J. Appl. Math. & Informatics Vol. 43(2025), No. 2, pp. 559 - 575
https://doi.org/10.14317/jami.2025.559

BINARY BAO FOR COMPUTING NON-ISOLATED METRIC DIMENSION PROBLEM

IQBAL M. BATIHA*, MOHAMMAD SHEHAB, HAMZAH O. AL-KHAWALDEH, NIDAL ANAKIRA, ALA ALI AMOURAH, BASMA MOHAMED

Abstract. The NP-hard problem of determining the minimum non-isolated resolving set of graphs is examined in this study. If a connected graph G has a vertex set X that resolves G, then every vertex in G may be uniquely identified by its vector of distances to the vertices in X. In addition, the resolving set X of G is referred to as the non-isolated resolving set if the non-isolated vertex causes $v \in X$ to not exist for all v. The smallest cardinality of a non-isolated resolving set in G is known as a non-isolated resolving number, or ni (G). Recently, a proposed heuristic technique called Aquila Optimizer (AO). It is an innovative optimization technique based on population size. The Aquila's natural behavior was imitated in its creation. Its creation was modeled after the way aquilas in the wild pursue and capture their prey. In order to address continuous optimization problems in its original form, the AO algorithm was created. The non-isolated metric dimension is computed by a binary version of the Aquila optimizer (BAO) algorithm. The objects of BAO are binary encoded and used to represent which one of the vertices of the graph belongs to the non-isolated resolving set. The feasibility is enforced by repairing solutions such that an additional vertex generated from vertices of G is added to X and this repairing process is iterated until X becomes the non-isolated resolving set. This is the first attempt to determine the non-isolated metric dimension problem (MDP) heuristically. The proposed BAO is compared to binary smell agent optimization (BSAO), binary dragonfly algorithm (BDA) and binary sand cat swarm optimization (BSCSO) algorithms. Computational results confirm the superiority of the BAO for computing the non-isolated metric dimension.

AMS Mathematics Subject Classification : 35D30, 35K58.
Key words and phrases : Weak solution, singular and degenerate parabolic problem, nonlinear integral condition.

Received November 24, 2024. Revised February 21, 2025. Accepted March 7, 2025.
*Corresponding author.

1. Introduction

Let G be a finite, connected, simple graph. The length of the shortest path from u to v in G is the distance between u and v in G, represented by $d(u, v)$, for u, $v \in V(G)$. Allow X to be an ordered subset of $V(G) = x_1, x_2, \cdots, x_k$. The metric dimension of G, represented as $dim(G)$, is the lowest cardinality of a resolving set of G. If X induces a subgraph of G without an isolated vertex, we consider X a non-isolated resolving set of G. The non-isolated resolving number of G, or $ni(G)$, is the cardinality of the minimum non-isolated resolving set of G. It follows that $dim(G) \leq ni(G)$ since a non-resolving set of G is similarly a resolving of G.

Example 1.1. *The complete graph K_5 is given in Figuer 1. The set $X = v_1, v_2, v_3, v_4$ is a minimal non-isolated resolving set Thus, $ni(K_5) = 4$.*

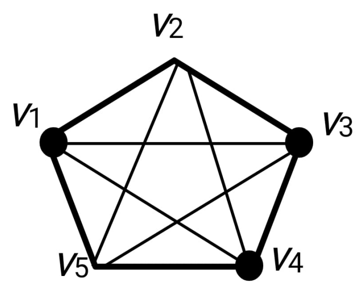

FIGURE 1. Complete graph K_5

It was only recently [1] that the fundamental metric dimension of graphs was introduced. The non-isolated metric dimension problem and the metric dimension problem are both NP-complete [2, 3]. Networks that use the non-isolated theory include wireless communication networks, electrical networks, business networks, and chemical structures [4, 5]. A minimal resolving set of a graph has been designed in [6, 7] to solve the problem of uniquely locating an intruder in a network. The authors independently proposed the idea in [3] of using a graph's cardinality number as the metric dimension and its smallest resolving set as the metric basis. For a number of graphs in the literature, the metric dimension is computed theoretically [8, 9, 10, 11, 12, 13, 14, 15, 16]. Also, for other types of graphs, the reader may refer to the references [17, 18, 19, 20, 21, 22, 23, 24]. Some realization results for the bounds on the metric dimension of a bipartite graph were given by Jothi et al. [8]. Two kinds of bicyclic graphs were the subject of metric dimension studies by Khan et al. [9]. Their metric dimension is constant, as demonstrated by the observed results. The metric dimensions of polygonal networks, specifically the subdivided honeycomb, Aztec diamond, and subdivided Aztec diamond networks, were examined by Zhang et al. [10]. The metric dimension and distance matrix of sunflower graphs and flower snarks were characterized by Girisha et al. [11]. Mohamed et al.'s [12] research looked into the metric dimension of subdivisions of a number of graphs, such as the tadpole, star, and coconut trees, as well as the Lilly and Bistar trees. The first attempt to use a binary version of the equilibrium optimization algorithm (BEOA) to compute the minimal connected dominant resolving set of graphs heuristically was given by Amin et al. [13]. The connected resolving set was found heuristically by Mohamed et al. [14] using BEHHO, which combines chaotic local search, opposition-based learning, and traditional HHO. It also has an S-shaped transfer function that allows it to transform a continuous variable into a binary one. The precise value of the secure resolving set for a number of networks, including the tortoise, open ladder, $Z-(Pn)$, and trapezoid networks, was ascertained by Amin et al. [15]. Heuristically, Mohamed et al. [16] identified the primary metric dimension problem. Nonetheless, the literature has suggested a few methods for heuristically computing the metric dimension. These are genetic algorithm [25], particle swarm optimization [26], and variable neighborhood search [27].

The non-isolated metric dimension is investigated in [28, 29, 30]. The non-isolated resolving number of graphs with homogeneous pendant edges with G and a route P_n, complete graph K_n, and cycle C_n are determined theoretically in [28]. The Cartesian product of a road and a particular graph (a cycle, a complete graph, a complete bipartite graph, or a friendship graph) can theoretically have its non-isolated resolving number found in [29]. In [30], Chitra et al. began their study of non-isolated resolving sets. They have offered some important discoveries and issues that need further investigation. Kamala et al. [31] initiated a study of non-isolated resolving sets and presented several important findings as well as difficulties that required further investigation. Hakanen et al. introduced and examined the idea of strong base forced vertices in unicyclic networks [32]. The MDP was resolved by Wang et al. [33]. Specifically, a data structure of non-resolving tables is provided, and a hybrid approach to learning the MDP is developed. Sedlar et al. [34] described three graph configurations and showed that if and only if the graph contained at least one of these configurations, the value taken was the bigger of the two potential values. Krekovski et al. [35] dealt with several forms of metric dimension in order to highlight that dealt with in particular scenarios with the conventional metric dimension. Kuziak et al. [36] not only survived the state of knowledge on powerful resolving graphs but also came up with some original theories on their characteristics. See [37, 38, 39, 40] for additional results. A number of graphs, such as the line graph, the alternate triangular belt graph, the bistar graph, the triangular snake graph, and others, were the subject of an investigation by Mohamed et al. [37] about their dominance and independent dominating set. In their study, Amin et al. [38] examined ladder graphs with related metric dimensions, including circular, open, triangular, open diagonal, and slanting ladder graphs. The secure metric dimension of specific graphs, including globe, flag, H-graph of path, bistar, and tadpole graphs, was found by Mohamed et al. [39]. The AO approach can be applied to many other disciplines, such as data clustering, global optimization, stochastic programming, deep learning, classification issues, various engineering designs, and more [40]. This report details the initial heuristic attempt to identify the minimal non-isolated resolving set of graphs. We alter the phases of a binary variant of the Aquila optimizer (BAO) method to solve the issue. The suggested BAO is tested using graph findings that have been theoretically calculated. The suggested approach is evaluated with competitive algorithms using hypothetically computed graphs.

The paper is organized as follows: Section 2 gives an overview of Aquila Optimizer (AO). Section 3 contains the BAO for calculating the non-isolated metric dimension. Section 4 reports the computational results. A final conclusion is presented in Section 5.

2. Overview of Aquila Optimizer (AO)

The population-based meta-heuristic Aquila Optimizer (AO) was first developed by Abualigah et al. [41]. The Latin term for eagles, or birds of prey, is Aquila. These birds are considered to be among the most cunning and proficient predators, possessing powerful, sharpened claws and strapping feet that allow them to take their prey with speed and agility. Aquila uses four main hunting techniques: strolling and seizing prey, contour fighting with a short glide, low fly with a slow descent, and high soar with a vertical stoop. These four core Aquila hunt mechanisms served as the basis for the creation of the Aquila Optimizer (AO), a nature-inspired optimization algorithm that fundamentally clarifies the actions of each hunt stage. The five significant steps—initialization, expanded exploration, narrowed exploration, expanded exploitation, and narrowed exploitation—are the main focus of the traditional Aquila Optimizer (AO). One of the most important features of the algorithm—current iteration $\leq (2-3)$ maximum iteration—usually governs how the AO algorithm follows a path from the exploration to the exploitation step. The exploration step will be activated if the condition mentioned is true; otherwise, the exploitation step will be carried out. A novel Binary version of the Aquila Optimizer (BAO) algorithm is put out in [42] in order to address the GS issue. The AO structure was changed once more in [43] to address issues with binary optimization. Real-world issues are not typically characterized by continuous values. In order to increase the searching performance for global optimization issues, Wang et al. [44] presented an improved hybrid Aquila Optimizer (AO) and Harris Hawks Optimization (HHO) algorithm, called IHAOHHO. In order to decrease overall energy usage, Utama et al. [45] presented the Hybrid Aquila Optimizer (HAO) as a solution to the Hybrid Flow Shop Scheduling Problem (HFSSP). In order to decrease power consumption and increase network lifetime, Taha et al. [46] presented a unique enhancement technique that makes use of Aquila Optimizer (AO) to improve energy balance in clusters between sensor nodes during network communications.

Step 1: Initialization
The population of solutions has been produced at random in this first phase, and the other parameters of AO are initialized.

Step 2: More Investigation
Eq. (1) provides a mathematical representation of the main hunting strategy used by Aquila, which is the high soar with vertical stoop, which is essentially shown by expanded exploration. The algorithms' aim is to determine the search space from the high ascend.

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + (X_M(t) - X_{best}(t) * rand) \tag{1}$$

where the following iteration of t's solution, denoted as $X_1(t+1)$, is produced by the first search method, X_1. At this point, the best solution found is $X_{best}(t)$. The exploration is controlled by $(1 - \frac{1}{T})$. The symbols T and t stand for the maximum and current iterations, respectively. The location mean value of the present solution is denoted by $X_M(t)$. The range of the rand value is 0 to 1. The formula for X_M is as follows:

$$X_M(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t), \forall i = 1, 2, \cdots, Dim \tag{2}$$

In this case, N is the population size and Dim is the dimension.

Step 3: Narrowed Exploration
The second hunting strategy used by Aquila, known as the contour battle with brief glide, is projected in the third phase, or narrowed exploration. In this approach, the aquila arranges the terrain such that it can circle around the target animal to attack. Eq. (3) is used to precisely highlight this Aquila behavior.

$$X_2(t+1) = X_{best}(t) \times Levy(D) + (X_R(t) + (y - x) * rand) \tag{3}$$

The next iteration of the problem is represented by $X_2(t+1)$, the best solution is represented by $X_{best}(t)$, the random solution belonging to $[1, N]$ is represented by $X_R(t)$, the random number belonging to [0, 1] is represented by $rand$, and the spiral form in the search is represented by x and y, which are denoted using Eq. (4).

$$x = r \times \sin(\theta), \; y = r \times \cos(\theta) \tag{4}$$

Additionally, within the same structure, the Lévy fight distribution function, Levy(D), utilized in Eq. (3) is shown in Eq. (5), and σ is determined by Eq. (6), in that order.

$$levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{5}$$

$$\sigma = \frac{\tau(1+\beta) \times \sin\left(\frac{\beta\pi}{2}\right)}{\tau\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \tag{6}$$

The Lévy fight distribution function at dimension space D is shown by Levy (D), s is a constant value assigned as 0.1, u and v are random numbers within [0,1], β is the constant value [0.5], and σ is one of the parameters in the Lévy fight distribution function.

Step 4: Expanded Exploitation

The concept found in the third hunting mechanism is essentially depicted in the fourth phase, called expanded exploitation, as a low flight attack with a slow descent, wherein Aquila approaches the target to attack by progressively diving into the intended area. Thus, with Eq. (7), this action of Aquila is accurately revealed.

$$X_3(t+1) = (X_{best}(t) \times (X_M(t)) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta \quad (7)$$

where T is the highest possible number of iterations. The variables α and δ represent the fixed exploitation adjustment parameters, and UB and LB stand for the upper and lower bounds, respectively.

Step 5: Narrowed Exploitation

The concept for the final step, known as narrowed exploitation, is derived from the walk-and-grab attack, which is the last offensive strategy used by Aquila. This can be expressed mathematically using Eq. (8).

$$X_4(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times rand) - (G_2 \times Levy(D) + rand \times G_1) \quad (8)$$

Additionally, different motions G_1 and combat slope G_2 are represented using Eqs. (9, 10, and 11), respectively, and the function known as $QF(t)$ (Quality Function), which is used to balance the search technique.

$$QF(t) = t^{\frac{2 \times rand - 1}{(1-T)^2}} \quad (9)$$

$$G_1 = 2 \times rand - 1 \quad (10)$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \quad (11)$$

3. Binary Aquila Optimizer for Non-Isolated Resolving Number

Because it keeps track of a population of solutions and searches a large area for the best global solution, the Aquila Optimizer can handle hard optimization problems with numerous locally optimal solutions. Because of this benefit, the binary version of the method can be used to solve the non-isolated resolving number problem. Solutions can traverse the search space in the continuous version of AO through the use of position vectors in the continuous real domain. Through the use of the S-shaped transfer function, we may change the continuous variable AO into binary numbers. Position changes need a flip between 0 and 1 in the discrete binary search space.

The initialization phase makes use of the following equation:

$$SObinary_{ij} = \begin{cases} 1, & rand() > 0.5 \\ 0, & else \end{cases} \tag{12}$$

where a random number between 0 and 1 is called a rand.

To convert continuous numbers to binary ones, a transfer function is used. The sigmoid function (S) is applied in this study in the following ways:

$$S = \frac{1}{1 + e^{-10x^d}} \tag{13}$$

where S is the function's output and x^d denotes the continuous-valued location at dimension d. To produce a binary value, apply the equation below.

$$Obinary_{ij} = \begin{cases} 1, & rand() < S \\ 0, & otherwise \end{cases} \tag{14}$$

The suggested algorithm tackles the non-isolated resolving set problem as an optimization problem, finding the optimal solution to allow each object to be represented as a vector in one dimension. Obinaryij, denoted as $(O_{i1}, O_{i2}, \cdots, O_{ij})$, is a binary-valued position vector. If the j-th element of the vector has a value of 1, it means that vertex j belongs to B. B is a non-isolated resolving set if each $v \in V(G)$ has a unique representation, $r(v|B)$.

Finding the value of the S-shaped transfer function gives the value of a binary-valued position vector. A solution in the BAO algorithm is repaired by adding a vertex from $\frac{V}{B}$ when it is not feasible as a non-isolated resolving set. Until that item becomes a non-isolated resolving set, this repair is applied.

Every solution in the population is represented by the algorithm as a string of binary values, where 1 denotes the selection of the non-isolated resolving set and corresponds to a value of "1". If the non-isolated resolving set is not chosen, the corresponding value is "0".

Thus, the pseudocode in Algorithm 1.

Algorithm 1 Pseudocode of BAO Algorithm

Initialization:
Set the population solution X
Set the initial parameters
while $t < T$ **do**
  Evaluate the cost function
  Decide the best solution
  for $i = 1, 2, \ldots, N$ **do**
    Update the mean value of the current solution $X_M(t)$
    Update the algorithm parameters $(x, y, G_1, G_2, Levy(D), etc.)$
    if $t \leq \frac{2}{3} * T$ **then**
      if $rand \leq 0.5$ **then**
        Update the existing solution using Eq.(1)
        Convert each $\overrightarrow{SO_i}$ into binary using the S-shaped transfer function to obtain $SObinary_{ij}$
        Calculate the fitness of each $SObinary_{ij}$
        Update the new position of the solution using Eq.(14)
      else
        Update the existing solution using Eq.(3)
        Convert each $\overrightarrow{SO_i}$ into binary using the S-shaped transfer function to obtain $SObinary_{ij}$
        Calculate the fitness of each $SObinary_{ij}$
        Update the new position of the solution using Eq.(14)
      end if
    else
      if $rand \leq 0.5$ **then**
        Update the existing solution using Eq.(7)
        Convert each $\overrightarrow{SO_i}$ into binary using the S-shaped transfer function to obtain $SObinary_{ij}$
        Calculate the fitness of each $SObinary_{ij}$
        Update the new position of the solution using Eq.(14)
      else
        Update the existing solution using Eq.(8)
        Convert each $\overrightarrow{SO_i}$ into binary using the S-shaped transfer function to obtain $SObinary_{ij}$
        Calculate the fitness of each $SObinary_{ij}$
        Update the new position of the solution using Eq.(14)
      end if
    end if
  end for
end while
Return the best solution X_{best}

4. Results and Discussion

This section summarizes the results of BAO applied to graph instances that are computed theoretically.

4.1. Experimental Results. The AO imitates how Aquila's hunt in nature. While slow-moving prey hunting tactics demonstrate the algorithm's capacity for local exploitation, fast-moving prey hunting techniques showcase the algorithm's capacity for global exploration. It is not possible to apply the canonical AO to problems involving discrete, binary, or mixed-integer variables since it only works on the continuous solution search space. While the AO's search space is continuous, the BAO algorithm uses an S-shaped transfer function to determine the position of solutions in the binary space.

This section compares the proposed BAO with the following algorithms: BSAO, BDA and BSCSO. Path Pn, Complete graph K_n, Complete bipartite graph $K_{m,n}$, and friendship graph G with k-triangle graphs are among the graph types to which the techniques are used. The computer system used for the algorithm tests and comparisons was Windows 10 Ultimate 64-bit; the CPU was an Intel Core i7 with 16 GB of RAM, the hard drive was a 1TB HDD+1TBSSD, and the code was programmed in MATLAB 2021b. Table 1 displays the values for the parameter settings. The maximum number of iterations is set at 1000, with population size equal to 50 and termination condition.

Table 1. Parameter setting

Algorithms	Parameter	Value
AO	U	0.00565
	r_1	10
	ω	0.005
	α	0.1
	δ	0.1
	G_1	$\in [-1, 1]$
	G_2	$\in [2, 0]$
SCSO	r_G	2 up to 0
	R	-2.r_G to 2.r_G
BOA	a	0.1
SAO	T	3
	SL	2.5
	M	2.4
BPSO	C_1	Increases linearly 0.5-2.5
	C_2	Decreases linearly 2.5-0.5
	Inertia weight (w)	0.8

For every graph, all algorithms have been run 20 times; the outcomes are presented in tables 2–5. The structure of the tables is as follows:
-The number of nodes (n), edges (M), non-isolated resolving number (ni), CPU time (t) used to indicate the precisely non-isolated resolving number are contained in the columns, respectively.

TABLE 2. Results on complete graph K_k

Instance	n	m	$\mathbf{BAO}_{best}$	t	BSAO	t	BDA	t	BSCSO	t
K_1	3	3	2	2.65	2	5.91	2	8.23	2	4.07
K_2	4	6	3	7.51	3	24.09	3	39.17	3	5.69
K_3	5	10	4	20.7	4	79.15	4	84.52	4	32.16
K_4	6	15	5	51.4	5	109.43	5	132.09	5	85.13
K_5	7	21	6	84.6	6	221.07	6	241.81	6	145.59
K_6	8	28	7	126.7	7	334.02	7	350.15	7	232.04
K_7	9	36	8	168.3	8	492.3	8	445.27	8	317.38
K_8	10	45	9	289.8	9	627.6	9	537.11	9	262.29
K_9	11	55	10	345.9	10	838.91	10	643.04	10	534.15
K_{10}	12	66	11	408.2	11	910.72	11	781.23	11	641.73
K_{11}	13	78	12	492.6	12	1034.6	12	935.18	12	728.26
K_{12}	14	91	13	548.4	13	1193.1	13	1012.4	13	850.14
K_{13}	15	105	14	627.03	14	1254.5	14	1306.1	14	986.58
K_{14}	16	120	15	721.8	15	1316.2	15	1525.6	15	1147.1
K_{15}	17	136	16	769.07	16	1405.8	16	1694.3	16	1298.5
K_{16}	18	153	17	846.9	17	1509.3	17	1289.7	17	1379.1
K_{17}	19	171	18	929.6	18	1652.1	18	1591.4	18	1490.8
K_{18}	20	190	19	1042	19	1803.6	19	1877.9	19	1673.3
K_{19}	21	210	20	863.2	20	2019.4	20	2084.1	20	1936
K_{20}	22	231	21	935.1	21	2198.1	21	2473	21	2079.5

Our stopping criterion is the cardinality of the non-isolated resolving set that reaches the known non-isolated metric dimension of the complete graph. For K5, the time needed for BAO is 84.6 sec.

TABLE 3. Results on path graph

Instance	n	m	$\mathbf{BAO}_{best}$	t	BSAO	t	BDA	t	BSCSO	t
P_3	3	2	2	1.03	2	5.08	2	7.19	2	4.12
P_4	4	3	2	2.14	2	9.39	2	15.27	2	8.53
P_5	5	4	2	9.45	2	20.04	2	23.39	2	25.06
P_6	6	5	2	34.18	2	56.11	2	98.15	2	42.19
P_7	7	6	2	73.01	2	132.03	2	162.54	2	78.62
P_8	8	7	2	103.14	2	171.58	2	209.31	2	85.09
P_9	9	8	2	141.25	2	218.26	2	385.49	2	129.17
P_{10}	10	9	2	212.53	2	287.31	2	506.2	2	314.03
P_{11}	11	10	2	301.28	2	374.09	2	613.7	2	405.25
P_{12}	12	11	2	389.05	2	496.7	2	721.04	2	490.01
P_{13}	13	12	2	461.57	2	721.04	2	807.3	2	652.83
P_{14}	14	13	2	548.22	2	892.2	2	1103	2	709.37
P_{15}	15	14	2	639.16	2	1107.6	2	1358.9	2	815.09
P_{16}	16	15	2	704.89	2	1288.1	2	1481.2	2	977.54
P_{17}	17	16	2	832.09	2	1416	2	1672.7	2	1218.5
P_{18}	18	17	2	911.3	2	1542.1	2	1899.1	2	1435
P_{19}	19	18	2	774.2	2	1747.8	2	1965	2	1596.9
P_{20}	20	19	2	1078.2	2	1963.1	2	2129.1	2	1803.2

Regarding BAO results, Table 3 shows that for the path graph P_n, $3 \leq n \leq 20$, BAO has reached an optimal solution. For example, in P_5, the time needed for BAO is 9.45 sec. The non-isolated resolving number of graphs with a path P_n, a complete graph K_n, and a cycle C_n with homogeneous pendant edges with G is determined theoretically in [35].

TABLE 4. Results on Km,n

Instance	n	m	BAO$_{best}$	t	BSAO	t	BDA	t	BSCSO	t
$K_{2,2}$	4	4	2	7.13	2	16.02	2	12.9	2	11.4
$K_{3,3}$	6	9	4	19.51	4	28.25	4	25.06	4	17.02
$K_{4,4}$	8	16	6	37.08	6	54.31	6	43.2	6	34.7
$K_{5,5}$	10	25	8	82.27	8	143.1	8	99.4	8	127.3
$K_{6,6}$	12	36	10	116	10	250.4	10	176.1	10	201.9
$K_{7,7}$	14	49	12	204	12	384.2	12	287.2	12	296.7
$K_{8,8}$	16	64	14	333	14	513.9	14	391.7	14	373.1
$K_{9,9}$	18	81	16	471	16	651	16	514.	16	490.2
$K_{10,10}$	20	100	18	593	18	792.3	18	716.8	18	658.3
$K_{11,11}$	22	121	20	682	20	871.1	20	895.2	20	824.5
$K_{12,12}$	24	144	22	799	22	1025	22	983.1	22	959
$K_{13,13}$	26	169	24	925	24	1194	24	1047	24	1033
$K_{14,14}$	28	196	26	1031	26	1349	26	1252	26	1228
$K_{15,15}$	30	225	28	885.9	28	1484	28	1475	28	1464
$K_{16,16}$	32	256	30	1102	30	1659	30	1742	30	1580
$K_{17,17}$	34	289	32	1393	32	1807	32	1825	32	1729
$K_{18,18}$	36	324	34	1173	34	2013	34	1953	34	1989
$K_{19,19}$	38	361	36	1247	36	2130	36	2209	36	2056
$K_{20,20}$	40	400	38	1358	38	2306	38	2493	38	2192

Regarding BAO results, Table 4 shows that for the complete bipartite graph $K_{n,m},\ 2 \leq n,\ m \leq 20$, BAO has reached an optimal solution. For instance, in $K_{5,5}$, the time needed for BAO is 82.27 sec.

TABLE 5. Results on friendship with k-triangles

Instance	n	m	BAO$_{best}$	t	BSAO	t	BDA	t	BSCSO	t
F_3^2	5	6	3	6.08	3	15.2	3	21.5	3	9.3
F_3^3	7	9	4	34.5	4	72.08	4	54.3	4	27.8
F_3^4	9	12	5	61.9	5	95.4	5	101.9	5	531.2
F_3^5	11	15	6	97.04	6	178.1	6	183.4	6	145.6
F_3^6	13	18	7	126.8	7	236.2	7	257.8	7	242.8
F_3^7	15	21	8	183.09	8	351.03	8	330.5	8	396.05
F_3^8	17	24	9	248.2	9	482.9	9	461.3	9	408.4
F_3^9	19	27	10	331.5	10	580.4	10	536.9	10	653.1
F_3^{10}	21	30	11	401.3	11	694.7	11	688.4	11	742.5
F_3^{11}	23	33	12	539.2	12	789.03	12	806.4	12	503.9
F_3^{12}	25	36	13	665.9	13	906.2	13	1057.3	13	794.8
F_3^{13}	27	39	14	598.5	14	1091	14	1173.2	14	957.1
F_3^{14}	29	42	15	679.4	15	1257.9	15	1268.3	15	1039.5
F_3^{15}	31	45	16	813.6	16	1348.1	16	1399.2	16	1228.2
F_3^{16}	33	48	17	897.3	17	1579.2	17	1527.5	17	1490.1
F_3^{17}	35	51	18	962.08	18	1651.5	18	1642.3	18	1618.9
F_3^{18}	37	54	19	718.3	19	1740.1	19	1856.2	19	1796.6
F_3^{19}	39	57	20	1046.5	20	1913.6	20	1993	20	1852
F_3^{20}	41	60	21	1183.2	21	2092.4	21	2178.9	21	1907.2

Regarding BAO results, Table 5 shows that for the friendship with k-triangles graph $F_3^k,\ 2 \leq k \leq 20$, BAO has reached an optimal solution. For instance, in F_3^6, the time needed for BAO is 126.8 sec.

5. Comparison

The results for a variety of graphs are shown in Tables 2, 3, 4, and 5. They demonstrate that, particularly for the path graph and the complete graph, the proposed BAO may reach the most optimal solution (known non-isolated metric dimension) in an appropriate duration of time. It demonstrates the accuracy and superiority of the proposed BAO.

Experiments in this paper are performed on a subset of complete graph instances with $n \leq 22$ and $m \leq 231$ in Table 2, path graph instances with $n \leq 20$ and $m \leq 21$ in Table 3, complete bipartite graph instances with $n \leq 40$ and $m \leq 400$ in Table 4, and friendship with k-triangles graph instances with $n \leq 41$ and $m \leq 60$ in Table 5.

The superiority of the proposed BAO in relation to the non-isolated metric dimension is demonstrated in Figures 2, 3, 4, and 5. For example, the non-isolated metric dimension by BAO for $K_{4,4}$ is 6 and reaches 37.08 sec. P_5 is 2 and reaches 9.45 sec. All figures show the superiority of the proposed BAO.

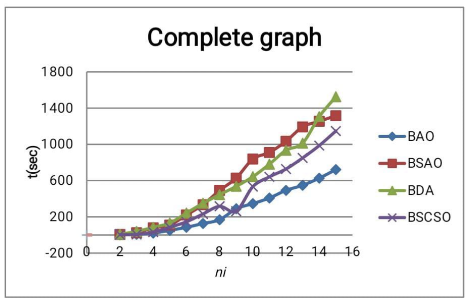

FIGURE 2. Comparison between BAO_{best} and t (sec) for computing the non-isolated metric dimension of a complete graph.

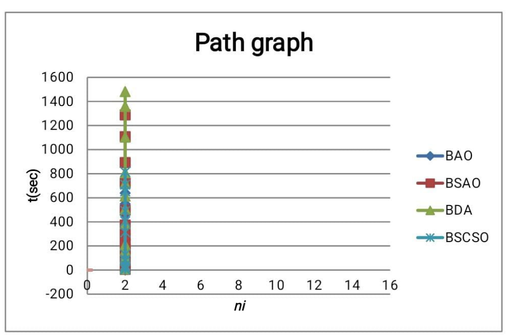

FIGURE 3. Comparison between BAO_{best} and t (sec) for computing the non-isolated metric dimension of a path graph.

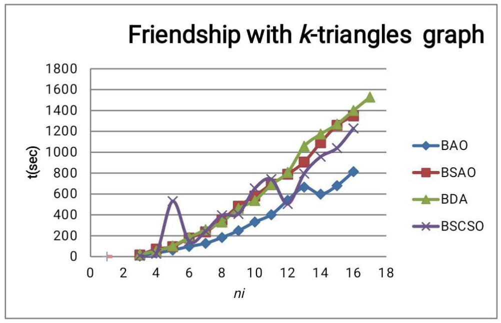

FIGURE 5. Comparison between BAO_{best} and t (sec) for computing the non-isolated metric dimension of friendship with k-triangles graph.

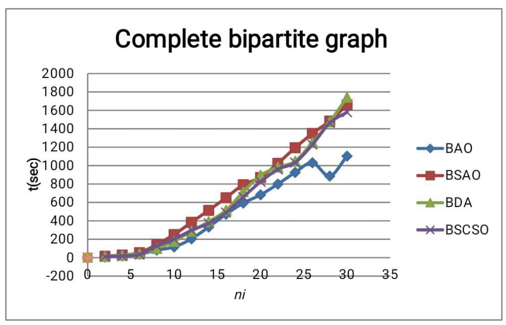

FIGURE 4. Comparison between BAO_{best} and t (sec) for computing the non-isolated metric dimension of a complete bipartite graph.

6. Conclusion

In this study, the non-isolated metric dimension problem is solved by adapting the operations of a binary version of the Aquila optimizer algorithm BAO. The proposed BAO is tested using graph results that are computed theoretically. The proposed algorithm is compared to competitive algorithms on graphs that are computed theoretically. The performance of the proposed BAO outperforms that of the BSAO, BDA, and BSCSO. Our goal is to determine the non-isolated metric dimension of many graphs in the future, such as subdivisions of crown graphs, square pyramid graph P_4, cocktail party graphs, and triangle pyramid graph P_3. We also plan to compare our method with other metaheuristic algorithms that compute variants of the metric dimension. Additionally, we use a binary version of the Aquila optimizer (BAO) algorithm to compute the edge dominating metric dimension problem and compare it to other metaheuristic techniques.

Conflicts of interest : The authors declare that they have no conflicts of interest.

Data availability : No underlying data were collected or produced in this study.

Acknowledgments : We acknowledge and thank the reviewers for their thorough discussion and comments.

References

1. F.M. Feng, M. Xu, K. Wang, *On the metric dimension of line graphs*, Discrete Applied Mathematics **161** (2013), 802-805.
2. M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1979.
3. F.H. Melter, F. Harary, *On the metric dimension of a graph*, Ars Combinatoria **2** (1976), 191-195.
4. J.L. Hurink, T. Nieberg, *Approximating minimum independent dominating sets in wireless networks*, Information Processing Letters **109** (2008), 155-160.
5. A.H. Karbasi, R.E. Atani, *Application of dominating sets in wireless sensor networks*, International Journal of Security and Its Applications **7** (2013), 185-202.
6. D. Vukičević, A. Klobučar, *K-Dominating sets on linear benzenoids and on the infinite hexagonal grid*, Croatica Chemica Acta **80** (2007), 187-191.
7. P.J. Slater, *Leaves of trees*, Congressus Numerantium **14** (1975), 549-559.
8. L. Abualigah, N.K. Al-Okbi, S. Mirjalili, M. Alshinwan, H. Al Hamad, A.M. Khasawneh, A.H. Gandomi, *Moth-Flame Optimization Algorithm, Arithmetic Optimization Algorithm, Aquila Optimizer, Gray Wolf Optimizer, and Sine Cosine Algorithm: A Comparative Analysis Using Multilevel Thresholding Image Segmentation Problems*, In Handbook of Moth-Flame Optimization Algorithm, CRC Press, 2022, 241-263.
9. H.A. Shehadeh, I.H. Jebril, G.M. Jaradat, D. Ibrahim, R. Sihwail, H. Al Hamad, M.A. Alia, *Intelligent Diagnostic Prediction and Classification System for Parkinson's Disease by Incorporating Sperm Swarm Optimization (SSO) and Density-Based Feature Selection Methods*, International Journal of Advances in Soft Computing & Its Applications **15** (2023), 2074-8523.
10. M. Alshinwan, O.A. Khashan, M. Khader, O. Tarawneh, A. Shdefat, N. Mostafa, D.S. AbdElminaam, *Enhanced Prairie Dog Optimization with Differential Evolution for solving engineering design problems and network intrusion detection system*, Heliyon **10** (2024), e36663.
11. S. Alzoubi, L. Abualigah, M. Sharaf, M.S. Daoud, N. Khodadadi, H. Jia, *Synergistic Swarm Optimization Algorithm*, CMES-Computer Modeling in Engineering & Sciences **139** (2024), 2557-2604,
12. D.R. Ibrahim, H.A. Shehadeh, M.A. Aladaileh, K. Alieyan, G.M. Jaradat, W.A. Telfah, X. Wang, *A hybrid facial recognition approach based on grasshopper optimization method*, AIP Conference Proceedings **2979** (2023), 070005.
13. B. Mohamed, L. Mohaisen, M. Amin, *Binary equilibrium optimization algorithm for computing connected domination metric dimension problem*, Scientific Programming **2022** (2022), Article ID 6076369.

14. B. Mohamed, L. Mohaisen, M. Amin, *Computing connected resolvability of graphs using binary enhanced Harris Hawks optimization*, Intelligent Automation & Soft Computing **36** (2023), 2349-2361.
15. B. Mohamed, M. Amin, *Domination number and secure resolving sets in cyclic networks*, Applied and Computational Mathematics **12** (2023), 42-45.
16. B. Mohamed, L. Mohaisen, M. Amin, *Binary Archimedes optimization algorithm for computing dominant metric dimension problem*, Intelligent Automation & Soft Computing **38** (2023), 19-34.
17. B. Mohamed, I.M. Batiha, M. Odeh, M. El-Meligy, *Computing the independent domination metric dimension problem of specific graphs*, Journal of Mechanics of Continua and Mathematical Sciences **19** (2024), 256-264.
18. I.M. Batiha, I.H. Jebril, B. Mohamed, *Determining the dominant metric dimension for various graphs*, Journal of Mechanics of Continua and Mathematical Sciences **19** (2024), 105-116.
19. I.M. Batiha, N. Anakira, A. Hashim, B. Mohamed, *A special graph for the connected metric dimension of graphs*, Mathematical Models in Engineering **10** (2024), 193-201.
20. I.M. Batiha, B. Mohamed, I.H. Jebril, *Secure metric dimension of new classes of graphs*, Mathematical Models in Engineering **10** (2024), 161-167.
21. I.M. Batiha, B. Mohamed, *Binary rat swarm optimizer algorithm for computing independent domination metric dimension problem*, Mathematical Models in Engineering **10** (2024), 119-132.
22. I.M. Batiha, N. Anakira, B. Mohamed, *Algorithm for finding domination resolving number of a graph*, Journal of Mechanics of Continua and Mathematical Sciences **19** (2024), 18-23.
23. N. Vijaya, B. Rajan, I. Venkat, *Crossing numbers of join of a graph on six vertices with a path and a cycle*, International Journal of Advances in Soft Computing and its Applications **12** (2020), 45-58.
24. A. Kumar, S. Singh, *Fuzzy graph theory approach to network vulnerability assessment*, International Journal of Advances in Soft Computing and its Applications **11** (2019), 123-134.
25. J. Kratica, V. Kovačević-Vujčić, M. Čangalović, *Computing the metric dimension of graphs by genetic algorithms*, Computational Optimization and Applications **44** (2009), 343-361.
26. D.T. Murdiansyah, Adiwijaya, *Computing the metric dimension of hypercube graphs by particle swarm optimization algorithms*, Recent Advances on Soft Computing and Data Mining, Springer 2017, 171-178.
27. N. Mladenović, J. Kratica, V. Kovačević-Vujčić, M. Čangalović, *Variable neighborhood search for metric dimension and minimal doubly resolving set problems*, European Journal of Operational Research **220** (2012), 328-337.
28. R. Alfarisi, D. Dafik, A.I. Kristiana, E.R. Albirri, I.H. Agustin, *Non-isolated resolving number of graphs with homogeneous pendant edges*, AIP Conference Proceedings **2014** (2018), 020001.
29. I.M. Hasibuan, A.N.M. Salman, S.W. Saputro, *Non-isolated resolving sets of certain graphs Cartesian product with a path*, Journal of Physics: Conference Series **1008** (2018), 012045.
30. P.J.B. Chitra, S. Arumugam, *Resolving sets without isolated vertices*, Procedia Computer Science **74** (2015), 38-42.
31. G. Kamala, R. Kali, *Resolving sets without isolated vertices*, Malaya Journal of Matematik **5** (2020), 4576-4579.
32. A. Hakanen, V. Junnila, T. Laihonen, I.G. Yero, *On the unicyclic graphs having vertices that belong to all their (strong) metric bases*, Discrete Applied Mathematics **353** (2024), 191-207.
33. J. Wu, L. Wang, W. Yang, *Learning to compute the metric dimension of graphs*, Applied Mathematics and Computation **432** (2022), 127350.

34. J. Sedlar, R. Škrekovski, *Vertex and edge metric dimensions of unicyclic graphs*, Discrete Applied Mathematics **314** (2022), 81-92.
35. J. Sedlar, R. Škrekovski, *Bounds on metric dimensions of graphs with edge disjoint cycles*, Applied Mathematics and Computation **396** (2021), 125908.
36. D. Kuziak, M.L. Puertas, J.A. Rodríguez-Velázquez, I.G. Yero, *Strong resolving graphs: the realization and the characterization problems*, Discrete Applied Mathematics **236** (2018), 270-287.
37. B. Mohamed, M. Badawy, *Some new results on domination and independent dominating set of some graphs*, Applied and Computational Mathematics **12** (2024), 53-57.
38. M.I. Batiha, M. Amin, B. Mohamed, H.I. Jebril, *Connected metric dimension of the class of ladder graphs*, Mathematical Models in Engineering **10** (2024), 65-74.
39. S. Almotairi, O. Alharbi, Z. Alzaid, B. Almutairi, B. Mohamed, *The secure metric dimension of the Globe graph and the Flag graph*, Advances in Operations Research **2024** (2024), Article ID 3084976.
40. B. Sasmal, A.G. Hussien, A. Das, K.G. Dhal, *A comprehensive survey on Aquila optimizer*, Archives of Computational Methods in Engineering **30** (2023), 4449-4476.
41. S. Mahajan, L. Abualigah, A.K. Pandit, M. Altalhi, *Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks*, Soft Computing **26** (2022), 4863-4881.
42. E. Pashaei, *Mutation-based Binary Aquila optimizer for gene selection in cancer classification*, Computational Biology and Chemistry **101** (2022), 107767.
43. E. Baş, *Binary Aquila optimizer for 0-1 knapsack problems*, Engineering Applications of Artificial Intelligence **118** (2023), 105592.
44. S. Wang, H. Jia, Q. Liu, R. Zheng, *An improved hybrid Aquila Optimizer and Harris Hawks Optimization for global optimization*, Mathematical Biosciences and Engineering **18** (2021), 7076-7109.
45. D.M. Utama, M.D. Primayesti, *A novel hybrid Aquila optimizer for energy-efficient hybrid flow shop scheduling*, Results in Control and Optimization **9** (2022), 100177.
46. A.A. Taha, H.O. Abouroumia, S.A. Mohamed, L.A. Amar, *Enhancing the lifetime and energy efficiency of wireless sensor networks using Aquila Optimizer algorithm*, Future Internet **14** (2022), 365.

Iqbal M. Batiha received M.Sc. from Al-Albayt University and Ph.D. at University of Jordan. He is currently a professor at Al Zaytoonah University of Jordan since 2022. His research interests include applied analysis and numerical computations.

Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, Jordan,
Nonlinear Dynamics Research Center (NDRC), Ajman University, Ajman 346, UAE.
e-mail: i.batiha@zuj.edu.jo

Mohammad Shehab received M.Sc. and Ph.D. in Computer Science from Universiti Sains Malaysia. He is currently an Assistant Professor at Amman Arab University since 2021. His research interests are computer science and graph theory.

College of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan.
e-mail: m.shehab@aau.edu.jo

Hamzah O. Al-Khawaldeh received M.Sc. from Al-Albayt University. He is currently a teacher at Ministry of Jordan since 2020. His research interests are applied analysis and numerical computations.

Department of Mathematics, Al Al-Bayt University, Mafraq, Jordan.
e-mail: hamzahabusanad@gmail.com

Nidal Anakira received M.Sc. Ph.D. from in Applied Mathematics from the School of Mathematical Science, University Kebangsaan, Malaysia in 2015. His research interests are applied analysis and numerical computations.

Faculty of Education and Arts, Sohar University, Sohar 3111, Oman.
Applied Science Research Center, Applied Science Private University, Amman 11937, Jordan.
e-mail: Nanakira@su.edu.om

Ala Ali Amourah received M.Sc. Ph.D. from in Applied Mathematics from the School of Mathematical Science, University Kebangsaan, Malaysia in 2015. His research interests are complex analysis and applied analysis.

Faculty of Education and Arts, Sohar University, Sohar 3111, Oman.
Jadara Research Center, Jadara University, Irbid 21110, Jordan.
e-mail: AAmourah@su.edu.om

Basma Mohamed received the B.S. degree, M.S. degree, and Ph.D. degrees in computer science from the Faculty of Science, Menoufia University, in 2011, 2017, and 2023, respectively. He is currently an assistant professor at Giza Higher Institute for Managerial Sciences since 2024. Her research interests are graph theory, discrete mathematics, algorithm analysis, and meta-heuristic algorithms.

College of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan.
Giza Higher Institute for Managerial Sciences, Tomah, Eygpt.
e-mail: bosbos25jan@yahoo.com