Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-26
https://doi.org/10.22049/cco.2025.29133.1855

CCO
Commun. Comb. Optim.

Research Article

Hybrid ant colony optimization algorithm with binary gray wolf optimization for detour metric dimension and bi-metric dimension problem

H. Hendy[1,2], Mohammad Isa Irawan[1,*], Imam Mukhlash[1,†], R. Rinurwati[1,‡]

[1]Department of Mathematics, Faculty of Science and Data Analytics,
Institut Teknologi Sepuluh Nopember,
Surabaya, Indonesia
[*]m.isa.irawan@gmail.com
[†]7002211006@mhs.its.ac.id
[‡]rinur@matematika.its.ac.id

[2]Faculty of Engineering, Universitas Kadiri, Kediri, Indonesia
hendy@unik-kediri.ac.id

Received: 31 October 2023; Accepted: 17 February 2025
Published Online: 25 April 2025

Abstract: In this work, two class NP-hard optimization problems on the graph are discussed: the detour metric dimension and the bi-metric dimension. Both are used in many distinct areas, as well as pattern recognition, keeping track of the movement of robots on a network, and reviewing the structural properties of chemical structures. The metric dimension $dim(G)$ of graph G is the minimum number of vertices such that every vertex of G is uniquely assigned by its vector of distances to the selected vertices. This concept was expanded into the detour metric dimension $D\beta(G)$ and the bi-metric dimension $\beta_b(G)$ by considering the detour distance of two vertices. A computational approach is needed to solve these two problems on large graphs. In this research, we propose the BGWO algorithm to determine the metric dimension of some generalized antiprism graphs. In addition, we develop a probabilistic-based metaheuristic algorithm, namely ant colony optimization, to find the detour distance and then modify the binary gray wolf optimization (BGWO) algorithm to solve the detour metric dimension and the bi-metric dimension on some families of graphs. The simulation shows that the BGWO algorithm gives better results for the generalized antiprism graphs. Also, the hybrid ACO-BGWO algorithm gives the same detour dimension result as in the literature. We show that the bi-metric dimension of the generalized antiprism graph is the same as its metric dimension.

Keywords: distance in graph, computational intelligence, metaheuristic algorithm, resolving set.

AMS Subject classification: 05C85, 68T20, 05C12, 90C59

* *Corresponding Author*

© 2025 Azarbaijan Shahid Madani University

1. Introduction

The notion of a resolving set first came in 1975, proposed by Slater [36]. Harrary and Melter construct the same concept with a different name, metric dimension [15]. In simple terms, the metric dimension of a connected graph A or dim(A) is the cardinality of the minimum resolving set. The minimum resolving set is also often mentioned as the basis. It can also be concluded that all of the vertices of graph A have a disparate representation of the basis [4].
Let G be a connected graph. For an ordered set $W = \{w_1, w_2, \ldots, w_k\}$ of vertices and a vertex v in a connected graph G, the $k-$vector $r(v|W) := (d(v, w_1), (d(v, w_2), \ldots, (d(v, w_k))$ is called the metric dimension of v with respect to W, where $d(x, y)$ is the distance between two vertices x and y. The set W is called a resolving set for G if distinct vertices of G have distinct representation with respect to W. A minimum resolving set is called a basis, and the metric dimension of G, $dim(G)$, is the cardinality of a basis for G. For example, consider the graph G of Figure 1. The set $W_1 = \{w_1, w_5\}$ is not a resolving set for G since $r(w_2|W_1) = (1, 1) = r(w_3|W_1)$. On the other hand, $W_2 = \{w_2, w_3, w_5\}$ is a resolving set for G since the representation for the vertices of G with respect to W_2 are: $r(w_1|W_2) = (1, 1, 2)$, $r(w_2|W_2) = (0, 1, 1)$, $r(w_3|W_2) = (1, 0, 1)$, $r(w_4|W_2) = (2, 1, 1)$, $r(w_5|W_2) = (1, 1, 0)$.
However, W_2 is not a minimum resolving set since $W_3 = \{w_1, w_4\}$ is also resolving set. Since no single vertex constitutes a resolving set for G, it follows that W_3 is a minimum resolving set [4].

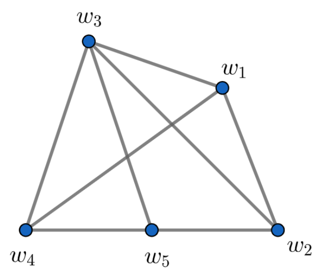

Figure 1. **Graph** G

Research on the metric dimension has significantly advanced conceptual development and its application across various fields. For conceptual development, metric dimensions have evolved into multiple frameworks addressing various resolving set problems, achieved by establishing new definitions of the resolving set itself. These advancements include the detour metric dimension and the bi-metric dimension.
The Gray Wolf Optimizer (GWO) is a population-based metaheuristic algorithm inspired by the hunting behavior of gray wolves. It was developed by mimicking their social structure and hunting strategies. The bibliometric analysis results show that the GWO method is computer science's most cited metaheuristic method [17]. The binary version of GWO: BGWO has been proven to solve various problems in science,

health science, and engineering [9],[22],[32] and [30]. The results of the first part of this paper show that the BGWO algorithm gives satisfactory results in determining the metric dimension of a generalized antiprism graph. The resulting metric dimension for a generalized antiprism graph is smaller than the upper bound, which has been previously proven. They all motivate applying the BGWO algorithm to solve the detour metric and bi-metric dimension problems.

Next, to determine the detour metric and bi-metric dimensions, it is necessary to compute the longest distance between two vertices in a graph. A computational tool is needed to handle this task in large graphs because manually calculating it would take time and effort. The ant colony optimization (ACO) algorithm is a probabilistic method employed to address computational problems that can be transformed into the task of identifying optimal paths within graphs. The ACO algorithm is inspired by the foraging behavior of ant colonies. This algorithm effectively solves combinatorial problems like the Traveling Salesman Problem (TSP) [1, 8]. We are motivated to apply the ACO algorithm to determine the longest distance between two vertices in a graph.

In this paper, we give a better result for the metric dimension of some generalized antiprism graphs than its upper bound. We get the result from the computational approach, Binary Gray Wolf Optimization Algorithm. Because there has been no computational approach to solve the detour metric dimension and bi-metric dimension problem, we propose a hybrid algorithm between the ACO and BGWO algorithms for determining the detour metric dimension and the bi-metric dimension. To determine their detour metric dimension, we apply that hybrid algorithm to several family graphs: cycle, Jahangir, and friendship graphs. Also, we apply that hybrid algorithm to some families of graphs: complete graphs and generalized antiprism graphs.

The organization of this paper is as follows: the definition and example of the concept of metric dimension, detour metric dimension, and bi-metric dimension are presented in Section 2. In Section 3, the details of the ACO Algorithm and its procedure are introduced in Section 3.1, and the details of Gray Wolf Optimization and its concept are described in 3.2. The proposed algorithm is discussed in Section 4. In Section 5, computational results are reported.

2. Metric, detour metric, and bi-metric dimension

2.1. Metric dimension

The study of the metric dimension problem (MDP) continues to expand. The advancement of this field begins with the determination of the metric dimensions for various families of graphs. Hernando investigated the metric dimension of some of them [18]. Fehr attempted to identify the metric dimensions of the Cayley digraph [11]. Imran and Vetrik investigated circular graphs metric dimensions [19, 38]. Chau continued his research by examining the MDP of circular graphs and their operation results [6], Vetrik again continued his research on the metric dimensions of circulant graphs and directed circulant graphs [39]. Guo has carried out research on metric dimensions that have regular distances [14]. In addition, Feng succeeded in identifying the metric dimension of graphs with bilinear form [12]; Feng researched the metric dimensions of line graphs [13]. Eroh researched the metric dimensions of the functigraph [10]. Saputro researched the metric dimensions of a biregular graph [34]. Shao conducted a study of the metric dimensions of some generalized Petersen graphs [35], and elsewhere, Imran conducted a study of the metric dimensions of the generalized Petersen multigraph family [20] and gear graphs [21]. Liu investigated the metric dimensions of several Toeplitz graph families [24]. In the same year, Akhter studied the metric dimensions of Fullerene graphs [2], and Sooryanarayana studied the metric dimensions of generalized wheel graphs [37]. Bensmail found the relationship between the metric dimensions of a graph and an oriented graph [3]. Rehman investigated the metric dimensions of arithmetic graphs [33]. In [16], Jason described a character of the 2-dimensional tree.

The metric dimension problem (MDP) was first modeled as an integer programming by Chartrand et al. [5]. In the computational approach, the range of its solution space spreads exponentially with the problem dimension. To handle this, the implementation of metaheuristic algorithms called genetic algorithms has been studied by Kratica [23]. In [29], Murdiansyah presented a particle swarm optimization (PSO) to solve MDP. Mladenovic designed a variable neighborhood search technique for MDP and searched a minimal doubly resolving set of graphs [26]. Mohamed et al. presented a binary of equilibrium to find the connected domination metric dimension [28]. In [40], Wu et al. combined the hybrid algorithm with graph representation learning to solve MDP. Mohamed introduced a hybrid of the water cycle and whale optimization methods to solve MDP and manage the optimization procedures [27].

2.2. Detour metric dimension

The concept of MDP was developed. One of which is the detour metric dimension. Let G be a connected graph. The set V is the vertex set of G. The set E is the edge set of G. Let $u, v \in V$. The distance $D(u, v)$ is the length of the longest $u - v$ path in G. Let $W^\star = \{w_1, w_2, \ldots, w_p\}$ be an ordered subset of V. For every $v \in V$, a representation of v with respect to $W^\star$ is defined as p tuples, $R(v|W^\star) = (D(v, w_1), D(v, w_2), \ldots, D(v, w_p))$. The set $W^\star$ is a *detour resolving set* of G if every two distinct vertices $u, v \in V$ satisfy $R(u|W^\star) \neq R(v|W^\star)$. A detour basis of G is a detour resolving set of G with minimum cardinality, and the detour metric dimension of G refers to its cardinality, denoted by $D\beta(G)$ [7]. For example, consider the cycle graph C_8 of Figure 2. The detour distance matrix of this graph is:

$$\begin{bmatrix} 0 & 7 & 6 & 5 & 4 & 5 & 6 & 7 \\ 7 & 0 & 7 & 6 & 5 & 4 & 5 & 6 \\ 6 & 7 & 0 & 7 & 6 & 5 & 4 & 5 \\ 5 & 6 & 7 & 0 & 7 & 6 & 5 & 4 \\ 4 & 5 & 6 & 7 & 0 & 7 & 6 & 5 \\ 5 & 4 & 5 & 6 & 7 & 0 & 7 & 6 \\ 6 & 5 & 4 & 5 & 6 & 7 & 0 & 7 \\ 7 & 6 & 5 & 4 & 5 & 6 & 7 & 0 \end{bmatrix}$$

From this matrix, it can be seen that each column always contains at least two identical elements. This shows that no detour resolving set of C_8 consists of only one vertex. Next, suppose $S = \{v_2, v_4\}$. The representation of each vertex in S is as shown in Figure 2. Thus, S is the minimum detour resolving set of C_8 and $D\beta(C_8) = 2$

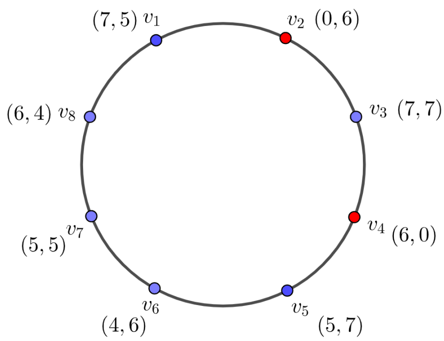

Figure 2. **Cycle graph C_8**

2.3. Bi-metric Dimension

An additional advancement in the Metric Dimension Problems (MDP) concept has been achieved. For any two vertices x and y, $d(x, y)$ and $\delta(x, y)$ respectively denote the length of the shortest and longest path between x and y and are namely *distance* and *detour distance* between x and y. Let $G(V, E)$ be a simple connected graph. For each vertex $x \in V$, we link a pair of vectors (u, v), denoted by S_x, with respect to a subset $S = \{s_1, s_2, \ldots, s_k\}$ of vertices of G where $u = (d(x, s_1), d(x, s_2), \ldots, d(x, s_k))$ and $v = (\delta(x, s1), \delta(x, s2), \ldots, \delta(x, sk))$. The subset S is then said to *bi-resolve* G if $S_x \neq Sy$, whenever $x \neq y$. The minimum cardinality of a *bi-resolving set* S is termed as bi-metric dimension of G and is denoted by $\beta_b(G)$ [31]. There is a difference in [7] and [31] in defining the longest distance from a vertex to itself. In [7], the longest distance from a vertex v to itself is 0. However, in [31], the longest distance from a vertex v to itself is sought according to the given definition: the length of the longest path from v to v. In this paper, we use the definition in [7] to find the detour metric dimension and the definition in [31] to find the bi-metric dimension.
To illustrate finding the bi-resolving set of a graph, consider the graph G in Figure 1.

The bi-representation of each vertex in G with respect to W_3 is shown in Figure 3. Since no single vertex constitutes a bi-resolving set for G, it follows that W_3 is also a minimum bi-resolving set, and thus $\beta_b(G) = 2$.

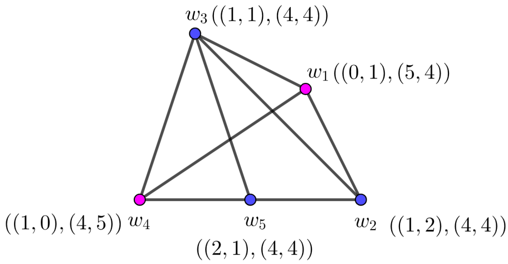

Figure 3. **Bi-representation of G**

Furthermore, consider the graph H as shown in Figure 4. The set $S = \{w_1, w_2, w_3, w_5\}$ is a minimum resolving set; hence, their metric dimension is 4. But the set $M = \{w_1, w_3, w_6\}$ is a minimum bi-resolving set for the graph H. Thus, the bi-metric dimension of $H = \beta_b(H) = 3 < 4 = dim(H)$. So we are motivated to study the bi-metric dimension further. In certain graphs, the bi-metric dimension parameter yields a smaller value than the metric dimension. This will lead to a reduction in the cost of establishing a network with fewer navigation agents. However, it will take time and effort to determine the longest distance, detour metric dimension, and bi-metric dimension of the graph in a large graph with many vertices and edges. Computational tools are needed to solve these three problems.

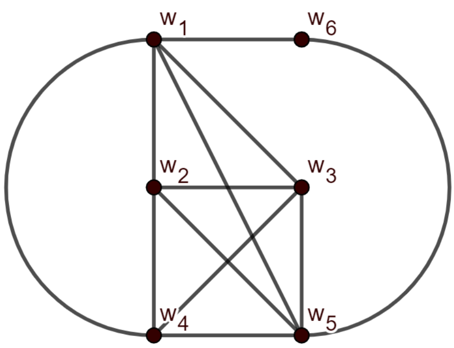

Figure 4. **graph H having a bi-metric dimension less than its metric dimension**

3. Ant Colony Optimization Algorithm and Gray Wolf Optimization Algorithm

3.1. The Ant Colony Optimization (ACO)

The Ant Colony Optimization (ACO) algorithm is probabilistic, inspired by the foraging behavior of an ant species. Because ants are blind, they move randomly from one place to another and choose routes based on probability. When traveling, ants release certain amounts of pheromones. This will mark the route for other ants to follow. When other ants are looking for food, most of the other ants will follow the route with more pheromones until all the ants finally choose to take the shortest route to reach the food source. This mechanism of ants is modeled in a mathematical formulation. Ants select the following city to be visited through a stochastic mechanism. When ant k is in the city i and has constructed the partial solution s^p. The probability of going to city j is given by:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{C_{il}\in N(s^P)} \tau_{il}^{\alpha}\eta_{il}^{\beta}} & \text{if } C_{ij} \in N(s^P), \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

$N(s^p)$ represents the collection of feasible components, referring to edges (i, l) where l signifies a city that the ant k has not yet visited. The values of α and β dictate the significance between the pheromone levels and the heuristic information $\eta_{i,j}$, which is given by:

$$\eta_{i,j} = \frac{1}{d_{i,j}}, \tag{3.2}$$

$d_{i,j}$ represents the distance measurement between cities i and j.
The pheromone $\tau_{i,j}$, linked with the edge joining cities i and j, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{ij}^k, \tag{3.3}$$

Here, ρ represents the evaporation rate, m is the total number of ants, and $\Delta\tau_{ij}^k$ indicates the amount of pheromone deposited on edge (i, j) by ant indexed as k:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

In this context, Q denotes a constant, and L_k represents the route length created by ant k [1].

3.2. Gray Wolf Optimization Algorithm

The GWO algorithm was developed based on the hunting patterns of a group of gray wolves. These animals tend to favor residing in communal settings or groups. The first level is the alpha wolf, which contains a male or female wolf who is the leader of the gray wolf pack. Alpha is accountable for making choices regarding hunting, selecting a sleeping location, determining wake-up times, and similar considerations. Alpha wolves are also called dominant wolves because the pack must follow their orders. Beta is the subsequent tier within the gray wolf pack's social structure. Betas are alpha subordinates who help alphas in decision-making or other behavioral activities. The lowest level in the gray wolf hierarchy is omega. Omega wolves are always submissive to all the other dominant wolves. Omega is not an essential individual in the group. If a wolf in a pack is not alpha, beta, or omega, or the wolf dominates the omega wolf, then this wolf is called a delta. Delta wolves are required to yield to the alpha and beta members, but they dominate omega wolves. When designing the GWO algorithm, The fittest solution is α (alpha). The second fittest solution is β (beta), and the third is δ (delta). Apart from the three types above, gray wolves are classified as ω (omega). [25].

A pack of gray wolves surrounds their prey in the process of hunting. In mathematical modeling, this process is written in the form of the subsequent equation:

$$\vec{D} = \mid \vec{C}.\vec{X}_p(t) - \vec{X}(t) \mid \tag{3.5}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{3.6}$$

In this context, t represents the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors. $\vec{X}_p$ denotes the position vector of the prey, $\vec{X}$ denotes the position vector of the gray wolf. The vector $\vec{A}$ and vector $\vec{C}$ are determined based on the following formula:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{3.7}$$

$$\vec{C} = 2\vec{r_2} \tag{3.8}$$

Here, $\vec{a}$ linearly decreases from 2 to 0 throughout the iteration process, and $\vec{r_1}$ and $\vec{r_2}$ represent random vectors within the range of [0,1].

In mathematically simulating the hunting behavior of the gray wolf, alpha (the most optimal candidate solution), beta, and delta had better knowledge of the prospective position of the prey. Hence, we preserve the initial three most optimal solutions acquired and then require the other search wolves (incorporating the omega members) to adjust their positions according to the positions of the most proficient searching wolves. Formulas (3.9)-(3.15) represent this hunting activity.

$$\vec{D_\alpha} = \mid \vec{C_1}.\vec{X_\alpha} - \vec{X} \mid, \tag{3.9}$$
$$\vec{D_\beta} = \mid \vec{C_2}.\vec{X_\beta} - \vec{X} \mid, \tag{3.10}$$
$$\vec{D_\delta} = \mid \vec{C_3}.\vec{X_\delta} - \vec{X} \mid \tag{3.11}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1}.(\vec{D_\alpha}), \tag{3.12}$$
$$\vec{X_2} = \vec{X_\beta} - \vec{A_2}.(\vec{D_\beta}), \tag{3.13}$$
$$\vec{X_3} = \vec{X_\delta} - \vec{A_3}.(\vec{D_\delta}) \tag{3.14}$$

$$\vec{X}_{(t+1)} = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{3.15}$$

4. The Proposed ACO-BGWO Algorithm

4.1. BGWO Algorithm for Metric Dimension Problem

In the gray wolf optimization (GWO) algorithm, wolves consistently alter their positions to any location within the space. In metric dimension problems, the solutions are confined within the binary set $\{0, 1\}$. In this study, the modification of binary GWO algorithm from [9] proposed for the MDP also detour metric dimension and bi-metric dimension.

Within the BGWO approach, the primary updating equation is expressed as illustrated in equation (4.1); refer to Algorithm 1 for further details.

$$X_i^{t+1} = Crossover(x_1, x_2, x_3) \tag{4.1}$$

Here, $Crossover(x_1, x_2, x_3)$ 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 (4.2),(4.5), and (4.8) correspondingly.

$$x_1^d = \begin{cases} 1 & if(x_\alpha^d + bstep_\alpha^d) \geq 1 \\ 0 & otherwise \end{cases} \tag{4.2}$$

Here, x_α^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 (4.3).

$$bstep_\alpha^d = \begin{cases} 1 & if cstep_\alpha^d \geq rand \\ 0 & otherwise \end{cases} \tag{4.3}$$

Here, $rand$ is a randomly generated number from a uniform distribution within the $[0, 1]$ range. $cstep_\alpha^d$ stands for the continuous-valued step size for dimension d and can be computed using a sigmoidal function as outlined in equation (4.4).

$$cstep_\alpha^d = \frac{1}{1 + e^{-10(A_1^d D_\alpha^d - 0.5)}} \tag{4.4}$$

Here, A_1^d and D_α^d are determined by equations (3.7), and (3.12) in the dimension d.

$$x_2^d = \begin{cases} 1 & \text{if } (x_\beta^d + bstep_\beta^d) \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

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 (4.6).

$$bstep_\beta^d = \begin{cases} 1 & \text{if } cstep_\beta^d \geq rand \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Here, the $rand$ denotes a randomly generated number from a uniform distribution within the range $\in [0, 1]$, 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 (4.7).

$$cstep_\beta^d = \frac{1}{1 + e^{-10(A_1^d D_\beta^d - 0.5)}} \tag{4.7}$$

Here, A_1^d and D_β^d are computed using equations (3.7) and (3.12) in the dimension d.

$$x_3^d = \begin{cases} 1 & \text{if } (x_\delta^d + bstep_\delta^d) \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Here 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 (4.9).

$$bstep_\delta^d = \begin{cases} 1 & \text{if } cstep_\delta^d \geq rand \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

Where $rand$ is a random number drawn from the uniform distribution $\in [0,1]$, and $cstep_\delta^d$ is the continuous-valued step size for dimension d and can be calculated using sigmoidal function as in equation (4.10).

$$cstep_\delta^d = \frac{1}{1+e^{-10(A_1^d D_\delta^d - 0.5)}}. \tag{4.10}$$

A stochastic crossover strategy is implemented per dimension to perform the crossover operation among the solutions a, b, c solutions, as demonstrated in equation (4.11).

$$x_d = \begin{cases} a_d & \text{if } rand < 1 \\ b_d & \text{if } \frac{1}{3} \leq rand < \frac{2}{3} \\ c_d & \text{otherwise.} \end{cases} \tag{4.11}$$

Here, a_d, b_d, c_d represent the binary values for the first, second, and third parameter 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]. The procedure of the proposed BGWO algorithm, as seen in Algorithm 1.

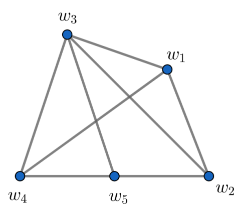

Figure 5. **Graph** A_3^3

To illustrate, consider a graph A_3^3 with $V(A_3^3) = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ in Figure 5. Graph A_3^3 will initially be transformed into an adjacent matrix denoted by M, facilitating computational processing. For this purpose, the elements on the diagonal are 0; if two vertices are adjacent, the value is 1. If two vertices are not adjacent, then the value is 0.

Algorithm 1 BGWO Algorithm for Metric Dimension Problem

Input: n number of nodes, distance matrix, popsize, MaxIt
Output: x_α optimal gray wolf, Metric dimension

1. Initialize a population of a popsize wolves positions at random $\in [0, 1]$
2. Find the representation of every node with respect to wolves
3. Replace the population based on objective value
4. Find the α, β, and δ based on fitness
5. **while** stopping criteria not met **do**

foreach $wolf_i \in pack$ **do**

Calculate x_1, x_2, x_3 using equations (4.2), (4.5), and (4.8)
$x_i^{t+1} \leftarrow$ crossover among x_1, x_2, x_3 using equation (4.11)

end (for)

I. Update a, A, C
II. Evaluate the positions of individual wolves
III. Update α, β, and δ
end (while)

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Suppose we aim to determine the metric dimension of graph A_3^3. First, we apply the Floyd-Warshall algorithm to obtain a distance matrix of A_3^3. We skip the detailed step and present the distance matrix of A_3^3 as D.

$$D = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 0 & 2 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 0 & 1 & 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 2 \\ 1 & 2 & 1 & 1 & 1 & 0 & 2 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 2 & 0 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 1 \\ 2 & 2 & 2 & 1 & 2 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Let the number of gray wolves in the pack be $n = 5$, and the number of iterations for optimization is $N_{iter} = 10$. The α is the optimal gray wolf binary position: minimum resolving set, and $f(\alpha)$ is the best fitness value: metric dimension.

Step 1

Initialize a (candidate) population of n wolves position at random $\in [0, 1]$. Suppose that :

$$\text{Wolf } 1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$
$$\text{Wolf } 2 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$
$$\text{Wolf } 3 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$
$$\text{Wolf } 4 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$
$$\text{Wolf } 5 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The following presents a method for determining the fitness value of each wolf.

Wolf 1 $\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$

From the vector generated at Wolf 1, the (candidate) resolving set is $S_1 = \{v_1, v_3, v_4, v_7, v_8, v_9\}$. In step 1, Determine the representation of all vertices in the graph A_3^3 with respect to S_1. Computationally, the elements in the representation vectors are taken from the elements of D.

$$\begin{aligned}
r(v_1|S_1) &= (0, 1, 1, 2, 2, 2),\\
r(v_2|S_1) &= (1, 1, 1, 2, 2, 2),\\
r(v_3|S_1) &= (1, 0, 2, 2, 2, 2),\\
r(v_4|S_1) &= (1, 2, 0, 1, 2, 1),\\
r(v_5|S_1) &= (2, 1, 1, 1, 1, 2).\\
r(v_6|S_1) &= (1, 1, 1, 2, 1, 1),\\
r(v_7|S_1) &= (2, 2, 1, 0, 1, 1),\\
r(v_8|S_1) &= (2, 2, 2, 1, 0, 1),\\
r(v_9|S_1) &= (2, 2, 1, 1, 1, 0).
\end{aligned}$$

Step 2

Sort the representation vectors generated in Step 1 according to the lexicographic order. The sorted result for the representation of wolf 1 is as follows:

$$\begin{aligned}
r(t_1|S_1) = r(v_1|S_1) &= (0, 1, 1, 2, 2, 2),\\
r(t_2|S_1) = r(v_3|S_1) &= (1, 0, 2, 2, 2, 2),\\
r(t_3|S_1) = r(v_6|S_1) &= (1, 1, 1, 2, 1, 1),\\
r(t_4|S_1) = r(v_2|S_1) &= (1, 1, 1, 2, 2, 2),\\
r(t_5|S_1) = r(v_4|S_1) &= (1, 2, 0, 1, 2, 1).\\
r(t_6|S_1) = r(v_5|S_1) &= (2, 1, 1, 1, 1, 2),\\
r(t_7|S_1) = r(v_7|S_1) &= (2, 2, 1, 0, 1, 1),\\
r(t_8|S_1) = r(v_9|S_1) &= (2, 2, 1, 1, 1, 0),\\
r(t_9|S_1) = r(v_8|S_1) &= (2, 2, 2, 1, 0, 1).
\end{aligned}$$

Table 1. **The results of steps 1-3 for wolves 1-5.**

wolf	S_i	q	S_i^*	$wolf(i)^*$
1	$S_1 = \{v_1, v_3, v_4, v_7, v_8, v_9\}$	5	$S_1^* = \{v_1, v_3, v_4, v_7, v_8\}$	wolf 1*= [1 0 1 1 0 0 1 1 0]
2	$S_2 = \{v_1, v_2, v_4, v_6, v_7\}$	4	$S_2^* = \{v_1, v_2, v_4, v_6\}$	wolf 2*= [1 1 0 1 0 1 0 0 0]
3	$S_3 = \{v_1, v_4, v_5, v_6, v_9\}$	4	$S_3^* = \{v_1, v_4, v_5, v_6\}$	wolf 3*= [1 0 0 1 1 1 0 0 0]
4	$S_4 = \{v_2, v_4, v_6, v_8, v_9\}$	4	$S_4^* = \{v_2, v_4, v_6, v_8\}$	wolf 4*= [0 1 0 1 0 1 0 1 0]
5	$S_5 = \{v_1, v_3, v_4, v_5, v_9\}$	5	$S_5^* = S_5$	wolf 5*=wolf 5

Next, for $i = 1, 2, ..., (n-1)$ find $j(i)$ as the minimal coordinate where the vectors $r(t_i, S)$ and $r(t_{i+1}, S)$ are different. If such coordinate does not exist for some index i, i.e. $r(t_i, S) = r(t_{i+1}, S)$, then S is not a resolving set, and the procedure stops. From the calculation of the value of $j(i)$ at wolf 1: $j(1) = 1, j(2) = 2, j(3) = 5, j(4) = 2, j(5) = 1, j(6) = 2, j(7) = 4, j(8) = 3$.

Step 3

Determine $q = max\{j(1), j(2), ..., j(n-1)\}$, and keep in S only the first q elements. [23]. For wolf 1: $q = max\{1, 2, 5, 2, 1, 2, 4, 3\} = 5$, thus $S_1* = \{v_1, v_3, v_4, v_7, v_8\}$, and Wolf 1* $= \begin{bmatrix}1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0\end{bmatrix}$ with the fitness value equal to the number of element 1 for a wolf 1* = 5. Table 1 presents the steps 1-3 results for wolves 1-5. Table 1 presents the results of steps 1-3 for wolves 1-5.
Thus, the initial population of gray wolves is:

$$\text{Wolf } 1^* = \begin{bmatrix}1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0\end{bmatrix}$$

$$\text{Wolf } 2^* = \begin{bmatrix}1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

$$\text{Wolf } 3^* = \begin{bmatrix}1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}$$

$$\text{Wolf } 4^* = \begin{bmatrix}0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0\end{bmatrix}$$

$$\text{Wolf } 5^* = \begin{bmatrix}1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1\end{bmatrix}$$

Based on the fitness values, wolves 2,3,4 have the same fitness value = 4. This value is the smallest. Alpha is chosen randomly from the three wolves, as are beta and omega.

$$\alpha = \begin{bmatrix}1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

$$\beta = \begin{bmatrix}1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}$$

$$\delta = \begin{bmatrix}0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0\end{bmatrix}$$

Next, we apply equations (3.7) until (3.14).

For t=1
$a = 2 - 2.\frac{1}{3} = 2 - \frac{2}{3} = \frac{4}{3} = 1.3333.$

Suppose that:

$$\vec{r_1} = \begin{bmatrix} 0.5392 \\ 0.2111 \\ 0.6792 \\ 0.5733 \\ 0.8245 \\ 0.9713 \\ 0.5324 \\ 0.3333 \\ 0.2178 \end{bmatrix} \quad \vec{r_2} = \begin{bmatrix} 0.9211 \\ 0.5439 \\ 0.6745 \\ 0.5556 \\ 0.4321 \\ 0.5543 \\ 0.7214 \\ 0.8151 \\ 0.6253 \end{bmatrix}$$

then:

$$\vec{A} = 2a\vec{r_1} - a = 2 \begin{bmatrix} 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \end{bmatrix} \circ \begin{bmatrix} 0.5392 \\ 0.2111 \\ 0.6792 \\ 0.5733 \\ 0.8245 \\ 0.9713 \\ 0.5324 \\ 0.3333 \\ 0.2178 \end{bmatrix} - \begin{bmatrix} 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \\ 1.3333 \end{bmatrix} = \begin{bmatrix} 0.1045 \\ -0.7704 \\ 0.4779 \\ 0.1955 \\ 0.8648 \\ 1.2568 \\ 0.0864 \\ -0.4445 \\ -0.7525 \end{bmatrix}$$

Operation $\circ$ is a Hadamard (Schur) product.

$$\vec{C} = 2\vec{r_2} = 2 \begin{bmatrix} 0.9211 \\ 0.5439 \\ 0.6745 \\ 0.5556 \\ 0.4321 \\ 0.5543 \\ 0.7214 \\ 0.8151 \\ 0.6253 \end{bmatrix} = \begin{bmatrix} 1.8422 \\ 1.0878 \\ 1.3490 \\ 1.1112 \\ 0.8642 \\ 1.1086 \\ 1.4428 \\ 1.6302 \\ 1.2506 \end{bmatrix}$$

$\vec{C_1} = \vec{C_2} = \vec{C_3} = \vec{C}.$

Next, $\vec{D_\alpha} = \vec{C_1} \circ \vec{X_\alpha} - \vec{X} = \begin{bmatrix} 1.8422 \\ 1.0878 \\ 1.3490 \\ 1.1112 \\ 0.8642 \\ 1.1086 \\ 1.4428 \\ 1.6302 \\ 1.2506 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8422 \\ 0.0878 \\ -1 \\ 0.1112 \\ 0 \\ 1.1086 \\ -1 \\ -1 \\ 0 \end{bmatrix}$,

$$\vec{D_\beta} = \vec{C_2} \circ \vec{X_\beta} - \vec{X} = \begin{bmatrix} 1.8422 \\ 1.0878 \\ 1.3490 \\ 1.1112 \\ 0.8642 \\ 1.1086 \\ 1.4428 \\ 1.6302 \\ 1.2506 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8422 \\ 0 \\ -1 \\ 0.1112 \\ 0.8642 \\ 1.1086 \\ -1 \\ -1 \\ 0 \end{bmatrix},$$

$$\vec{D_\delta} = \vec{C_3} \circ \vec{X_\delta} - \vec{X} = \begin{bmatrix} 1.8422 \\ 1.0878 \\ 1.3490 \\ 1.1112 \\ 0.8642 \\ 1.1086 \\ 1.4428 \\ 1.6302 \\ 1.2506 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1.0878 \\ -1 \\ 0.1112 \\ 0 \\ 1.1086 \\ -1 \\ 0.6302 \\ 0 \end{bmatrix},$$

Next, we apply equations (4.2) until (4.11).

For d=1

$cstep_\alpha^1 = \frac{1}{1+e^{-10(0.1045.0.8422-0.5)}} = 0.0159.$

A random number is generated. For example, $r = 0.8147$. Because of $cstep_\alpha^1 = 0.0159 < 0.8147$, then $bstep_\alpha^1 = 0$.
Because of $x_\alpha^1 + bstep_\alpha^1 = 1 + 0 = 1 \geq 1$, then $x_1^1 = 1$.

The calculation continues until $d = n$, and similar way for β and δ.

Table 2. Result of BGWO Algorithm for Metric Dimension on some Generalized Antiprism Graphs.

graph	n	m	popsize	maxIt	resolving set	*dim*
A_3^3	9	21	20	50	$\{a_2, b_1, c_1\}$	3
A_3^4	12	30	20	50	$\{a_2, a_3, c_1, c_3\}$	4
A_3^5	15	39	25	50	$\{a_3, b_3, c_3, d_1, e_1\}$	5
A_3^6	18	48	25	50	$\{a_1, a_2, c_3, d_3, e_2, e_3\}$	6
A_3^7	21	57	25	50	$\{b_1, b_2, d_2, e_2, f_2, f_3\}$	7
A_3^8	24	66	30	75	$\{a_1, a_2, c_2, d_1, d_2, f_3, g_1, g_2\}$	8
A_3^9	27	75	50	150	$\{a_1, b_3, d_1, d_2, e_1, f_2, g_2, h_1, h_2\}$	9
A_3^{10}	30	84	75	200	$\{a_1, b_1, d_1, d_3, e_3, f_2, h_3, i_2, i_3\}$	9
A_4^3	12	28	20	50	$\{a_2, a_4, b_1\}$	3
A_4^4	16	40	20	50	$\{a_1, b_1, b_3, b_4\}$	4
A_4^5	20	52	35	100	$\{a_4, b_2, b_3, b_4\}$	4
A_4^6	24	64	50	150	$\{a_3, b_2, d_1, d_3, d_4\}$	5
A_4^7	28	76	70	200	$\{a_4, b_2, d_1, e_1, e_4, f_1\}$	6
A_4^8	32	88	80	300	$\{a_1, b_3, c_1, c_4, f_3, f_4, g_2\}$	7
A_4^9	36	100	90	400	$\{a_4, c_3, c_4, e_1, e_3, f_4, g_1, h_1, h_2\}$	9
A_4^{10}	40	112	90	400	$\{a_1, a_4, b_1, c_1, e_1, e_3, f_4, h_3, h_4, i_2\}$	10

4.2. ACO-BGWO Algorithm for Detour Metric Dimension Problem

In this section, we proposed a new hybrid algorithm, ACO-BGWO, which combines ACO and BGWO techniques. First, ACO is used to search for the detour distance of two vertices in a graph (see algorithm 2). Secondly, we modified the BGWO algorithms to find the detour metric dimension of the graph. The main structure of the BGWO algorithm is presented in algorithm 3.
As an illustration, suppose the results of determining the detour distance from every two points on the graph A_3^3 are presented in the matrix Δ.

$$\Delta = \begin{bmatrix} 0 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 0 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 8 & 0 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 8 & 8 & 0 & 8 & 8 & 8 & 8 & 8 \\ 8 & 8 & 8 & 8 & 0 & 8 & 8 & 8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 0 & 8 & 8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & 0 & 8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 0 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 0 \end{bmatrix}$$

This Δ matrix replaces the distance matrix D in determining the detour metric dimension in algorithm 3.

Algorithm 2 ACO Algorithm for Detour Distance Problem

Input: n number of nodes, adjacency matrix, maxAnt, MaxIt, startNode, endNode
Output: longest route, detour distance Initialize stage

foreach i = 1 : maxAnt **do**
$P(i,j) = P_0(i,j) = 1$
find visibility $\eta_{(i,j)} = h.d_{(i,j)}$
where $h = \begin{cases} \frac{1}{2} & \text{if } j \text{ adjacent to endNode} \\ \frac{1}{4} & \text{if } j \text{ is endNode} \\ 0 & \text{if } j \text{ has been passed} \\ 1 & \text{otherwise} \end{cases}$, and $d_{i,j}$ is element of adjacency matrix
end (for)
while it ¡ maxIt **do**
place Ant on startNode
foreach i = 1 : maxAnt
Start from startNode
Calculate the pheromone level for each edge which is connected with startNode
Calculate the transition probability from a starNode to another node
Generate random numbers
Use roulette wheel selection to select the nextNode
end (for)
foreach i = maxAnt **do** calculate x = total route distance generated by each Ant
Update pheromone matrix $P_k(i,j) = 0.5 * P_{k-1}(i,j) + x$
end (for)
end (while)

Algorithm 3 BGWO Algorithm for Detour Metric Dimension Problem

Input: n number of nodes, detour distance matrix, popsize, MaxIt
Output: x_α optimal gray wolf, Detour metric dimension

1. Initialize a population of a popsize wolves positions at random $\in [0,1]$
2. Determine the representation of each node for each gray wolf based on the detour distance matrix
3. Replace the population based on objective value
4. Find the α, β, and δ based on fitness
5. **while** stopping criteria not met **do**

foreach $wolf_i \in pack$ **do**

Calculate x_1, x_2, x_3 using equations (4.2), (4.5), and (4.8)
$x_i^{t+1} \leftarrow$ crossover among x_1, x_2, x_3 using equation (4.11)

end(for)

I. Update a, A, C

II. Evaluate the positions of individual wolves
III. Update α, β, and δ
end(while)

Table 3. Result of the ACO-BGWO Algorithm for Detour Metric Dimension on some Cycle Graphs.

graph	n	m	popsize	maxIt	detour resolving set	$D\beta(G)$
C_8	8	8	20	50	$\{a_2, a_3\}$	2
C_9	9	9	20	50	$\{a_6, a_9\}$	2
C_{10}	10	10	30	100	$\{a_4, a_7\}$	2
C_{11}	11	11	30	100	$\{a_4, a_8\}$	2
C_{12}	12	12	30	100	$\{a_3, a_5\}$	2
C_{13}	13	13	30	100	$\{a_7, a_8\}$	2
C_{14}	14	14	40	150	$\{a_3, a_{13}\}$	2
C_{15}	15	15	40	150	$\{a_{11}, a_{14}\}$	2

Table 4. Result of the ACO-BGWO Algorithm for Detour Metric Dimension on some Jahangir Graphs.

graph	n	m	popsize	maxIt	$D\beta(G)$
$J_{2,5}$	11	15	50	300	9
$J_{2,6}$	13	18	50	300	11
$J_{2,7}$	15	21	50	300	13
$J_{2,8}$	17	24	50	300	15
$J_{2,9}$	19	27	50	300	17
$J_{2,10}$	21	30	50	300	19
$J_{2,11}$	23	33	50	300	21
$J_{2,12}$	25	36	50	300	23

Table 5. Result of the ACO-BGWO Algorithm for Detour Metric Dimension on some Friendship Graphs.

graph	n	m	popsize	maxIt	$D\beta(G)$
F_8	17	24	50	400	17
F_9	19	27	50	400	19
F_{10}	21	30	50	400	21
F_{11}	23	33	50	400	23
F_{12}	25	36	50	400	25
F_{13}	27	39	50	400	27
F_{14}	29	42	50	400	29
F_{15}	31	45	50	400	31

4.3. ACO-BGWO Algorithm for Bi-metric Dimension Problem

In this section, first, ACO is used to search for the detour distance of two vertices in a graph (see Algorithm 2). Secondly, we modified the BGWO algorithms to find the bi-metric dimension of the graph. The main structure of the BGWO algorithm is presented in algorithm 4. In Algorithm 4, the distance matrix D and the detour distance matrix Δ are considered as inputs in the bi-metric dimension search.

Algorithm 4 BGWO Algorithm for Bi-metric Dimension Problem

Input: n number of nodes, distance matrix, detour distance matrix, popsize, MaxIt
Output: x_α optimal gray wolf, Bi-metric dimension

1. Initialize a population of a popsize wolves positions at random $\in [0, 1]$
2. Determine the representation of each node for each gray wolf based on the union matrix of the distance matrix and the detour distance matrix
3. Replace the population based on objective value
4. Find the α, β, and δ based on fitness
5. **while** stopping criteria not met **do**

foreach $wolf_i \in pack$ **do**

Calculate x_1, x_2, x_3 using equations (4.2), (4.5), and (4.8)
$x_i^{t+1} \leftarrow$ crossover among x_1, x_2, x_3 using equation (4.11)

end (for)

I. Update a, A, C
II. Evaluate the positions of individual wolves
III. Update α, β, and δ
end (while)

5. Results and Discussions

First, we give an upper bound on the metric dimension of the graph A_3^3.

Theorem 1. $Dim(A_3^n) \leq n$

Proof. Suppose $n = 3$, and $V(A_3^3) = \{a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3\}$

Since A_3^3 is not a path, then $dim(A33) \geq 2$. Next, there are five possibilities of resolving sets with cardinality 2 as follows:

Suppose $S_1 = \{a_1, a_2\}$. Because $r(a_3 \mid \{a_1, a_2\}) = (1, 1) = r(b_1 \mid \{a_1, a_2\})$, then S_1 is not a resolving set.

Table 6. Result of the ACO-BGWO Algorithm for Bi-metric Dimension on some Complete Graphs and Generalized Antiprism Graphs.

graph	n	m	popsize	maxIt	$\beta_b(G)$
K_8	8	28	50	500	7
K_9	9	36	50	500	8
K_{10}	10	45	50	500	9
K_{11}	11	55	50	500	10
K_{12}	12	66	50	500	11
A_3^3	9	21	30	400	3
A_4^3	12	30	30	400	4
A_5^3	15	39	30	400	5
A_6^3	18	48	50	400	6
A_7^3	21	57	50	400	7

Suppose $S_2 = \{b_1, b_2\}$. Because $r(b_2 \mid \{b_1, b_2\}) = (2, 1) = r(c_1 \mid \{b_1, b_2\})$, then S_2 is not a resolving set.

Suppose $S_3 = \{a_1, b_1\}$. Because $r(b_2 \mid \{b_1, b_2\}) = (1, 1) = r(c_1 \mid \{b_1, b_2\})$, then S_3 is not a resolving set.

Suppose $S_4 = \{a_1, b_2\}$. Because $r(a_2 \mid \{a_1, b_2\}) = (1, 1) = r(a_3 \mid \{a_1, b_2\})$, then S_4 is not a resolving set.

Suppose $S_5 = \{a_1, c_1\}$. Because $r(a_2 \mid \{a_1, c_1\}) = (1, 2) = r(a_3 \mid \{a_1, c_1\})$, then S_5 is not a resolving set.

Now consider the set $S = \{a_1, b_1, c_3\}$. The representation of each vertex in A_3^3 are as follows:

$$\begin{aligned}
r(a_1 \mid \{a_1, b_1, c_3\}) &= (0, 1, 2),\\
r(a_2 \mid \{a_1, b_1, c_3\}) &= (1, 1, 2),\\
r(a_3 \mid \{a_1, b_1, c_3\}) &= (1, 2, 2),\\
r(b_1 \mid \{a_1, b_1, c_3\}) &= (1, 0, 1),\\
r(b_2 \mid \{a_1, b_1, c_3\}) &= (2, 1, 2),\\
r(b_3 \mid \{a_1, b_1, c_3\}) &= (1, 1, 1),\\
r(c_1 \mid \{a_1, b_1, c_3\}) &= (2, 1, 1),\\
r(c_2 \mid \{a_1, b_1, c_3\}) &= (2, 2, 1),\\
r(c_3 \mid \{a_1, b_1, c_3\}) &= (2, 1, 0).
\end{aligned}$$

Thus S is a basis of A_3^3 and $dim(A_3^3) = 3$.
Suppose $n > 3$, and $V(A_3^3 = \{a_1^1, a_1^2, a_1^3, a_2^1, a_2^2, a_2^3, \ldots, a_3^1, a_3^2, a_3^3\}$. Choose $S_k = a_1^1, a_2^1, a_3^3, ..., a_n^3 \subseteq V(A_3^3)$. The representation of each vertex in A_3^3 are as follows:

$r(a_1^1 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (0, 1, 2, \ldots, n-1),$
$r(a_1^2 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (1, 1, 2, \ldots, n-1),$
$r(a_1^3 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (1, 2, 2, \ldots, n-1),$
$r(a_2^1 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (1, 0, 1, \ldots, n-2),$
$r(a_2^2 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (2, 1, 2, \ldots, n-2),$
$r(a_2^3 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (1, 1, 1, \ldots, n-2),$
$r(a_3^1 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (2, 2, 1, \ldots, n-3),$
$r(a_3^2 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (2, 1, 1, \ldots, n-2),$
$r(a_3^3 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (2, 1, 0, \ldots, n-3),$
$\vdots$
$r(a_n^1 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (n-1, n-2, \ldots, 2, 2, 1),$
$r(a_n^2 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (n-1, n-2, \ldots, 2, 1, 1),$
$r(a_n^3 \mid \{a_1^1, a_2^1, a_3^3, \ldots, a_n^3\}) = (n-1, \ldots, 2, 1, 0).$
Because of all the representation are different, thus S_k is a basis of A_3^n and $dim(A_3^n) \leq n$. □

In this section, these experiments have been run using Matlab R2023b installed on Windows 10 Pro, which runs on a Core i5 and 16 GB RAM. The BGWO Algorithm for the metric dimension of generalized antiprism graphs is given in Table 2. This table shows that the BGWO Algorithm gives better results than the upper bound from Theorem 1. From the simulation result, the $dim(A_3^{10}) = 9$ is less than that upper bound (=10). To solve the detour metric dimension, we apply the hybrid ACO-BGWO, namely algorithm 2 and algorithm 3. The hybrid ACO-BGWO is simulated for several family graphs: cycle, Jahangir, and friendship. The results are presented in tables 3, 4, and 5. This simulation shows that the hybrid ACO-BGWO algorithm gives the same result as in the literature. To solve the bi-metric dimension, we apply the hybrid ACO-BGWO, namely Algorithm 2 and Algorithm 4. The hybrid ACO-BGWO is simulated for several family graphs: complete and generalized antiprism graphs. The results are presented in tables 6. This simulation results show that, for some complete graphs, the hybrid algorithm gives the same result as in the literature. For the generalized antiprism graphs, the bi-metric dimension is the same as its metric dimension.

6. Conclusion

In this work, we apply the BGWO algorithm in generalized antiprism graphs to find the metric dimension. The results are better than the upper bound of its metric dimension. Also, we construct a hybrid algorithm of ACO-BGWO for detour metric dimension, as shown by Algorithm 2 and Algorithm 3. The results on the graph are shown in Table 3, 4, and 5. This hybrid algorithm gives the same result as in the literature. In addition, we construct a hybrid algorithm of ACO-BGWO for bi-metric dimension, as shown by Algorithm 2 and Algorithm 4. This hybrid algorithm gives the same result as in the literature and shows that the bi-metric dimension of the generalized antiprism graph is the same as its metric dimension. This result confirms the capabilities of the hybrid ACO-BGWO algorithm in solving the problems.

Acknowledgements: The author would like to thank the Directorate of Higher Education, Ministry of Education and Culture, for supporting this research with the BPI doctoral research scholarship.

Conflict of Interest: The authors declare that they have no conflict of interest.

Data Availability: Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

References

[1] *Ant colony optimization: A]rtificial Ants as a Computational Intelligence Technique, author=Dorigo, M. and Birattari, M. and Stutzle, T., journal=IEEE Comput. Intell. Mag., volume=1, number=4, pages=Article ID: 28, year=2006.*

[2] S. Akhter and R. Farooq, *Metric dimension of fullerene graphs.*, Electron. J. Graph Theory Appl. **7** (2019), no. 1, 91–103.
https://doi.org/10.5614/ejgta.2019.7.1.7.

[3] J. Bensmail, F.M. Inerney, and N. Nisse, *Metric dimension: From graphs to oriented graphs*, Discrete Appl. Math. **323** (2022), 28–42.
https://doi.org/10.1016/j.entcs.2019.08.011.

[4] G. Chartrand, L. Eroh, M.A. Johnson, and O.R. Oellermann, *Resolvability in graphs and the metric dimension of a graph*, Discrete Appl. Math. **105** (2000), no. 1-3, 99–113.
https://doi.org/10.1016/S0166-218X(00)00198-0.

[5] ———, *Resolvability in graphs and the metric dimension of a graph*, Discrete Appl. Math. **105** (2000), no. 1-3, 99–113.
https://doi.org/10.1016/S0166-218X(00)00198-0.

[6] K. Chau and S. Gosselin, *The metric dimension of circulant graphs and their Cartesian products*, Opuscula Math. **37** (2017), no. 4, 509–534.
http://dx.doi.org/10.7494/OpMath.2017.37.4.509.

[7] V.K. Devi and K. Marimuthu, *A new encryption technique using detour metric dimension*, J. Infor. Math. Sci. **9** (2017), no. 3, 949–955.
https://doi.org/10.26713/jims.v9i3.1015.

[8] F. El Asri, C. Tajani, and H. Fakhouri, *A combined ant colony optimization with levy flight mechanism for the probabilistic traveling salesman problem with deadlines*, Math. Model. Comput. **11** (2024), no. 1, 290–299.
https://doi.org/10.23939/mmc2024.01.290.

[9] E. Emary, H.M. Zawbaa, and A.E. Hassanien, *Binary grey wolf optimization approaches for feature selection*, Neurocomputing **172** (2016), 371–381.
https://doi.org/10.1016/j.neucom.2015.06.083.

[10] L. Eroh, C.X. Kang, and E. Yi, *On metric dimension of functigraphs*, Discrete Math. Algorithms Appl. **5** (2013), no. 04, Article ID: 1250060. https://doi.org/10.1142/S1793830912500607.

[11] M. Fehr, S. Gosselin, and O.R. Oellermann, *The metric dimension of Cayley digraphs*, Discrete Math. **306** (2006), no. 1, 31–41. https://doi.org/10.1016/j.disc.2005.09.015.

[12] M. Feng and K. Wang, *On the metric dimension of bilinear forms graphs*, Discrete Math. **312** (2012), no. 6, 1266–1268. https://doi.org/10.1016/j.disc.2011.11.020.

[13] M. Feng, M. Xu, and K. Wang, *On the metric dimension of line graphs*, Discrete Appl. Math. **161** (2013), no. 6, 802–805. https://doi.org/10.1016/j.dam.2012.10.018.

[14] J. Guo, K. Wang, and F. Li, *Metric dimension of some distance-regular graphs*, J. Comb. Optim. **26** (2013), no. 1, 190–197. https://doi.org/10.1007/s10878-012-9459-x.

[15] F. Harrary and R.A. Melter, *On the metric dimension of a graph*, Ars Combin. **2** (1976), 191–195.

[16] J.T. Hedetniemi, S.T. Hedetniemi, R.C. Laskar, and M. Mulder, *The 2-dimension of a tree*, Commun. Comb. Optim. **5** (2020), no. 1, 69–81. https://doi.org/10.22049/cco.2019.26495.1119.

[17] H. Hendy, M.I. Irawan, I. Mukhlash, and S. Setumin, *A bibliometric analysis of metaheuristic research and its applications*, Regist.: J. Ilm. Teknol. Sist. Inf. **9** (2023), no. 1, 1–17. http://doi.org/10.26594/register.v9i1.2675.

[18] C. Hernando, M. Mora, I.M. Pelayo, C. Seara, J. Cáceres, and M.L. Puertas, *On the metric dimension of some families of graphs*, Electron. Notes Discrete Math. **22** (2005), 129–133. https://doi.org/10.1016/j.endm.2005.06.023.

[19] M. Imran, A.Q. Baig, S.A.U.H. Bokhary, and I. Javaid, *On the metric dimension of circulant graphs*, Appl. Math. Lett. **25** (2012), no. 3, 320–325. https://doi.org/10.1016/j.aml.2011.09.008.

[20] M. Imran, M.K. Siddiqui, and R. Naeem, *On the metric dimension of generalized Petersen multigraphs*, IEEE access **6** (2018), 74328–74338. https://doi.org/10.1109/ACCESS.2018.2883556.

[21] S. Imran, M.K. Siddiqui, M. Imran, M. Hussain, H.M. Bilal, I.Z. Cheema, A. Tabraiz, and Z. Saleem, *Computing the metric dimension of gear graphs*, Symmetry **10** (2018), no. 6, Article ID: 209. https://doi.org/10.3390/sym10060209.

[22] T. Jayabarathi, T. Raghunathan, B.R. Adarsh, and P.N. Suganthan, *Economic dispatch using hybrid grey wolf optimizer*, Energy **111** (2016), 630–641. https://doi.org/10.1016/j.energy.2016.05.105.

[23] J. Kratica, V. Kovačević-Vujčić, and M. Čangalović, *Computing the metric dimension of graphs by genetic algorithms*, Comput. Optim. Appl. **44** (2009), no. 2, 343–361.

https://doi.org/10.1007/s10589-007-9154-5.

[24] J.B. Liu, M.F. Nadeem, H.M.A. Siddiqui, and W. Nazir, *Computing metric dimension of certain families of toeplitz graphs*, IEEE Access **7** (2019), 126734–126741.
https://doi.org/10.1109/ACCESS.2019.2938579.

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

[26] N. Mladenović, J. Kratica, V. Kovačević-Vujčić, and M. Čangalović, *Variable neighborhood search for metric dimension and minimal doubly resolving set problems*, European J. Oper. Res. **220** (2012), no. 2, 328–337.
https://doi.org/10.1016/j.ejor.2012.02.019.

[27] B. Mohamed and M. Amin, *A hybrid optimization algorithms for solving metric dimension problem*, International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC) **15** (2023), no. 1/2, 1–10.
https://doi.org/10.5121/jgraphoc.2023.15201.

[28] B. Mohamed, L. Mohaisen, and M. Amin, *Binary equilibrium optimization algorithm for computing connected domination metric dimension problem*, Sci. Program. **2022** (2022), no. 1, Article ID: 6076369.
https://doi.org/10.1155/2022/6076369.

[29] D.T. Murdiansyah and Adiwijaya, *Computing the metric dimension of hypercube graphs by particle swarm optimization algorithms*, Recent Advances on Soft Computing and Data Mining (Cham) (T. Herawan, R. Ghazali, N.M. Nawi, and M.M. Deris, eds.), Springer International Publishing, 2017, pp. 171–178.

[30] R. Pal, M. Saraswat, S. Kumar, A. Nayyar, and P.K. Rajput, *Energy efficient multi-criterion binary grey wolf optimizer based clustering for heterogeneous wireless sensor networks*, Soft Comput. **28** (2024), no. 4, 3251–3265.
https://doi.org/10.1007/s00500-023-09316-0.

[31] A. Raghavendra, B. Sooryanarayana, and C. Hegde, *Bi-metric dimension of graphs*, British J. Math. & Comput. Sci. **4** (2014), no. 18, Article ID: 2699.

[32] R.R. Rajammal, S. Mirjalili, G. Ekambaram, and N. Palanisamy, *Binary grey wolf optimizer with mutation and adaptive k-nearest neighbour for feature selection in parkinson's disease diagnosis*, Knowledge-Based Systems **246** (2022), Article ID: 108701.
https://doi.org/10.1016/j.knosys.2022.108701.

[33] S. Rehman, M. Imran, and I. Javaid, *On the metric dimension of arithmetic graph of a composite number*, Symmetry **12** (2020), no. 4, Article ID: 607.
https://doi.org/10.3390/sym12040607.

[34] S.W. Saputro, *On the metric dimension of biregular graph*, J. Inf. Process. **25** (2017), 634–638.
https://doi.org/10.2197/ipsjjip.25.634.

[35] Z. Shao, S.M. Sheikholeslami, Pu Wu, and J.B. Liu, *The metric dimension of some generalized Petersen graphs*, Discrete Dyn. Nat. Soc. **2018** (2018), no. 1,

Article ID: 4531958.
https://doi.org/10.1155/2018/4531958.

[36] P. Slater, *Leaves of trees*, Proc. 6th Southeastern Conf. on Combinatorics, Graph Theory, and Computing, Congr. Numer., vol. 14, Elsevier, 1975, pp. 549–559.

[37] B. Sooryanarayana, S Kunikullaya, and N.N. Swamy, *Metric dimension of generalized wheels*, Arab J. Math. Sci. **25** (2019), no. 2, 131–144.

[38] T. Vetrík, *The metric dimension of circulant graphs*, Canad. Math. Bull. **60** (2017), no. 1, 206–216.
https://doi.org/10.4153/CMB-2016-048-1.

[39] ______, *On the metric dimension of directed and undirected circulant graphs*, Discuss. Math. Graph Theory **40** (2019), no. 1, 67–76.
http://dx.doi.org/10.7151/dmgt.2110.

[40] J. Wu, L. Wang, and W. Yang, *Learning to compute the metric dimension of graphs*, Appl. Math. Comput. **432** (2022), Article ID: 127350.
https://doi.org/10.1016/j.amc.2022.127350.