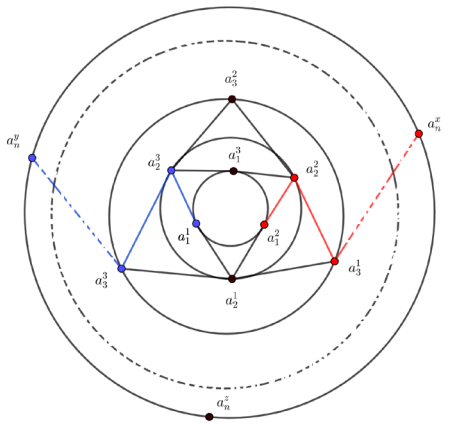


Figure 3. A basis of the generalized antiprism graph A_3^n

Theorem 2. MMD of the generalized antiprism graph A_3^n, $dim_m(A_3^n)\le 2n$.

Proof. Choose the set S^n as described in Theorem. 1. Consequently, the representation of the edges is presented as in the proof of Theorem 1. Furthermore, each vertex's representation in A_3^n can be stated as:

$$r(a_1^1|S^n)=(0,1,2,1,2,2,\dots,n-1,n-1)$$
$$r(a_1^2|S^n)=(1,0,1,2,2,2,\dots,n-1,n-1)$$
$$r(a_1^3|S^n)=(1,1,1,1,2,2,\dots,n-1,n-1)$$
$$r(a_2^1|S^n)=(1,1,1,1,1,1,\dots,n-2,n-2)$$
$$r(a_2^2|S^n)=(2,1,0,1,1,2,\dots,n-2,n-2)$$
$$r(a_2^3|S^n)=(1,2,1,0,2,1,\dots,n-2,n-2)$$
$$r(a_3^1|S^n)=(2,2,1,2,0,1,\dots,n-3,n-3)$$
$$r(a_3^2|S^n)=(2,2,1,1,1,1,\dots,n-3,n-3)$$
$$r(a_3^3|S^n)=(2,2,2,1,1,0,\dots,n-3,n-3)$$
$$\vdots$$
$$r(a_{(n-1)}^1|S^n)=(n-2,n-2,n-3,n-3,\dots,b_1,c_1)$$
$$r(a_{(n-1)}^2|S^n)=(n-2,n-2,n-3,n-3,\dots,b_2,c_2)$$
$$r(a_{(n-1)}^3|S^n)=(n-2,n-2,n-3,n-3,\dots,b_3,c_3)$$
$$r(a_n^1|S^n)=(n-1,n-1,n-2,n-2,,\dots,2,2,d_1,e_1,f_1,g_1)$$
$$r(a_n^2|S^n)=(n-1,n-1,n-2,n-2,,\dots,2,2,d_2,e_2,f_2,g_2)$$
$$r(a_n^3|S^n)=(n-1,n-1,n-2,n-2,,\dots,2,2,d_3,e_3,f_3,g_3)$$

where:

$$(b_1,c_1)=\begin{cases}(1,2)\ \text{if } n=3t-2, t\in N\\(2,1)\ \text{if } n=3t-1, t\in N\\(1,1)\ \text{otherwise}\end{cases},(b_2,c_2)=\begin{cases}(1,1)\ \text{if } n=3t-2, t\in N\\(1,2)\ \text{otherwise}\end{cases}$$

$$(b_3,c_3)=\begin{cases}(1,1)\ \text{if } n=3t-1, t\in N\\(2,1)\ \text{otherwise}\end{cases},\ (d_1,e_1)=\begin{cases}(1,1)\ \text{if } n=3t-1, t\in N\\(1,2)\ \text{otherwise}\end{cases}$$

$$(d_2,e_2)=\begin{cases}(1,1)\ \text{if } n=3t, t\in N\\(2,1)\ \text{otherwise}\end{cases},\ (d_3,e_3)=\begin{cases}(1,1)\ \text{if } n=3t-2, t\in N\\(1,2)\ \text{if } n=3t-1, t\in N\\(2,1)\ \text{otherwise}\end{cases}$$

$$(f_1,g_1)=\begin{cases}(1,1)\ \text{if } n=3t-1, t\in N\\(0,1)\ \text{otherwise}\end{cases},\ (f_2,g_2)=\begin{cases}(1,0)\ \text{if } n=3t-2, t\in N\\(0,1)\ \text{if } n=3t-1, t\in N\\(1,1)\ \text{otherwise}\end{cases}$$

$$(f_3,g_3)=\begin{cases}(2,1)\ \text{if } n=3t-2, t\in N\\(1,0)\ \text{otherwise}\end{cases}$$

Because of all the edge and vertex representations being different, S^n is a mixed generator of A_3^n and $dim_m(A_3^n)\le 2n$. ■

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

Table 6. Results of BGWO Algorithm for EMD on generalized antiprism graphs A_3^n

Graph	n	m	popsize	It	dim_e	edge metric generator
A_3^3	9	21	20	1	6	$\{a_2, a_3, a_5, a_6, a_7, a_8\}$
A_3^4	12	30	20	1	8	$\{a_1, a_3, a_4, a_6, a_8, a_9, a_{11}, a_{12}\}$
A_3^5	15	39	20	2	10	$\{a_1, a_2, a_4, a_5, a_8, a_9, a_{10}, a_{12}, a_{13}, a_{14}\}$
A_3^6	18	48	20	1	12	$\{a_1, a_3, a_4, a_5, a_8, a_9, a_{10}, a_{11}, a_{13}, a_{14}, a_{16}, a_{18}\}$
A_3^7	21	57	20	1	14	$\{a_1, a_2, a_4, a_5, a_8, a_9, a_{10}, a_{12}, a_{14}, a_{15}, a_{16}, a_{18}, a_{19}, a_{21}\}$
A_3^8	24	66	20	2	16	$\{a_2, a_3, a_4, a_5, a_8, a_9, a_{11}, a_{12}, a_{14}, a_{15}, a_{17}, a_{18}, a_{19}, a_{21}, a_{22}, a_{23}\}$
A_3^9	27	75	20	2	18	$\{a_2, a_3, a_5, a_6, a_7, a_8, a_{10}, a_{11}, a_{13}, a_{15}, a_{17}, a_{18}, a_{19}, a_{21}, a_{22}, a_{23}, a_{26}, a_{27}\}$
A_3^{10}	30	84	20	2	20	$\{a_1, a_2, a_4, a_6, a_7, a_9, a_{10}, a_{12}, a_{13}, a_{15}, a_{16}, a_{18}, a_{20}, a_{21}, a_{23}, a_{24}, a_{25}, a_{27}, a_{28}, a_{30}\}$
A_3^{11}	33	93	20	2	22	$\{a_2, a_3, a_5, a_6, a_7, a_9, a_{11}, a_{12}, a_{14}, a_{15}, a_{16}, a_{18}, a_{19}, a_{20}, a_{23}, a_{24}, a_{26}, a_{27}, a_{28}, a_{30}, a_{32}, a_{33}\}$
A_3^{12}	36	102	20	2	24	$\{a_1, a_2, a_5, a_6, a_7, a_9, a_{10}, a_{11}, a_{14}, a_{15}, a_{17}, a_{18}, a_{20}, a_{21}, a_{22}, a_{23}, a_{25}, a_{27}, a_{29}, a_{30}, a_{31}, a_{32}, a_{34}, a_{35}\}$

Table 7. Results of BGWO Algorithm for MMD on generalized antiprism graphs A_3^n

Graph	n	m	popsize	It	dim_e	mixed metric generator
A_3^3	9	21	20	1	6	$\{a_1, a_2, a_5, a_6, a_7, a_9\}$
A_3^4	12	30	20	4	8	$\{a_2, a_3, a_4, a_6, a_7, a_8, a_{11}, a_{12}\}$
A_3^5	15	39	20	1	10	$\{a_1, a_2, a_5, a_6, a_7, a_9, a_{10}, a_{11}, a_{14}, a_{15}\}$
A_3^6	18	48	20	1	12	$\{a_2, a_3, a_4, a_6, a_7, a_8, a_{11}, a_{12}, a_{13}, a_{15}, a_{16}, a_{17}\}$
A_3^7	21	57	20	7	14	$\{a_1, a_3, a_4, a_5, a_8, a_9, a_{10}, a_{12}, a_{14}, a_{15}, a_{16}, a_{18}, a_{19}, a_{20}\}$
A_3^8	24	66	20	5	16	$\{a_2, a_3, a_4, a_6, a_7, a_8, a_{10}, a_{12}, a_{14}, a_{15}, a_{16}, a_{17}, a_{20}, a_{21}, a_{22}, a_{24}\}$
A_3^9	27	75	20	4	18	$\{a_2, a_3, a_4, a_6, a_7, a_8, a_{11}, a_{12}, a_{14}, a_{15}, a_{16}, a_{18}, a_{19}, a_{20}, a_{23}, a_{24}, a_{25}, a_{27}\}$
A_3^{10}	30	84	20	3	20	$\{a_1, a_2, a_5, a_6, a_7, a_9, a_{10}, a_{12}, a_{13}, a_{15}, a_{16}, a_{18}, a_{19}, a_{21}, a_{22}, a_{24}, a_{25}, a_{26}, a_{29}, a_{30}\}$
A_3^{11}	33	93	20	8	22	$\{a_1, a_2, a_5, a_6, a_7, a_9, a_{10}, a_{11}, a_{13}, a_{15}, a_{16}, a_{17}, a_{20}, a_{21}, a_{22}, a_{24}, a_{25}, a_{26}, a_{29}, a_{30}, a_{31}, a_{33}\}$
A_3^{12}	36	102	20	13	24	$\{a_1, a_2, a_5, a_6, a_7, a_9, a_{10}, a_{12}, a_{13}, a_{14}, a_{16}, a_{18}, a_{19}, a_{20}, a_{22}, a_{23}, a_{26}, a_{27}, a_{29}, a_{30}, a_{31}, a_{33}, a_{34}, a_{35}\}$

The following presents the simulation results of the BGWO algorithm for determining the EMD and MMD of the generalized antiprism, as shown in Tables 6 and 7. The results demonstrate that the BGWO algorithm yields values consistent with those established in Theorems 1 and 2, with efficient completion times indicated by the required number of iterations.

5. Conclusion

This paper presents a modified binary Gray Wolf Optimizer (BGWO) algorithm to address the mixed-metric dimension problem. The algorithm is designed to reduce the number of vertices required to uniquely identify all elements of a graph based on distance constraints. The proposed BGWO algorithm incorporates several enhancements aimed at improving performance in the MMDP. First, validation of the uniqueness of the row ensures that all solutions maintain valid matrix representations. Second, dynamic stagnation handling detects and resolves stagnation through partial population restarts. Third, mutation promotes diversity and prevents premature convergence. Lastly, iterative crossover balances exploration and exploitation by dynamically updating the positions of the wolves. In some graphs with many vertices or edges, an elitism process is necessary. Elitism ensures that the best solution found so far is never lost during iterations. The results demonstrate that the modified BGWO algorithm achieves efficient completion times and delivers competitive outcomes for this problem.

Moreover, the BGWO algorithm shows practical utility in the following areas: (1) conducting pattern recognition to identify resolving set patterns within specific graph families. The algorithm's outputs provide various minimum resolving sets, allowing the selection of resolving sets for pattern identification and analysis; (2) developing hypotheses about the mixed metric dimension values for specific types or families of graphs; (3) generating mixed metric dimension values that may coincide with or fall below the upper bounds obtained from theoretical results. This phenomenon occurs because theoretical results rely on fixed patterns or regularities, while the BGWO algorithm operates through random calculations using an iterative refinement mechanism.

6. Acknowledgments

The author sincerely acknowledges the support provided by various divisions within the Ministry of Higher Education, Science, and Technology through the BPI doctoral research scholarship and Universitas Kadiri, which facilitated the completion of this study.

7. References

Abbas, Waseem et al. 2024. "Investigating Metric Dimension and Edge Metric Dimension of Hexagonal Boron Nitride and Carbon Nanotubes." *European Journal of Pure and Applied Mathematics* 17(3): 2055–72.

Beerliova, Zuzana et al. 2006. "Network Discovery and Verification." *IEEE Journal on Selected Areas in Communications* 24(12): 2168–81.

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

Chartrand, Gary, Linda Eroh, Mark A. Johnson, and Ortrud R. Oellermann. 2000. "Resolvability in Graphs and the Metric Dimension of a Graph." *Discrete Applied Mathematics* 105(1–3): 99–113.

Chelladurai, Xavier. 2020. "Algorithms for the Metric Dimension of a Simple Graph." *Data Science and Security, Lectures Notes in Network and Systems* (132): 91–105.

Chen, Xujin, Xiaodong Hu, and Changjun Wang. 2016. "Approximation for the Minimum Cost Doubly Resolving Set Problem." *Theoretical Computer Science* 609: 526–43. http://dx.doi.org/10.1016/j.tcs.2015.03.048.

Danas, Milica Milivojević. 2021. "The Mixed Metric Dimension of Flower Snarks and Wheels." *Open Mathematics* 19(1): 629–40.

Darmaji, and N. Azahra. 2022. "The Mixed Metric Dimension of Wheel-like Graphs." *Journal of Physics: Conference Series* 2157(1).

Diaz, Josep, Olli Pottonen, Maria Serna, and Erik Jan van Leeuwen. 2017. "Complexity of Metric Dimension on Planar Graphs." *Journal of Computer and System Sciences* 83(1): 132–58. http://dx.doi.org/10.1016/j.jcss.2016.06.006.

Emary, E., Hossam M. Zawbaa, and Aboul Ella Hassanien. 2016. "Binary Grey Wolf Optimization Approaches for Feature Selection." *Neurocomputing* 172: 371–81. http://dx.doi.org/10.1016/j.neucom.2015.06.083.

Harrary, F, and R.A Melter. 1976. "On the Metric Dimension of a Graph." *Ars Combin* 2: 191–95.

Hauptmann, Mathias, Richard Schmied, and Claus Viehmann. 2012. "Approximation Complexity of Metric Dimension Problem." *Journal of Discrete Algorithms* 14: 214–22. http://dx.doi.org/10.1016/j.jda.2011.12.010.

Hendy, Hendy, Mohammad Isa, Imam Mukhlash, and Samsul Setumin. 2023. "A Bibliometric Analysis of Metaheuristic Research and Its Applications." *Register: Jurnal Ilmiah Teknologi Sistem Informasi* 9(January): 1–17. http://www.journal.unipdu.ac.id/index.php/register/article/view/2675.

Hertz, Alain. 2020. "An IP-Based Swapping Algorithm for the Metric Dimension and Minimal Doubly Resolving Set Problems in Hypercubes." *Optimization Letters* 14(2): 355–67.

Hoffmann, Stefan, Alina Elterman, and Egon Wanke. 2016. "A Linear Time Algorithm for Metric Dimension of Cactus Block Graphs." *Theoretical Computer Science* 630: 43–62. http://dx.doi.org/10.1016/j.tcs.2016.03.024.

Imran, Muhammad, Ali Ahmad, Muhammad Azeem, and Kashif Elahi. 2022. "Metric-Based Resolvability of Quartz Structure." *Computers, Materials and Continua* 71(1): 2053–71.

Kelenc, Aleksander, Dorota Kuziak, Andrej Taranenko, and Ismael G. Yero. 2017. "Mixed Metric Dimension of Graphs." *Applied Mathematics and Computation* 314: 429–38. http://dx.doi.org/10.1016/j.amc.2017.07.027.

Kelenc, Aleksander, Niko Tratnik, and Ismael G. Yero. 2018. "Uniquely Identifying the Edges of a Graph: The Edge Metric Dimension." *Discrete Applied Mathematics* 251(January 2016): 204–20.

Khuller, Samir, Balaji Raghavachari, and Azriel Rosenfeld. 1996. "Landmarks in Graphs." *Discrete Applied Mathematics* 70(3): 217–29.

Knor, Martin et al. 2021. "Graphs with the Edge Metric Dimension Smaller than the Metric Dimension." *Applied Mathematics and Computation* 401.

Koam, Ali N.A., Ali Ahmad, Sadia Husain, and Muhammad Azeem. 2023. "Mixed Metric Dimension of Hollow Coronoid Structure." *Ain Shams Engineering Journal* 14(7): 102000. https://doi.org/10.1016/j.asej.2022.102000.

Kratica, Jozef, Vera Kovačević-Vujčić, and Mirjana Čangalović. 2009. "Computing the Metric Dimension of Graphs by Genetic Algorithms." *Computational Optimization and Applications* 44(2): 343–61.

Kratica, Jozef, Vera Kovačević-Vujčić, Mirjana Čangalović, and Milica Stojanović. 2012. "Minimal Doubly Resolving Sets and the Strong Metric Dimension of Some Convex Polytopes." *Applied Mathematics and Computation* 218(19): 9790–9801.

Melter, Robert A., and Ioan Tomescu. 1984. "Metric Bases in Digital Geometry." *Computer Vision, Graphics and Image Processing* 25(1): 113–21.

Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. 2014. "Grey Wolf Optimizer." *Advances in Engineering Software* 69: 46–61. http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

Mladenović, Nenad, Jozef Kratica, Vera Kovačević-Vujčić, and Mirjana Čangalović. 2012. "Variable Neighborhood Search for Metric Dimension and Minimal Doubly Resolving Set Problems." *European Journal of Operational Research* 220(2): 328–37.

Mohame, Basma, Mohammed Ami, and Linda Mohaise. 2023. "Binary Archimedes Optimization Algorithm for Computing Dominant Metric Dimension Problem." *Intelligent Automation and Soft Computing* 38(1): 19–34.

Muhammad, F., and L. Susilowati. 2020. "Algorithm and Computer Program to Determine Metric Dimension of Graph." *Journal of Physics: Conference Series* 1494(1): 0–5.

Murdiansyah, D.T, and Adiwijaya. 2017. "Computing the Metric Dimension of Hypercube Graphs by Particle Swarm Optimization Algorithm." *Recent Advances on Soft Computing and Data Mining, Advances in Intelligent Systems and Computing* 549.

Peterin, Iztok, and Ismael G. Yero. 2020. "Edge Metric Dimension of Some Graph Operations." *Bulletin of the Malaysian*

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)

Mathematical Sciences Society 43(3): 2465–77. https://doi.org/10.1007/s40840-019-00816-7.

Seb, Andris, and Eric Tannier. 2016. "On Metric Generators of Graphs Author (s): András Sebő and Eric Tannier Published by : INFORMS Stable URL : Http://Www.Jstor.Org/Stable/30035689 On Metric Generators of Graphs." 29(2): 383–93.

Sedlar, Jelena, and Riste Škrekovski. 2021a. "Extremal Mixed Metric Dimension with Respect to the Cyclomatic Number." *Applied Mathematics and Computation* 404: 126238. https://doi.org/10.1016/j.amc.2021.126238.

Sedlar, Jelena, and Riste Škrekovski.. 2021b. "Mixed Metric Dimension of Graphs with Edge Disjoint Cycles." *Discrete Applied Mathematics* 300: 1–8. https://doi.org/10.1016/j.dam.2021.05.004.

Slater, P.J. 1975. "Leaves of Trees." *Congr. Numer* 14: 549–59.

Tillquist, Richard C., and Manuel E. Lladser. 2019. "Low-Dimensional Representation of Genomic Sequences." *Journal of Mathematical Biology*. https://doi.org/10.1007/s00285-019-01348-1.

Vietz, Duygu, Stefan Hoffmann, and Egon Wanke. 2019. 11355 LNCS Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) *Computing the Metric Dimension by Decomposing Graphs into Extended Biconnected Components: (Extended Abstract)*. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-10564-8_14.

Zhang, Yuezhong, and Suogang Gao. 2020. "On the Edge Metric Dimension of Convex Polytopes and Its Related Graphs." *Journal of Combinatorial Optimization* 39(2): 334–50. https://doi.org/10.1007/s10878-019-00472-4.

Zhu, Enqiang, Andrej Taranenko, Zehui Shao, and Jin Xu. 2019. "On Graphs with the Maximum Edge Metric Dimension." *Discrete Applied Mathematics* 257: 317–24. https://doi.org/10.1016/j.dam.2018.08.031.

DOI: https://doi.org/10.22452/mjs.vol45sp2.5
Malaysian Journal of Science 45 (Special Issue 2): 43-54 (November 2026)