Machine Learning Research
2023; 8(1): 9-16
http://www.sciencepublishinggroup.com/j/mlr
doi: 10.11648/j.mlr.20230801.12
ISSN: 2637-5672 (Print); ISSN: 2637-5680 (Online)

SciencePG
Science Publishing Group

Hybridizing Slime Mould Algorithm with Simulated Annealing for Solving Metric Dimension Problem

Basma Mohamed[1, *], Mohamed Amin[2]

[1]Department of Computer Science, Faculty of Computers and Artificial Intelligence, AlRyada University for Science and Technology, Sadat City, Egypt

[2]Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin Elkom, Egypt

Email address:

bosbos25jan@yahoo.com (Basma Mohamed)

[*]Corresponding author

To cite this article:

Basma Mohamed, Mohamed Amin. Hybridizing Slime Mould Algorithm with Simulated Annealing for Solving Metric Dimension Problem. *Machine Learning Research*. Vol. 8, No. 1, 2023, pp. 9-16. doi: 10.11648/j.mlr.20230801.12

Received: September 13, 2023; **Accepted**: October 8, 2023; **Published**: October 28, 2023

Abstract: In this paper, we consider the NP-hard problem of finding the metric dimension of graphs. A set of vertices B of a connected graph $G = (V, E)$ resolves G if every vertex of G is uniquely identified by its vector of distances to the vertices in B. The cardinality of the smallest resolving set of G is the metric dimension of G. The metric dimension problem arises in several different fields, such as robot navigation, telecommunication, and geographical routing protocol. The slime mould algorithm (SMA) is an efficient population-based optimizer based on the oscillation mode of slime mould in nature. The SMA has a specific mathematical model and very competitive results, along with fast convergence for many problems, particularly in real-world cases. SMA has good exploration and exploitation abilities for solving optimization problems. However, complex and high-dimensional SMA may fall into local optimal regions. SA is a very preferable technique among the other heuristic approaches as it provides practical randomness in the search to avoid the local extreme points. However, SA involves a trade-off between computing time and solution sensitivity. The SA is used to enhance the fitness of the best agent if it falls in a suboptimal region, which will lead to the enhancement of all individuals. We solve the problem as integer linear programming and introduce the hybrid algorithm SMA-SA, which combines simulated annealing SA and SMA for determining the metric dimension of graphs. Comparisons were performed on the graphs: k-home chain graph, tadpole graph, alternate triangular snake graph, and mirror graph. Finally, computational results and comparisons with pure SA, SMA, and PSO algorithms confirm the effectiveness of the proposed SMA-SA for solving metric dimension problem.

Keywords: Mirror Graph, Metric Dimension, Simulated Annealing Algorithm, Slime Mould Algorithm

1. Introduction

Let $G = (V, E)$ be a connected graph with a vertex set V and an edge set E. The distance between two vertices $x,y \in V$ is the length of the shortest path between them and is indicated by $d(x, y)$. Let $B = \{b_1, b_2, \ldots, b_k\}$ be an ordered set of vertices of G, and let v be a vertex of G. The $k-$vector

$$r(v|B) = (d\,(v,b_1), d(v,b_2),\ldots,d(v, b_k))$$

is the metric representation of v with respect to B. If $v_i \neq v_j$, then $r(v_i|B) \neq r(v_j|B)$, B is called a locating set or resolving set of G [1]. Let $Card\ (X)$ denote the cardinality of a set X. The metric dimension Md of G is defined as [2],

$$Md\,(G) = \min\,\{\text{Card}\,(B)\text{: } B \text{ is a resolving set of } G\}.$$

Let $D = [d_{ij}]$ be the distance matrix of G, $d_{ij} = d(v_i,v_j)$ for $1 \leq i, j \leq n$. For $x_i \in \{0, 1\}$, $1 \leq i \leq n$, the function F is defined by

$$F(x_1, x_2, \ldots\ldots, x_n) = x_1 + x_2 + \cdots + x_n.$$

Minimizing F subject to the $\binom{n}{2}$ constraints

$$|d_{i1} - d_{j1}|x_1 + |d_{i2} - d_{j2}|x_2 + \ldots\ldots\ldots\ldots + |d_{in} - d_{jn}|x_n > 0 \text{ for } 1 \leq i < j \leq n$$

is equivalent to finding a basis in the sense that if

x_1', x_2',......., x_n' is a set of values for which F attains its minimum, then B={v_i, x_i'=1}is a basis for G and conversely, if B={v_{i1}, v_{i2},......., v_{in}}is a basis for G and if we define

$$x_s' = \begin{cases} 1 \text{ if s } = i_j \text{ for some j } (1 \leq \text{j} \leq \text{k}) \\ \\ \qquad 0\ otherwise \end{cases}$$

then F(x_1', x_2',......., x_n') is a minimum subject to the given constraints.

Example 1: Consider the graph G of Figure 1. The set B={v_1,v_2} is not a resolving set for G since r(v_4|B) = r(v_5|B)= (2,2). On the other hand, B_1={v_1,v_4}is a resolving set for G since the representations for the vertices of G with respect to B_1 are r(v_1,B_1) = (0,2), r(v_2,B_1) = (1,2), r(v_3,B_1) = (1,1), r(v_4,B_1) = (2,0), r(v_5,B_1) = (2,1).

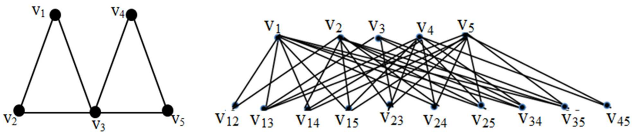

Figure 1. *The graph G and its resolving graph R(G).*

The problem is represented by integer linear programming in example 1 as follows:

$$\text{Min z} = x_1+x_2+x_3+x_4+x_5$$

s. t

$x_1+x_2 \geq 1$

$x_1+ x_3+x_4+x_5 \geq 1$

$x_1+x_2+ x_4+x_5 \geq 1$

$x_1+x_2+ x_4+x_5 \geq 1$

$x_2+ x_3+x_4+x_5 \geq 1$

$x_1+x_2+ x_4+x_5 \geq 1$

$x_1+x_2+ x_4+x_5 \geq 1$

$x_1+x_2+x_3+x_4 \geq 1$

$x_1+x_2+x_3+ x_5 \geq 1$

$x_4+x_5 \geq 1$

$x_1, x_2, x_3, x_4, x_5 \geq 0$

Assume that V is the vertex set of G and V_p is the collection of all $\binom{n}{2}$ pairs of vertices of G. Let $R(G)$ be the bipartite graph with the partite sets V and V_p such that x in V is connected to a pair {u, v} in V_p if and only if x resolves u and v in G. The resolving graph of G is known as $R(G)$. The metric dimension of G is the minimum cardinality of a subset S of V such that the neighbourhood $N(S)$ of S in $R(G)$ is V_p.

Assume that V = {v_1, v_2,..., v_5} and V_p = {s_1, s_2,..., $s_{\binom{5}{2}}$ }. Let A = (a_{ij}) be the $\binom{5}{2} \times 5$ matrix with

$$a_{ij} = \begin{cases} 1\ if\ s_i v_j \in E(R(G)) \\ \quad 0\ otherwise \\ \end{cases}$$

for $1 \leq i \leq \binom{5}{2}$ and $1 \leq j \leq 5$.

Garey [3] has shown that it is an NP- complete problem to evaluate the metric dimension of an arbitrary graph. The metric dimension problem occurs in several different fields, such as robot navigation [4], telecommunication [5], and geographical routing protocols [6].

Metric dimension are determined theoretically for several graphs in the literature. Metaheuristic algorithms (MAs) have become prevalent in many applied disciplines in recent decades because of their higher performance and lower required computing capacity and time than deterministic algorithms in many optimization problems [7]. Only a few algorithms have been proposed to compute heuristically the metric dimension problem [8–11]. In [8], Kratica et al. have developed a genetic algorithm that can be tested on various types of graph instances to find the metric dimension of graphs. In [9], an algorithm has been proposed for determining the metric dimension of a graph. In [10], a variable neighborhood search approach has been proposed for tackling metric dimension and minimal doubly resolving set problems in order to enhance the current upper bounds. In [11], a particle swarm optimization has been proposed for solving metric dimension problem.

Recently, the slime mould algorithm (SMA) has been proposed [12]. SMA is inspired by a unique slime mould, i e., Physarum polycephalum, which is an organism that can live freely as a single cell but can also form communicating aggregates while searching for food sources. Slime mould uses a population that is randomly distributed to begin its search for food. Once having identified a food concentration during the random search, the slime mould will approach and wrap the food and secrete enzymes to digest it while still maintaining some exploration ability to search for better food sources. To mimic the exploring and exploitation behaviours of slime mould.

SMA has been applied to many applications, such as engineering problems [13], global optimization [14], wireless sensor networks [15], and optimal reactive power dispatch [16].

The SMA has several new features and a unique mathematical model that employs adaptive weights to simulate the process of creating positive and negative feedback on the propagation wave of slime mould based on a bio-oscillator to form the optimal path for connecting food with excellent exploratory ability and exploitation propensity. SMA solves the optimization problem by mimicking the foraging and movement behaviors of slime mould. It can successfully find a promising, optimal solution. However, it still has several drawbacks, including the inconsistent convergence speed because of being trapped in local minima, the imprecise search accuracy, and the inability to find a locally optimal solution when faced with difficult optimization problems.

Simulated annealing is a popular local search meta-heuristic capable of escaping from local optima. Its simplicity of implementation, convergence properties, and utilization of hill-climbing moves to avoid local optima. The SMA is integrated with the SA to flee from local optima because it can accept a worse solution based on probability. Moreover, the bitwise operations can increase the diversity of the population. Also, see more details in the literature [17-20].

In this paper, a new hybrid algorithm that combines both simulated annealing and the slime mould algorithm is presented. The developed algorithm, which is called SMA_SA is used to solve the metric dimension problem and is compared with the original Slime Mould Algorithm (SMA), Simulated Annealing (SA), and Particle Swarm Algorithm (PSO). The developed approach achieves better results compared to other algorithms in terms of time and minimum resolving set. However, as with all optimizers, it may have a bit of slow convergence in high- dimensional problems.

The paper is organized as follows: Section 2 includes the main features of SA. Section 3 explains SMA.

Section 4 contains a hybrid SMA_SA for the metric dimension problem. In Section 5 contains results and discussions on graphs such as the k-home chain graph, the tadpole graph, the alternate triangular snake graph and the mirror graph. Finally, the work is concluded in Section 6.

2. Simulated Annealing

Simulated Annealing (*SA*) [21] is a global stochastic optimization technique which has been applied to a wide range of combinatorial optimization problems [22-28]. It is a variant of local search that enables for controlled upward movement acceptance. If the cost of a new solution is less than the cost of the present solution, the usual SA algorithm will adopt it.

As can be seen, the SA has numerous elements, including a Problematic-oriented method. The creation of a suitable annealing algorithm is not easy, and it usually entails three steps: (1) neighborhood structure, (2) cost function and (3) cooling schedule.

3. Slime Mould Algorithm

The SMA [12] is similar to other swarm-based algorithms in that the individuals would spread out across the research domain and are directed towards the global optimum during iterations. The method of optimization would be divided into several steps, including initialization, searching and exploiting.

3.1. The Initializing Procedure

Individuals in swarms would be automatically and uniformly initialized all over the domain [LB, UB]

$$x_1= r_1 (\mathrm{UB} - \mathrm{LB}) + \mathrm{LB} \quad (1)$$

where r is a random number from the gauss distribution. For each parameter, such as population size, the maximum allowed iteration time maxIter and etc, all values should be initialized and optimized.

3.2. The Iterations

During the exploration and exploiting procedure, iterations would be carried out and the locations of each individual would be changed and oriented to the global optimum:

$$x_t(t+1) = \begin{cases} r_2.(UB-LB)+LB\; r_2 < z \\ x_b + v_b.[W.x_A(t) - x_B(t)]\; r_3 < p \\ v_c.x_i(t)\; p \le r_3 \le 1 \end{cases} \quad (2)$$

where x_i *(t)* and x_i *(t+1)* denote the position in the current iteration of the *i-th* candidate in swarms t and next iteration $t+1$; r_2 is another random number in Gauss distribution; z is a proportional number to randomly pick certain candidates as defaults to restart the initialization, z=0.03.

In the current iteration, $x_A(t)$ and x_B (t) are two randomly chosen candidates. v_a and v_b are two more random numbers in uniform distribution with intervals of [-a, a] and [-b, b] respectively. Here, a and b are two variables relating to the number of iterations and the maximum allowable iteration time:

$$a = \mathrm{atanh}\left(1 - \frac{t}{maxIter}\right) \quad (3)$$

$$\mathrm{b}= 1 - \frac{t}{maxIter} \quad (4)$$

Another proportional number for regulating the collection of branches is p, which is vital to the DF global best fitness value:

$$\mathrm{p}=tanh|S_{i\_}\; DF| \quad (5)$$

Weights W is the most hard parameter, it is a matrix that can be calculated as follows:

$$W_{si(i)} = \begin{cases} 1 + r_4.\log\left(1 + \frac{bF-Si}{bF-wF}\right)\; condition \\ \\ 1 - r_4.\log\left(1 + \frac{bF-Si}{bF-wF}\right)\; other \end{cases} \quad (6)$$

Where bF, wF are the best and worst fitness values for all fitness values for all si (*i=1,2,…., n*) and si is the sort of all of the fitness values for each individual.

$$\mathrm{si} = \mathrm{sort(S)} \quad (7)$$

Algorithm 1 Pseudo-code of SMA

```
Initialize the parameters $popsize$, $Max\_iteraition$;
Initialize the positions of slime mould $X_i$ ($i$ = 1,2, … , $n$);
While ($t \le Max\_iteraition$)
    Calculate the fitness of all slime mould;
    $update\ bestFitness, Xb$
    Calculate the W by Eq. (6);
For $each$ search portion
    $update\ p, vb, vc$;
    $update\ positions\ by$ EP. (2);
End For
$t = t + 1$;
End While
Return $bestFitness, Xb$;
```

4. SMA_SA Algorithm

SA is a single-based algorithm which is used to enhance the leader position which will have a great impact to the whole generation.

4.1. Architecture of SMA_SA

SMA has a good exploration and exploitation abilities in solving optimization problems. However, in complex and high-dimensional SMA may fall in local optimal regions. To solve such a problem SMA algorithm is hybridized with SA. The SA is used to enhance the fitness of the best agent if it falls in suboptimal region which will lead to the enhancement of all individuals. The pseudo code of this algorithm is given in Alg. 2.

Algorithm 2 Pseudo-code of SMA_SA

```
Initialize the parameters $popsize$, SMA$Max\_iteraition$, SAMax_iteration, Number of vertices;
Create an initial generation in which each candidate represent a feasible resolving set
Initialize the positions of slime mould $X_i$ ($i$ = 1,2, … , $n$);
While ($t \leq$ SMA$Max\_iteraition$)
      Calculate the fitness of all slime mould;
      $update\ bestFitness, Xb$
      Calculate the W by Eq. (6);
For $each$ search portion
     $update\ p, vb, vc$;
    $update\ positions\ by$ EP. (2);
 End For
$t1$=0
While ($t1 \leq$ SA$Max\_iteraition$)
Update best agent using SA algorithm.
End While
 $t = t + 1$;
End While
Return $bestFitness$ (the minimum resolving set);
```

4.2. The Hybrid SMA_SA

In this section, the SMA will be combined with the SA to solve the metric dimension problem, where the SA is used to pay attention to the best so-far regions obtained by the SMA.

Specifically, SMA is applied at the outset to use up its exploration capability within the first half of the iteration for searching the search space. SA then starts to pay attention to searching for a better solution using the high-ability of SA that will exploit around the best-so-far if the distance between the fitness value of the current solution and the best-so-far solution is more than particular value created randomly.

Otherwise, we propose a hybrid algorithm SMA-SA is collaborative combination of the SMA and SA techniques. In this hybrid, firstly, SMA has many new features with a unique mathematical model that uses adaptive weights to simulate the mechanism of generating positive and negative feedback of the propagation wave of slime mould based on a bio-oscillator to form the optimal path for linking food with excellent exploratory ability and exploitation propensity. Secondly, the global minimum can be defined by the SA algorithm using stochastic searching technology from the means of probability and it assures that a global minimum can be found when the space of the parameter is sampled indefinitely many times during duration annealing. The SA is integrated with the slime Mould algorithm to flee from local optima because it can accept a worse solution based on a probability. In addition, the bitwise operations can increase the diversity in the population. Different and large dimension sizes of the datasets are used to study the feasibility of the proposed algorithm.

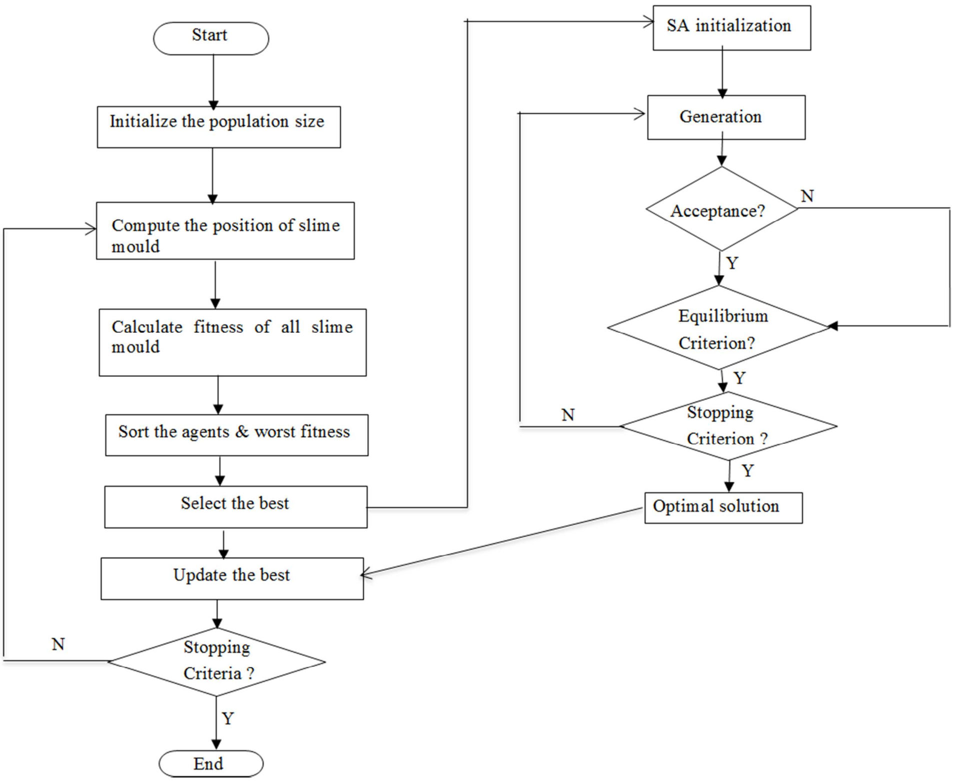

Figure 2. *SMA_SA.*

5. Results and Discussions

The metric dimension problem is an NP-hard combinatorial optimization problem with a solution space that grows exponentially with the problem dimension, It is generally solved using a heuristic technique. Heuristic techniques do not ensure that the best solution will be discovered, but they do provide good solutions for a variety of problems in a reasonable length of time. Generalized frameworks for producing heuristics, have lately gained popularity as a way to solve large combinatorial optimization problems. Metaheuristic techniques relies on two ideas to avoid being trapped in local minima: local search with globalization mechanisms and population search. We use certain metaheuristic algorithms to determine the metric dimension of some graphs in this section.

5.1. Experimental Results

In this section, we present the computational results on graphs such as *k*-home chain graph, tadpole graph, alternate triangular snake graph and mirror graph. Also, we show the superiority of a hybrid SMA_SA on the SMA, SA and PSO according to metric dimension. On the other hand, the proposed SMA_SA outperforms SMA, SA and PSO according to CPU time.

All parameters setting of the all algorithms have been taken from the original papers. All algorithms tests were performed on a Windows 10 Pro 64-bit operating system; the processor was an Intel Core i5 running at 4 GB of RAM and the code was implemented in MATLAB 2018a. All experiments use the same number of iterations and the same criteria.

Table 1. *Parameter setting.*

Parameter	Value
Population size	30
Number of iterations	500
Number of runs	20

1) The SMA_SA, SMA, SA and PSO have been run 20 times for each graph and the results are summarized in Tabs 2–5. The tables are organized as follows:
2) The columns contain the test the number of nodes *N*, edges *M*, metric dimension *Md*, the CPU time (*t*) used to indicate the exact metric dimension and iteration: the average number of iterations for finishing the algorithms to achieve best solution respectively.

Tables 2-5: Comparison for SMA_SA, SMA, SA and PSO algorithms for computing the metric dimension of *k*-home chain graph, tadpole graph, alternate triangular snake graph and mirror graph. (*N* is the number of vertices).

Table 2. *Results on k-home chain graph.*

***k*-home chain graph**													
N	***M***	**SMA_SA**			**SMA**			**SA**			**PSO**		
		Md	***t* (sec)**	**iteration (generation)**	***Md***	***t* (sec)**	**iteration**	***Md***	***t* (sec)**	**iteration**	***Md***	***t* (sec)**	**iteration**
5	6	2	0.03	1	2	0.2	1	2	0.5	1	2	0.39	1
8	12	2	1.15	1	2	2.43	1	2	5.31	2	2	3.98	1
11	18	2	3.27	2	2	6.33	5	2	10.99	13	2	7.15	3
14	24	2	5.46	3	2	11.88	11	2	23.45	27	2	10.61	16
17	30	2	8.74	3	2	19.69	23	3	45.32	59	2	22.17	13
20	36	2	10.96	4	2	32.11	39	4	72.45	74	2	36.07	47
23	42	2	37.93	5	2	24.25	44	5	222.78	96	3	54.25	52
26	48	2	13.12	5	2	107.83	56	5	398.33	151	3	91.22	83
29	54	2	29.77	7	2	98.34	63	6	536.92	192	3	112.76	127
32	60	2	35.14	8	2	182.56	69	7	787.21	217	4	198.09	192
35	66	2	44.9	11	2	221.67	85	8	856.8	281	4	312.15	135
38	72	2	125.45	15	3	109.13	97	9	1111.67	303	4	454.28	109
41	78	2	77.8	20	3	355.81	118	9	1321.44	327	5	579.03	173
44	84	3	111.32	22	4	429.28	131	10	1643.28	363	6	765.89	145
47	90	4	82.87	25	5	512.55	179	11	1986.75	395	6	833.52	209

Table 3. *Results on tadpole graph $T_{3,M}$.*

Tadpole graph													
N	***M***	**SMA_SA**			**SMA**			**SA**			**PSO**		
		Md	***t* (sec)**	**iteration (generation)**	***Md***	***t* (sec)**	**iteration**	***Md***	***t* (sec)**	**iteration**	***Md***	**t(sec)**	**iteration**
4	4	2	0.03	1	2	0.44	1	2	2.45	1	2	0.28	1
5	5	2	0.09	1	2	2.7	1	2	5.12	1	2	3.95	2
6	6	2	1.3	2	2	4.84	2	2	12.37	4	2	5.19	5
7	7	2	1.66	2	2	6.98	2	2	23.88	11	2	8.06	8
8	8	2	2.87	2	2	10.33	3	2	47.94	24	2	17.85	13
9	9	2	4.93	3	2	19.64	4	2	81.36	42	2	29.03	6
10	10	2	6.65	4	2	31.87	7	3	98.61	73	3	54.89	24

Tadpole graph														
		SMA_SA			SMA			SA			PSO		
N	*M*	*Md*	*t* (sec)	iteration (generation)	*Md*	*t* (sec)	iteration	*Md*	*t* (sec)	iteration	*Md*	t(sec)	iteration
11	11	2	8.12	4	2	38.11	19	3	135.42	98	3	49.63	51
12	12	2	31.49	6	2	24.27	34	4	196.77	127	3	104.18	18
13	13	2	15.11	6	3	79.65	52	5	248.35	151	3	163.32	74
14	14	2	20.94	11	3	113.99	61	5	294.13	189	4	201.28	97
15	15	2	78.39	12	3	47.32	75	6	361.71	214	4	219.11	110
15	15	2	55.63	15	3	198.56	94	7	439.15	278	5	256.02	134
16	16	3	36.32	21	4	252.17	69	8	491.18	319	5	337.24	171
17	17	3	44.83	27	4	293.81	156	9	557.19	393	6	389.15	206

Table 4. *Results on alternate triangular snake graph.*

Alternate triangular snake graph													
		SMA_SA			SMA			SA			PSO		
N	*M*	*Md*	*t* (sec)	iteration (generation)	*Md*	*t* (sec)	iteration	*Md*	*t* (sec)	iteration	*Md*	t(sec)	iteration
3	3	2	0.02	1	2	0.15	1	2	4.18	1	2	2.83	1
6	7	2	0.19	1	2	3.39	1	2	15.86	1	2	5.04	2
9	11	2	5.71	2	2	10.45	3	2	61.93	6	2	14.59	5
12	15	2	12.45	2	2	21.56	8	2	118.45	17	2	35.18	13
15	19	2	29.53	3	2	27.14	17	2	193.82	29	2	28.13	10
18	23	2	37.65	3	2	73.22	25	3	306.41	35	3	85.46	37
21	27	2	42.11	5	2	106.49	34	4	445.15	69	3	103.15	43
24	31	2	63.99	16	2	145.78	51	4	491.72	115	3	188.23	62
27	35	2	89.67	7	3	181.99	73	5	618.11	154	4	224.17	28
30	39	2	113.34	7	3	218.51	88	6	776.84	189	4	257.06	119
33	43	2	99.72	10	3	276.64	112	7	929.71	213	5	391.42	138
36	47	2	175.21	13	4	329.32	36	9	1087.56	287	5	459.13	91
39	51	3	226.49	25	4	387.59	133	9	1419.75	316	6	563.01	152
42	55	3	199.37	37	4	419.72	49	11	1502.39	342	6	699.84	197
45	59	4	157.81	28	5	502.39	82	12	1803.72	376	7	791.32	163

Table 5. *Results on mirror graph.*

Mirror graph													
N	*M*	SMA_SA			SMA			SA			PSO		
		Md	*t* (sec)	iteration	*Md*	*t* (sec)	iteration	*Md*	*t* (sec)	iteration	*Md*	*t* (sec)	iteration
4	4	2	0.03	1	2	0.17	1	2	4.72	1	2	1.53	1
6	7	2	0.14	1	2	3.26	1	2	23.12	1	2	4.72	3
8	10	2	4.36	2	2	7.69	2	2	51.72	4	2	10.04	6
10	13	2	12.98	2	2	10.31	3	2	109.37	11	2	27.19	2
12	16	2	16.76	3	2	29.81	7	2	185.47	19	2	51.87	19
14	19	2	35.43	14	2	52.97	32	2	296.11	28	2	94.15	25
16	22	2	27.16	26	2	79.13	39	3	365.35	51	3	128.11	48
18	25	2	61.79	31	2	101.75	54	4	461.19	103	4	199.02	71
20	28	2	94.32	12	3	88.19	17	5	578.14	142	4	271.19	79
22	31	2	85.14	27	3	161.33	88	5	646.84	171	5	379.04	54
24	34	2	142.87	42	3	117.38	53	6	829.59	199	5	447.25	123
26	37	2	125.62	21	3	199.13	144	7	1017.21	248	6	431.08	168
28	40	3	206.34	73	4	381.78	91	8	1325.52	301	6	495.63	191
30	43	3	178.11	49	4	476.39	72	10	1502.23	324	7	682.54	218
32	46	3	195.13	38	4	592.75	105	12	1701.38	352	8	723.84	246

5.2. Comparison

To further demonstrate the excellence of SMA_SA, we choose SMA, SA and PSO algorithms to conduct experiments under the same conditions and compared the results.

The results on graphs are shown in Tables 2, 3, 4 and 5, which indicate that SMA_SA algorithm, outperforms other algorithms on graphs, reaching 82.87 sec in SMA_SA, 512.55 sec in SMA, 1986.25 sec in SA and 833.52 in PSO for *k*-home chain graph, and 44.83 sec in SMA_SA, 293.81 sec in SMA, 557.19 sec in SA and 389.15 in PSO for tadpole graph and 157.51 sec in SMA_SA, 502.39 sec in SMA, 1803.73 sec in SA and 791.32 in PSO for alternate triangular snake graph and 195.13 sec in SMA_SA, 592.75 sec in SMA, 1701.38 sec in SA and 723.84 in PSO for mirror graph. It proves the correctness and superiority of SMA_SA.

Figures 2, 3, 4 and 5 show that the superiority of the proposed SMA_SA on the SMA, SA and PSO according to the metric dimension and CPU time. For example, the metric dimension for tadpole graph at *N*= 17 is 3, but it is 4, 9 and 6 by SMA_SA, SMA, SA and PSO respectively. All figures show that the superiority of the proposed SMA_SA on the SMA, SA and PSO.

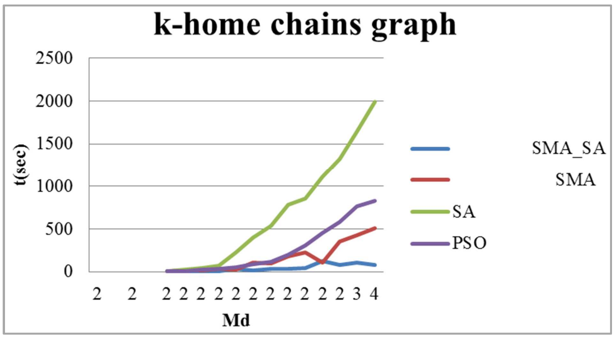

Figure 3. *Comparison between Md and t(sec) of SMA_SA, SMA, SA and PSO for computing the metric dimension of k-home chain graph.*

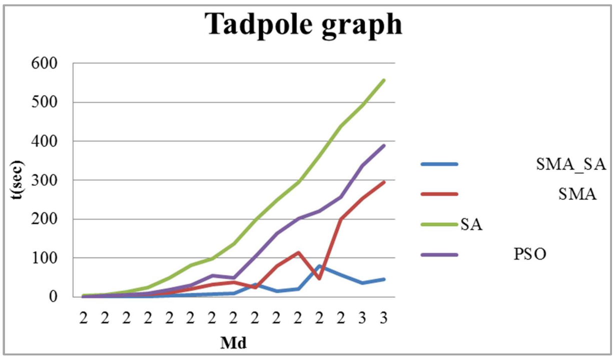

Figure 4. *Comparison between Md and t(sec) of SMA_SA, SMA, SA and PSO for computing the metric dimension of tadpole graph.*

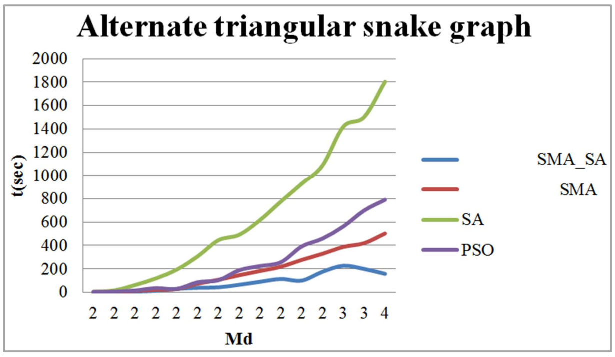

Figure 5. *Comparison between Md and t(sec) of SMA_SA, SMA, SA and PSO for computing the metric dimension of alternate triangular snake graph.*

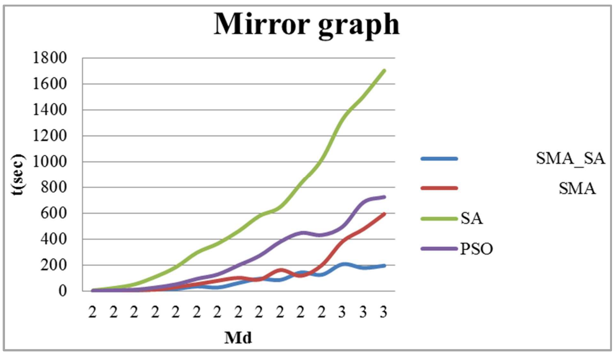

Figure 6. *Comparison between Md and t(sec) of SMA_SA, SMA, SA and PSO for computing the metric dimension of mirror graph.*

6. Conclusion

In this paper, we proposed a hybrid algorithm SMA-SA that combines SA with SMA for determining the metric dimension of graphs. The experimental results analysis with pure SMA, SA, and PSO algorithms confirms the effectiveness of the proposed algorithm SMA-SA, for solving metric dimension problem.

References

[1] P. J. Slater, "Leaves of trees," *Congressus Numerantium*, vol. 14, pp. 549-559, 1975.

[2] F. Harary and R. A. Melter, "On the metric dimension of a graph," *Ars Combinatoria,* vol. 2, pp. 191–195, 1976.

[3] M. R. Garey, Computers and Intractability: A Guide to the Theory of NP-Completeness WH Freeman and Company, New York, 1979.

[4] S. Khuller, B. Raghavachari and A. Rosenfeld, "Landmarks in graphs," *Discrete applied mathematics*, vol. 70, no. 3, pp. 217-229, 1996.

[5] Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann *et al.*,"Network discovery and verification," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2168-2181, 2006.

[6] K. Liu and N. Abu-Ghazaleh, "Virtual coordinates with backtracking for void traversal in geographic routing," In *International Conference on Ad-Hoc Networks and Wireless,* Springer, Berlin, Heidelberg, pp. 46-59, 2006.

[7] H. Faris, A. Z. Ala'M, A. A. Heidari, I. Aljarah, M. Mafarja, et al, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks," *Information Fusion*, vol. 48, pp. 67-83, 2019.

[8] J. Kratica, V. Kovačević-Vujčić and M. Čangalović, "Computing the metric dimension of graphs by genetic algorithms," *Computational Optimization and Applications*, vol. 44, no. 2, pp. 343-361, 2009.

[9] F. Muhammad and L. Susilowati, "Algorithm and computer program to determine metric dimension of graph," *In Journal of Physics: Conference Series*, IOP Publishing, vol. 1494, no. 1, pp. 012018, 2020.

[10] N. Mladenović, J. Kratica, V. Kovačević-Vujčić and M. Čangalović, "Variable neighborhood search for metric dimension and minimal doubly resolving set problems," *European Journal of Operational Research*, vol. 220 no. 2, pp. 328-337, 2012.

[11] D. T. Murdiansyah,"Computing the metric dimension of hypercube graphs by particle swarm optimization algorithms," In *International Conference on Soft Computing and Data Mining*, Springer, Cham, pp. 171-178, 2016.

[12] S. Li, H. Chen, M. Wang, A. A. Heidari and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300-323, 2020.

[13] S. Yin, Q. Luo, Y. Zhou, “EOSMA: an equilibrium optimizer slime mould algorithm for engineering design problems,” *Arabian Journal for Science and Engineering*, pp. 1-32, 2022.

[14] S. Yin, Q. Luo, Y. Du, Y. Zhou, “DTSMA: Dominant swarm with adaptive t-distribution mutation-based slime mould algorithm,” *Mathematical Biosciences and Engineering*, vol. 19, no. 3, pp. 2240-2285, 2022.

[15] S. Yin, Q. Luo, G. Zhou, Y. Zhou, B. Zhu, “ An equilibrium optimizer slime mould algorithm for inverse kinematics of the 7-DOF robotic manipulator, ” *Scientific Reports*; vol. 12, no. 1, pp. 1-28, 2022.

[16] Y. Wei, Y. Zhou, Q. Luo, W. Deng, “Optimal reactive power dispatch using an improved slime mould algorithm, ” Energy Reports, vol. 7, pp. 8742-8759, 2021.

[17] B. Mohamed and M. Amin,"The Metric Dimension of Subdivisions of Lilly Graph, Tadpole Graph and Special Trees", vol. 12, no. 1, pp. 9-14, 2023.

[18] B. Mohamed, L. Mohaisen and M. Amin,"Computing Connected Resolvability of Graphs Using Binary Enhanced Harris Hawks Optimization," *Intelligent Automation & Soft Computing*, vol. 36, no. 2, 2023.

[19] B. Mohamed, L. Mohaisen and M. Amin," Binary Equilibrium Optimization Algorithm for Computing Connected Domination Metric Dimension Problem," Scientific Programming, 2022.

[20] B. Mohamed and M. Amin,"Domination Number and Secure Resolving Sets in Cyclic Networks,"*Applied and Computational Mathematics*, vol. 12, no. 2, pp. 42-45, 2023.

[21] R. W. Eglese,"Simulated annealing: a tool for operational research,” *European journal of operational research*, vol. 46*,* no. 3, pp. 271-281, 1990.

[22] D. Abramson, "Constructing school timetables using simulated annealing: sequential and parallel algorithms,” *Management science*, vol. 37, no. 1, pp. 98-113, 1991.

[23] D. Abramson, M. Krishnamoorthy and H. Dang, "Simulated annealing cooling schedules for the school timetabling problem,” *Asia Pacific Journal of Operational Research*, *16*, pp. 1-22, 1999.

[24] E. Poupaert and Y. Deville, "Simulated annealing with estimated temperature,” *AI Communications*, vol. 13, no. 1, pp. 19-26, 2000.

[25] J. Thompson and K. A. Dowsland, "General cooling schedules for a simulated annealing based timetabling system,” in *Proc. ICPTAT*, Springer, Berlin, Heidelberg, pp. 345-363, 1995.

[26] 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*), vol. 15, no. 1, 2023.

[27] B. Mohamed,"A Comprehensive Survey on the Metric Dimension Problem of Graphs and Its Types, *International Journal of Theoretical and Applied Mathematics*, vol. 9, no. 1, 2023.

[28] B. Mohamed, Metric Dimension of Graphs and its Application to Robotic Navigation, *International Journal of Computer Applications*, vol. 184, no. 15, 2022.