A hybrid population-based algorithm for solving the Minimum Dominating Set Problem

Belkacem Zouilekh[1] **and Sadek Bouroubi**[1,*]

[1] *LIFORCE Laboratory, Department of Operations Research, Faculty of Mathematics, University of Sciences and Technology Houari Boumediene, P.B. 32 El-Alia, 16111, Bab Ezzouar, Algiers, Algeria.*
E-mail: ⟨{*bzouilekh, sbouroubi*}*@usthb.dz*⟩

Abstract. The Minimum Dominating Sets (MDS) problem is pivotal across diverse fields like social networks and ad hoc wireless networks, representing a significant NP-complete challenge in graph theory. To tackle this, we propose a novel hybrid cuckoo search approach. While cuckoo search is renowned for its wide search space exploration, we enhance it with intensification methods and genetic crossover operators for improved performance. Comprehensive experimental evaluations against state-of-the-art techniques validate our approach's effectiveness.

Keywords: genetic algorithm, hybrid cuckoo search algorithm, minimum dominating set

Received: March 16, 2024; accepted: June 5, 2024; available online: October 7, 2024

DOI: 10.17535/crorr.2024.0015

1. Introduction

Given an undirected graph $G = (V, E)$, a subset S of V is named a dominating set of G if each vertex $v \in V$ is either an element of S or is adjacent to an element of S. Such a subset is called a dominating set of G, and we say S dominates G or G is dominated by S. The minimum cardinality of a dominating set in G is denoted by $\gamma(G)$. If S dominates G, such as $|S| = \gamma(G)$, then S is called a Minimum Dominating Set, or MDS for short [10]. As an example, Figure 1 shows a subgraph of Twitter tweets with 85 vertices that spread specific 5G false news that is linked directly to the COVID-19 pandemic [20], with nodes denoting Twitter users and edges representing follower relationships. The MDS obtained by our approach is highlighted in red. The original status of the publisher is represented by the vertices of the dominating set, which are marked in red. If we assume the simplest scenario: followers, marked in black, cannot retweet and content cannot spread out of followers, only 2 users could probably spread misinformation among the other 83 users.

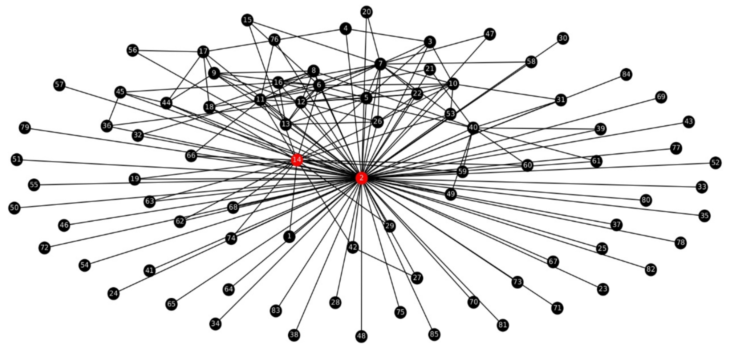


Figure 1: *Representations of a real-induced sub-graph from Twitter.*

*Corresponding author.

The minimum dominating sets and their variations, such as the minimum connected dominating set and the minimum weighted dominating set, have pervasive applications in a wide range of disciplines, besides routing in ad hoc wireless networks [26], sensor networks, and MANETs [3]. In addition, the MDS can be employed to determine the main subject sentences for document summary via sentence network design [27]. Furthermore, the MDS may be used to model and investigate the positive impact of social networks [7]. Moreover, controlling the spread of epidemics [22] also involves early diagnosis and control of epidemic spreading in various areas of human society, such as virus spread in computer systems, misinformation (as shown in Figure1), and content diffusion via social media [30]. In this paper, we address the minimum dominating set problem by proposing a new Hybrid Cuckoo Search Algorithm, henceforth called HCSA-MDS, that combines the cuckoo search algorithm and the genetic crossover operator with intensification schemes as well as repairing and cleaning to achieve effective exploration and exploitation.

1.1. Paper structure

The remainder of this paper is partitioned into many sections. In Section 2 that follows, we present some linked works, including exact algorithms and heuristics, applied to solving the MDS problem. Section 3 goes into detail about the Cuckoo Search Algorithm. The hybrid CSA algorithm is then presented and described in Section 4. Section 5 summarizes the results of the computation. The paper is finally concluded in Section 6.

2. Related works

The MDS problem is an NP-Hard combinatorial problem [8], and it has been thoroughly investigated using exact (exponential-time) techniques. The best known exact algorithm for the MDS problem performs in $\mathcal{O}(1.4864^n)$ time and polynomial space, constructed through the measure and conquer approach by Y. Iwata [14]. Unfortunately, the exact techniques that execute at an exponential scale are only possible for limited-size networks, severely limiting their effective uses. As a result, scientific researchers have mainly concentrated on stochastic computational heuristics and, lately, metaheuristics. Hence, many heuristic approaches have been adopted in the state of the art to handle the MDS problem, such as[16, 25, 2, 15]. Moreover, L. A. Sanchis [19] performed experimental research on different heuristic approaches in this perspective; he thoroughly investigated many greedy methods for the MDS problem. After that, Ho et al. [13] presented ACO-TS, an improved Ant Colony Optimization metaheuristic that integrates a technique for stimulating the building of different solutions based on a concept adopted from genetic algorithms called tournament selection. Subsequently, in [11], Hedar and Ismail presented a hybrid genetic algorithm referred to as HGA-MDS that employs a local search characterized by three intensification techniques. Furthermore, in [12], Hedar and Ismail also suggested a SAMDS metaheuristic addressing the MDS problem by employing a Stochastic Local Search (SLS) algorithm for strengthening a solution by looking for a stronger one in its near area. The SLS is enhanced by using a simulated annealing process. Recently, Abed and Rais [1] introduced a hybrid population-based technique known as the Hybrid Bat Algorithm, which is rooted in microbat bio-sonar characteristics and simulated annealing. The fact that SA is efficient in exploitation and the bat algorithm has a high capacity for the exploration of large regions in the search space helps to ensure a good balance between intensification and diversification in the search methodology.

3. Cuckoo Search Algorithm

The Cuckoo Search Algorithm (CSA), based on the fascinating breeding behavior of particular cuckoo species, such as brood parasitism, is one of the most recently developed metaheuristics

[9]. To describe the Cuckoo Search for clarity, Yang and Deb use the following three idealized principles [28]:

1. One egg is laid by each cuckoo at a time, and it deposits it in a nest that is selected at random.
2. The best nests with the highest quality eggs (solutions with the highest fitness) will be passed on to future generations.
3. The number of host nests is fixed, and a host has a probability of discovering an alien egg of $P_a \in [0, 1]$. In this scenario, the host bird can either discard the egg or leave the nest and create a new one at a different site.

A solution's fitness can simply be proportional to the objective function's value. A cuckoo egg signifies a new solution, and each egg in a nest indicates a solution. The goal is to replace the poor solutions in the nests with new and maybe improved ones (cuckoos). This approach can be expanded to a more sophisticated scenario of several eggs representing a set of solutions in each nest. Therefore, we will take the simplest approach possible, with each nest containing only one egg. The CSA pseudo-code is shown in Algorithm 1 based on these criteria.

Algorithm 1: Cuckoo Search Algorithm: CSA

```
   Input: Problem instance $s$
   Output: The best possible solution $s^*$
   Initialization:
   Initialize the population of $m$ host nests (solutions) $s = (s_1, \dots, s_m)$;,
   maxGen : Maximum number of generations,
   t ← 0.
1  while $t \leq maxGen$ do
2      Get a cuckoo (say $x_i$) at random using Lévy flights and evaluate its fitness $f(x_i)$;
3      Choose one of $n$ (say $y_i$) nests at random.
4      if $f(x_i) > f(y_i)$ then
5          Replace the old nest $x_i$ by the new one $y_i$;
6      else
7          Continue
8      A portion of the worst nests $p_a$ are removed and replaced with new ones.
9      Sort the solutions and choose the best one.
10     Refresh the current best solution.
11     $t = t + 1$
12 return $s^*$
```

After generating the initial population, CSA generates new solutions x^{i+1} associated to each cuckoo i in each iteration t by a random walk via Lévy flight:

$$x_i^{t+1} = x_i^t + \alpha \oplus L\acute{e}vy(\lambda) \tag{1}$$

Lévy flights are a type of random walk named after the French mathematician Paul Lévy [5], in which α above in the equation denotes the step size, which should correspond to the problem's interests (most of the time $\alpha = 1$), and the term product $\oplus$ denotes entrywise multiplications. The step lengths are selected from a probability distribution with a power law tail. Lévy distributions, or stable distributions, are the names given to these probability distributions.

$$L\acute{e}vy \sim u = l^{\lambda}, \tag{2}$$

where $1 < \lambda \leq 3$ and l is the flight length.

The initial purpose of CSA and Lévy flights was to solve continuous optimization problems. Yang and Deb [28] show the outperformance and robustness of CSA over GA and PSO because of its larger search space exploration capacity and fewer parameters to fine-tune than other algorithms. Actually, there is just one parameter P_a, separate from the population size N. CSA has been widely used and shown promising efficiency in a variety of optimization and computational intelligence applications [29]. On the other hand Boumedine and Bouroubi proposed a discrete hybrid cuckoo search algorithm to solve the protein folding problem [4]. In this paper, we present a discrete hybrid CS-based method, called Hybrid Cuckoo Search Algorithm for Minimum Dominating Set (HCSA-MDS), to solve the MDS problem, which is one of the most difficult combinatorial optimization problems. The suggested discrete HCSA-MDS employs an adaptive Lévy flight for the MDS problem.

4. Hybrid CSA for the MDS problem

We describe our approach to tackling the MDS problem in the following section. Our algorithm takes as input a problem instance $G = (V, E)$, in which G is an undirected, connected graph, V is a set of vertices, and E is a set of edges. The HCSA-MDS pseudo-code shown in Algorithm 4 combines the cuckoo search algorithm with the genetic crossover operator and the cleaning with repair technique to exploit the solutions. Lévy flight, with its ability to explore new regions in the search space, is a particularly useful strategy in the diversification phase. The goal of this hybridization is to establish a proper balance between exploration and exploitation. The main structure is shown in Figure 2.

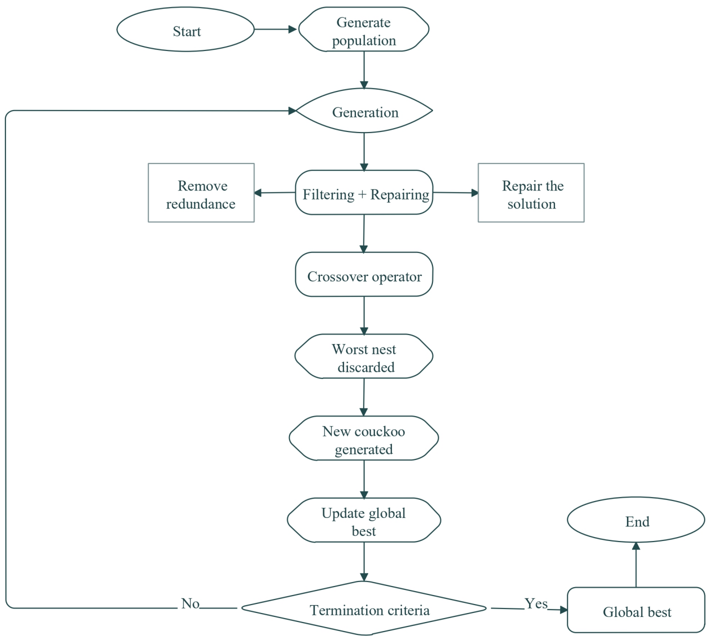


Figure 2: *HCSA-MDS flowchart.*

We first discuss the HCSA-MDS components before fully stating pseudo-code 4.

4.1. Solution and population representation

The HCSA-MDS algorithm starts with a population P that contains a set of m nests (solutions), some of which are non-feasible. As a result, a $0-1$ vector with a dimension equal to the number of nodes in the graph $G = (V, E)$ represents a solution x in P. The i-th node in G is part of the dominating set if the vector's i-th element has a value of 1. While the i-th vector's item has a value of 0, this indicates that a node is not part of the dominant set. Table 1 illustrates the binary representation of the solution to Example 1 presented in Section 1.

1	2	3	⋯	13	14	15	⋯	85
0	1	0	⋯	0	1	0	⋯	0

Table 1: *Binary representation of the best found MDS.*

4.2. A new cuckoo egg's production using the crossover operator

The crossover operator is one of the genetic operators that applies to two parents (solutions) and then chooses at random any of the crossover points p_h, $h \in \{0, 1, ..., n-1\}$ [23]. Two offspring (new solutions) are created by joining the parents at the crossover point for the next generation. However, for any solution from population P that we consider the first parent, we choose a second parent in a random way from the current population. Then, the one point crossover operation is used to generate two offspring. In the proposed algorithm, only the best offspring is chosen for the next stage; see the example in the following Table 2:

Parent	Binary representation		Offspring	Binary representation
First	1010|10110	⇒	First	1010|10010
Second	1111|10010		Second	1111|10110

Table 2: *An illustrative example of a crossover operator with a single point-crossover.*

In the previous example, two offspring were produced randomly by combining the first and second parents at the fourth position. The crossover operation in this example results in two novel solutions, where the first offspring exhibits superior performance in the minimum dominating set problem compared to both its parents and sibling.

4.3. Filtering and Reparation procedures

HCSA-MDS employs a filtering mechanism to enhance the MDS solutions by removing redundant nodes, thereby limiting the size of the dominating set while preserving its coverage [11]. This filtering procedure is detailed in Algorithm 2. Complementing this, the reparation procedure addresses solutions S that fail to cover all nodes. It begins by verifying if the solution is not a dominating set. Subsequently, the next vertex added to the set of dominators is chosen from the non-dominating vertices with the highest degree. This reparation process continues until S constitutes a valid dominating set, as illustrated in Algorithm 3.

Algorithm 2: Filtering procedure

```
  Input: Solution $S$.
  Output: Filtered solution.
1 for each node $x$ in $S$ do
2     if $x = 1$ then
3         Set $x = 0$
4         Compute the new fitness value of the solution $S$
5         if the new fitness value is increased then
6             Update the solution $S$ and set $x = 0$
7         else
8             Reset $x = 1$
9 return $S$
```

```
Algorithm 3: Repairing procedure
   Input: Solution $S$.
   Output: Repaired solution.
1  if There are non-covered nodes in the solution $S$ then
2      Select a node $x$ with maximum degree in non-covered nodes set;
3      Set $x = 1$ and update the solution $S$;
4      Remove $x$ from non-covered nodes and its neighbours;
5  return $S$
```

4.4. Construction of a new solution via Lévy flights

Due to its unique step-length characteristics, the Lévy flight strategy efficiently explores search spaces. To apply this technique to the Minimum Dominating Set (MDS) problem, we relate the step length, representing the length of the subset, to the values generated by Lévy flights. For instance, if S S is an n-dimensional solution, we partition the interval $[0, 1]$ into m subintervals and define the step length ($step_{length}$) as follows:

θ	$\left[0, \frac{1}{m}\right]$	$\left[\frac{1}{m}, \frac{2}{m}\right]$	...	$\left[\frac{m-1}{m}, 1\right]$
$step_{length}$	$\left[1, \frac{\max_l}{m}\right]$	$\left[\frac{\max_l}{m}, \frac{2\max_l}{m}\right]$	...	$\left[\frac{(m-1)\max_l}{m}, \max_l\right]$

where θ is the Lévy flight value acquired and $\max_l = \frac{n}{h}$, $h \in \{1, 2, 3, \ldots, n\}$ is the maximum subset we are able to invert. Let i be the length of the subset chosen at random from the interval generated by Lévy's flight value. Then, from the n-dimensional solution, we choose a random position between 1 and $n - i$, from which the i-inversion begins.

5. Experimental results

HCSA-MDS effectiveness is evaluated against a collection of well-known efficient algorithms for the Minimum Dominating Set problem that have been reported in the literature. The rest of this section will be as follows: First, all benchmarks used are presented. Then, we compare the results obtained by the proposed algorithm with state-of-the-art approaches. We use Wilcoxon's signed-rank test to compare the HCSA-MDS method with other approaches. This nonparametric test is suitable for matched-pair data and detects substantial performance differences. We also use Critical Difference (CD) diagrams to visualize results, as described in [6]. First, a Friedman test checks for significant performance differences among algorithms. If significant, post-hoc analysis follows. CD diagrams rank algorithms on a horizontal axis, with the best near 1. Statistically similar algorithms are connected by bars. For more details, visit this link: https://mirkobunse.github.io/CriticalDifferenceDiagrams.jl/stable.

5.1. Data Sets

The Minimum Dominating Set (MDS) problem commonly uses two benchmark datasets. The first includes 42 random geometric graphs with up to 400 nodes, as outlined in [12]. These networks are generated by randomly placing n nodes within an M × M area, following specific instructions detailed in [13]. Multiple graph instances are created per network based on specified range parameters. The second dataset comprises 20 graphs with 400 and 800 vertices sourced from [12], where optimal dominating sets are known. These graphs are generated using a methodology detailed in [19]. For additional benchmark details, please refer to Table 3.

Network Id	N° of nodes (n)	Range	Area (A)	N° of instances
$N1_A^n$	80	60-120	400×400	7
$N2_A^n$	100	80-120	600×600	5
$N3_A^n$	200	70-120	700×700	6
$N4_A^n$	200	100-160	1000×1000	7
$N5_A^n$	250	130-160	1500×1500	4
$N6_A^n$	300	180-220	2000×2000	5
$N7_A^n$	350	200-230	2500×2500	4
$N8_A^n$	400	210-240	3000×3000	4

(a) *First benchmark.*

Network	n	p	d	N° of instances
$N_{d,0.1}^{400}$	400	0.1	$8, 11, 14, 18, 23$	5
$N_{d,0.3}^{400}$	400	0.3	$3, 5, 8, 11, 14$	5
$N_{d,0.5}^{400}$	400	0.5	$3, 5, 8, 11$	4
$N_{d,0.1}^{800}$	800	0.1	$11, 14, 22$	3
$N_{d,0.3}^{800}$	800	0.3	$3, 5$	2
$N_{d,0.5}^{800}$	800	0.5	$3, 6$	2

(b) *Second benchmark.*

Table 3: *Benchmark data sets.*

5.2. Tuning algorithm parameters

To conduct our experiment, we set the cuckoo search parameters according to the commonly used values in the literature for various optimization problems [28]. For a fair comparison, we selected the population size N and the number of generations *maxGen* based on previous works addressing similar problem. The population size N was set to 40, the number of generations *maxGen* to 100, the probability of the host bird discovering a cuckoo egg Pa to 0.25, and the cuckoo's step length parameters γ and α to 1.5 and 1 respectively.

Algorithm 4: HCSA-MDS Algorithm

```
Input: A graph $G = (V, E)$.
Output: Dominating Set.
Initialization:
Generate an initial population $P$ of $m$ feasible solutions, $S = (x_i, \dots, x_m)$,
maxGen : Maximum number of generations,
Gen $\leftarrow 0$.
1  while $Gen \leq maxGen$ do
2      $i = 1$
3      while $i \leq m$ do
4          Filter the repaired solution to improve its quality
5          Select a random solution $x_j$ from $P$ as second parent
6          Produce a new cuckoo egg $y_j$ using the crossover operator for the selected
             parents $x_i$ and $x_j$
7          if $f(y_j) \geq f(x_i)$ then
8              Replace $x_i$ by the new produced solution $y_j$
9          $i = i + 1$
10     $Gen = Gen + 1$
11 The worst solutions are removed from $P$, and new ones are generated via Lévy flight
     proportional to $P_a \in [0, 1]$
12 Rank the solutions from the best to the worst and find the best one
13 Update the global optimal solution
14 return Dominating Set
```

5.3. The numerical performance

Each benchmark was performed 20 times on the first benchmark and 10 times on the second to ensure a fair comparison. Our HCSA-MDS algorithm was implemented in Python. All testing was conducted on a system with 16 GB of RAM and a 2.6 GHz Intel Core i5 CPU. The source code and datasets for reproducing the experiments are available at https://github.com/elkacem/HCSA_MDS .

5.3.1. Comparative analysis of HCSA-MDS against the state-of-the-art approaches on the first benchmark

We test the suggested HCSA-MDS on the first benchmark against HGA-MDS [11], SAMDS [12], and HBA [1] for the purpose of proving the performance of the HCSA-MDS approach. Check Table 4. The performance of HCSA-MDS is evaluated using the following metrics for each instance:

1. **Best**: The smallest dominating set found across all runs for each graph instance.
2. **Avg.(Std.)**: The average and standard deviation of the best solution values from individual runs for each graph instance.
3. **Worst**: The worst solution found in 10 runs for each graph instance (only for HCSA-MDS).
4. **Rea.**: The number of runs where the optimal solution (domination number) was achieved.

The following assessment can be drawn from the previous experiments (see Table 4):

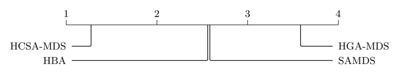
• HCSA-MDS is able to improve the best-known solution in 27 out of 42 graph instances and match the best-known solution for the rest of the 14 instances. And only in one instance ($N5^{250}_{1500}$, 160) did HBA provide a better solution. For the larger graphs, the differences between HCSA-MDS and the other algorithms begin to grow. For example, in instances with 300, 350, and 400 vertices, HCSA-MDS provided better solutions in all graph instances, with a substantial difference, as shown in Figure 4, which simulates the results of a best-solution comparison between HCSA-MDS and the other approaches. In addition, we used the critical difference diagram in Figure 3a to verify these results, showing that HCSA-MDS beats all other algorithms, followed by HBA, SAMDS, and HGA-MDS, which is the worst. Summarizing, we can assert that HCSA-MDS beats the state of the art in terms of solution quality by a large margin.

• Concerning the worst solution obtained over 20 runs, in 17 out of 42 graph instances, HCSA-MDS is able to outperform the currently best solution produced in the state-of-the-art approaches. Furthermore, HCSA-MDS's worst solution matches the best solution provided by other approaches in 19 instances, and in only 6 cases, the best solution provided by the other approaches is better than the worst solution by HCSA-MDS. Table 6 displays Wilcoxon's test statistics, which show that there is a significant difference between HCSA-MDS and other approaches. Finally, Figure 3b shows the critical difference plot, where clearly HCSA-MDS exceeds the state of the art in terms of the worst solution when compared to the best solution yielded by literature approaches.

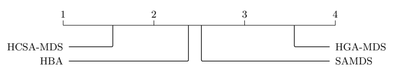
• The results of the standard deviation Table 4 demonstrates that HCSA-MDS outperforms HBA in terms of stability, as it produced stable dominating set values in 12 out of 42 instances, whereas HBA only provided stable values in 6 out of 42 instances. Additionally, the standard deviation for HCSA-MDS in the other cases is negligible, indicating high stability. In comparison, SAMDS performs better than HGA-MDS, which is the least effective, particularly in dense graphs. These findings are supported by the mean ranks presented in the critical difference plots shown in Figure 3c, which confirm that HCSA-MDS is statistically superior to the other algorithms.

Graph Id	HGA-MDS		SAMDS		HBA		HCSA-MDS		
Range	Best	Avg.(Std.)	Best	Avg.(Std.)	Best	Avg.(Std.)	Best	Avg.(Std.)	Worst
N1$_{400}^{80}$ 60	**15**	15.95(0.39)	**15**	16.35(0.67)	16	16(0)	**15**	15(0)	**15**
70	13	14(0.73)	**12**	14.40(1.14)	13	13.30(0.48)	**12**	12(0)	**12**
80	10	10.85(0.59)	10	12(1.03)	10	10.60(0.52)	**9**	9(0)	**9**
90	**8**	8.40(0.50)	**8**	9.60(1.27)	9	9.60(0.52)	**8**	8(0)	**8**
100	**7**	8.20(0.52)	**7**	9.10(0.97)	8	8.40(0.52)	**7**	7(0)	**7**
110	**6**	6.05(0.22)	**6**	7.55(0.69)	**6**	6.90(0.32)	**6**	6(0)	**6**
120	**5**	5.95(0.39)	**5**	7(1.08)	6	6.70(0.48)	**5**	5.10(0.30)	6
N2$_{600}^{100}$ 80	19	19.55(0.85)	**18**	20.55(1.23)	19	19(0)	**18**	18(0)	**18**
90	16	17.20(0.95)	**15**	17.65(1.73)	18	18(0)	**15**	15(0)	**15**
100	14	14.85(0.49)	**13**	14.95(1.05)	14	14.40(0.52)	**13**	13.15(0.36)	14
110	**11**	12.15(0.67)	**11**	12.85(1.23)	12	12.30(0.48)	**11**	11(0)	**11**
120	10	10.15(0.37)	**9**	12(1.49)	11	11(0)	**9**	9.25(0.43)	10
N3$_{700}^{200}$ 70	37	45.65(10.83)	35	39.95(2.09)	34	35.40(0.84)	**32**	32.45(0.59)	34
80	31	33.05(1.32)	29	33.50(1.73)	28	28.20(0.42)	**26**	27.45(0.59)	28
90	26	28.75(2.67)	25	29, 25(1.94)	25	26(0.67)	**23**	23.50(0.50)	24
100	21	24.70(4.94)	20	24.20(1.70)	21	21.80(0.42)	**18**	18.15(0.36)	19
110	19	19.95(0.51)	18	21.40(1.93)	17	17.70(0.48)	**16**	16.65(0.48)	17
120	17	17.30(0.47)	15	19.55(1.43)	15	16.20(0.79)	**14**	14(0)	**14**
N4$_{1000}^{200}$ 100	39	45.25(5.57)	38	41.35(1.87)	36	36.70(0.48)	**35**	36.15(1.01)	38
110	35	37.35(2.06)	33	37.20(2.44)	**30**	30.80(0.42)	**30**	30.35(0.48)	31
120	27	28.90(1.77)	26	30.10(1.45)	26	26.30(0.48)	**23**	23.30(0.46)	24
130	26	27.30(1.13)	25	28.15(1.42)	**23**	24.20(0.63)	**23**	23(0.59)	**23**
140	23	24.35(1.35)	22	25.70(1.84)	22	23(0.67)	**20**	20.95(0.59)	22
150	21	21.45(0.76)	20	23.55(1.43)	19	20.30(0.82)	**17**	17.95(0.74)	19
160	20	21.60(0.94)	19	21.30(1.30)	18	18(0)	**17**	17(0)	**17**
N5$_{1500}^{250}$ 130	55	75.45(24.01)	51	56.05(2.68)	49	50(0.67)	**47**	47.20(0.40)	48
140	48	59.75(12.78)	46	48.65(1.35)	42	42.60(0.70)	**40**	40.85(0.36)	41
150	44	48.30(3.34)	41	44.75(1.71)	**37**	37.90(0.32)	**37**	37.50(0.50)	38
160	38	41.65(3.42)	37	40.90(1.92)	**31**	32.60(1.71)	34	34.15(0.36)	35
N6$_{2000}^{300}$ 180	54	61.20(6.64)	47	52.35(2.41)	44	45.70(0.95)	**43**	43.45(0.50)	44
190	48	55.55(6.35)	46	50.20(2.24)	44	44.40(0.52)	**40**	41.25(0.83)	43
200	41	47.90(5.07)	40	45.25(2.55)	41	41.10(0.32)	**36**	36.50(0.67)	38
210	40	48.60(7.99)	39	43.60(1.70)	37	37.70(0.82)	**34**	34.40(0.58)	36
220	36	39.90(4.34)	36	40.65(1.90)	33	34.10(0.57)	**31**	31.70(0.46)	32
N7$_{2500}^{350}$ 200	67	93.45(23.58)	61	66.35(2.13)	58	58.90(0.32)	**54**	55.85(0.85)	58
210	63	91.20(26.70)	58	61.85(2.18)	53	55(1.05)	**48**	49.70(0.78)	51
220	55	76.85(30.55)	49	55.05(2.31)	51	51.80(0.63)	**44**	45.25(0.70)	47
230	51	67(21.02)	48	54.05(2.42)	45	47.10(1.37)	**43**	43.90(0.83)	45
N8$_{3000}^{400}$ 210	79	115.55(41.21)	75	80.15(3.10)	72	73.30(0.95)	**68**	69.45(0.74)	71
220	77	110.45(39.76)	73	79.25(3.18)	70	71(0.82)	**63**	64.50(0.92)	66
230	73	111.55(38.94)	71	74.10(2.22)	64	64(0)	**60**	60.85(0.57)	62
240	70	103.15(32.02)	63	68.80(2.98)	58	59.30(0.67)	**56**	56.95(0.74)	58

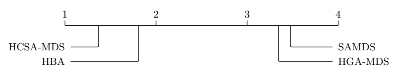
Table 4: *Performance comparison of various algorithms for the first benchmark sets.*



(a) Comparison of the best solutions (**Best**).



(b) HCSA-MDS worst solutions (**Worst**) compared to HBA,SAMDS, and HGA-MDS best solutions.



(c) Standard deviation (**std.**)

Figure 3: *Critical Difference plots evaluating HCSA-MDS, HBA, SAMDS, and HGA-MDS for the first benchmark.*

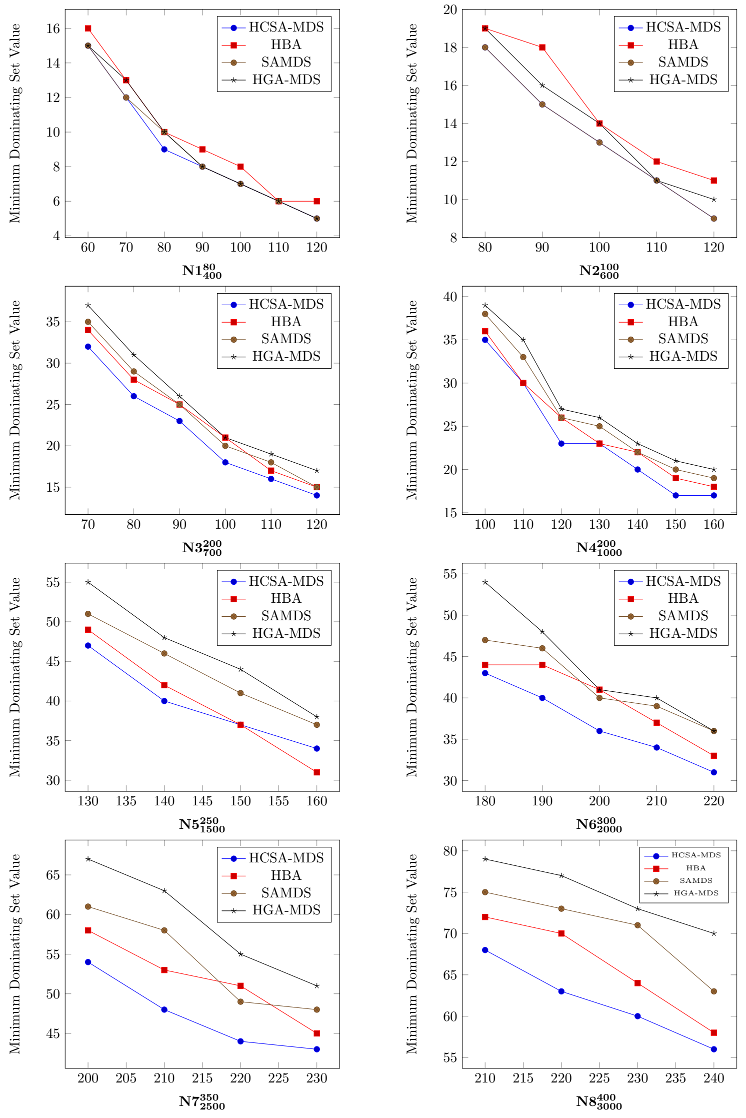


Figure 4: *Best solution analysis of various approaches.*

5.3.2. Numerical results for the HCSA-MDS and various approaches on the second benchmark

We discuss the effectiveness of the suggested HCSA-MDS on the second test case benchmark. Our method's results are evaluated against those of state-of-the-art algorithms. Hence, we compare HCSA-MDS against a variety of greedy algorithms from [19], where Sanshis L described and studied many greedy heuristics designated Greedy, GreedyRev, GreedyRan, GreedyVote, and GreedyVoteGr, which is an updated variant of GreedyVote. According to the evaluations, GreedyVoteGr beats other greedy heuristics when employing the local search process. Similarly, we compare HCSA-MDS with the Hybrid Genetic Algorithm abbreviated HGA-MDS from [11], as well as Simulated Annealing (SAMDS) from the paper [12], which shows better performance than HGA-MDS and the greedy heuristics. The numerical values are presented in Table 5. For each graph instance, the results of various approaches are presented in terms of the average solution (**Avg.**) obtained in 10 runs in each table and how many runs out of 10, the Minimum Dominating Set was reached, which is represented in the column labeled "**Rea.**". Three alternative graph densities are taken: from dense to sparse, 0.5, 0.3, and 0.1. The experiments in Table

Criteria	Algorithm	Z	Sig. (p value)
Optimal Reached (Opt.)	HGA-MDS ***vs.*** GrVoteGr	-3.306	0.001
	SAMDS ***vs.*** HGA-MDS	-2.364	0.018
	HCSA-MDS ***vs.*** HGA-MDS	-3.026	0.002
	HCSA-MDS ***vs.*** SAMDS	-2.555	0.011
Average (Avg.)	HGA-MDS ***vs.*** GrVoteGr	-3.351	0.001
	SAMDS ***vs.*** HGA-MDS	-1.601	0.109
	HCSA-MDS ***vs.*** HGA-MDS	-3.051	0.002
	HCSA-MDS ***vs.*** SAMDS	-2.677	0.007
Worst of HCSA-MDS VS Best of Others	HCSA-MDS vs HBA	-3.304	0
	HCSA-MDS vs SAMDS	-4.634	0
	HCSA-MDS vs HGA-MDS	-5.108	0

Table 6: *Comparison of statistical test results for different algorithms.*

5 shows that HCSA-MDS is the highest-performing algorithm, with absolute distinctions in terms of average solution and reaching the optimal solution. Moreover,almost all instances reached the optimal in all 10 runs except two instances ($\mathbf{N^{400}_{0.3,3}}$,$\mathbf{N^{800}_{0.5,3}}$3) optimal reached 8, 9 times respectively out of 10. To determine the relevance of the differences between the approaches results and HCSA-MDS, we used Wilcoxon's signed-rank test. It only makes sense to consider only the GreedyVoteGr, HGA-MDS, and SAMDS approaches, as they outperform the other greedy algorithms; moreover, the results of the rest of the greedy methods are too similar. The test statistics presented in Table 6 show that there is a significant difference between HGA-MDS and the best-performing algorithm in the Greedy series from [19], which is GreedyVoteGr, in terms of both criteria. Furthermore, HGA-MDS could perform equally well as SAMDS in terms of the mean of the solutions found, although SAMDS outperforms HGA-MDS in terms of the attainment rate of reaching the Minimum Dominating Set. Finally, HCSA-MDS outperforms SAMDS, HGA-MDS, and GreedyVoteGr since it is outperformed by HGA-MDS. To conclude, we can state that HCSA-MDS outperforms state-of-the-art approaches in terms of robustness and stability.

	Greedy	GreedyRev	GreedyRan	GreedyVote	GreedyVoteGr	HGA-MDS	SAMDS	HCSA-MDS	
Graph d	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Avg(Rea.)	Time(s)
$N^{400}_{0.1,d}$8	14.2(0)	16.1(4)	30.9(0)	9.2(4)	8(10)	8(10)	8(10)	8(10)	0.09
11	22.1(0)	25.2(0)	33.2(0)	16.8(0)	15.6(5)	11.1(9)	11.1(9)	11(10)	0.07
14	24(0)	25.6(0)	35.2(0)	19(1)	18.6(3)	14.4(6)	14.2(9)	14(10)	0.07
18	27.3(0)	27.3(0)	38.5(0)	21.8(0)	20.4(4)	18.4(6)	18(10)	18(10)	0.11
23	31.3(0)	28.8(0)	41.8(0)	24.9(0)	24.8(0)	24.2(4)	23(10)	23(10)	0.13
$N^{400}_{0.3,d}$3	6.8(0)	9.9(0)	10.9(0)	6.8(4)	6(4)	3(10)	3.8(8)	3.2(8)	0.81
5	9(0)	10.6(0)	11.7(0)	8.7(0)	8.7(0)	5.3(7)	5.2(8)	5(10)	0.15
8	10.9(0)	11.6(0)	13.3(0)	9.3(0)	9.2(0)	8.1(9)	9(8)	8(10)	0.12
11	14(0)	12.6(0)	15.6(0)	11.1(9)	11(10)	11(10)	11(10)	11(10)	0.10
14	16.1(0)	14.2(0)	18.2(0)	14(10)	14(10)	14(10)	14(10)	14(10)	0.10
$N^{400}_{0.5,d}$3	5(0)	6.3(0)	6.4(0)	5(0)	5(0)	3.1(9)	3.1(9)	3(10)	0.48
8	8.8(3)	8.2(8)	9(0)	8.1(9)	8(10)	8(10)	8(10)	8(10)	0.11
11	11.9(3)	11(10)	11.9(4)	11(10)	11(10)	11(10)	11(10)	11(10)	0.06
$N^{800}_{0.1,d}$11	26.3(0)	30.5(0)	40.3(0)	23.7(0)	22.8(1)	11.3(7)	11.3(7)	11(10)	0.45
14	28.1(0)	31.4(0)	41.9(0)	23.5(0)	22.4(2)	14(10)	14(10)	14(10)	0.45
22	34.9(0)	33.1(0)	48.3(0)	27.3(0)	27.3(0)	22.5(5)	22(10)	22(10)	0.60
$N^{800}_{0.3,d}$3	8.7(0)	12.1(0)	12.5(0)	9(0)	8.8(1)	7.9(6)	7.3(6)	3(10)	88.3
5	9.7(0)	12.2(0)	13.3(0)	9.4(0)	9.4(0)	6.8(5)	5(10)	5(10)	1.89
$N^{800}_{0.5,d}$3	6(0)	7.3(0)	7(0)	6(0)	6(0)	4.4(5)	3.4(8)	3.2(9)	3.25
6	7.4(2)	7.8(0)	8.3(0)	6.3(7)	6.2(8)	6.5(8)	6(10)	6(10)	2.41

Table 5: *Numerical results for various approaches on the second benchmark.*

5.4. Runtime analysis and computational complexity

Our method's execution time, relative to instance size, follows a polynomial function $\mathcal{C}(n) = \mathcal{P}(n)$. By conducting polynomial regression on the first benchmark's runtime, we derived an empirical complexity function: $\mathcal{P}(n) = 0.01791n^2 - 0.4981n + 6.206$. This polynomial function adequately represents the execution time of the best solution found so far. Our approach demonstrates polynomial complexity for the considered instances. The algorithm's theoretical complexity mainly depends on the repair and crossover functions, both with $\mathcal{O}(V^2 + VE)$ complexity, ensuring efficiency and scalability for larger instances (see Figure 5).

6. Conclusion

In this study, we tackled the Minimum Dominating Set problem, a challenging NP-hard problem in graph theory, by proposing a Hybrid Cuckoo Search Algorithm (HCSA-MDS). Our algorithm outperformed state-of-the-art techniques in experimental evaluations on multiple benchmark sets, demonstrating superior performance in best, average, and worst solution comparisons. HCSA-MDS combines the efficient exploration of the cuckoo search algorithm using Lévy flights with intensification schemes and a genetic crossover operator for solution exploitation. As a future perspective, we plan to explore hybridizing with single-solution metaheuristics to further enhance the effectiveness and efficiency of our approach.

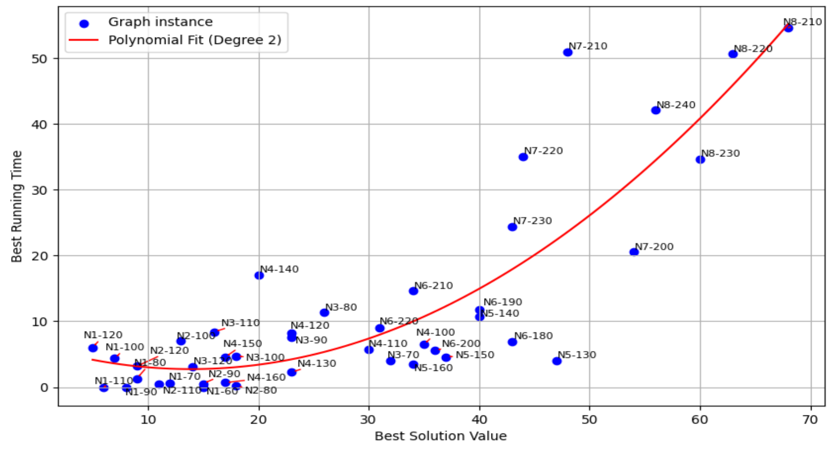


Figure 5: *HCSA-MDS execution time and polynomial fit.*

Acknowledgements

The authors would like to express their gratitude to the dedicated review team of two anonymous reviewers whose insightful comments greatly enhanced the quality of the paper.

References

[1] Abed, S. A. and Rais, H. M. (2017). Hybrid bat algorithm for minimum dominating set problem. Journal of Intelligent & Fuzzy Systems, 33(4), 2329-2339. IOS Press. doi: 10.3233/JIFS-17398

[2] Alkhalifah, Y. and Wainwright, R. L. (2004). A genetic algorithm applied to graph problems involving subsets of vertices. Proceedings of the 2004 Congress on Evolutionary Computation. 04TH8753(1), 303-308. IEEE. doi: 10.1109/CEC.2004.1330871

[3] Jeremy Blum, Min Ding, Andrew Thaeler, Xiuzhen C, Du DZ, and Pardalos P. (2004). Connected dominating set in sensor networks and manets. Handbook of Combinatorial, Springer, US. doi: 10.1007/0-387-23830-1_8

[4] Boumedine, N. and Bouroubi, S. (2022). Protein folding in 3D lattice HP model using a combining cuckoo search with the Hill-Climbing algorithms. Applied Soft Computing, 119, 108564. Elsevier. doi: 10.1016/j.asoc.2022.108564

[5] Brown, C. T., Liebovitch, L. S. and Glendon, R. (2007). Lévy flights in Dobe Ju/'hoansi foraging patterns. Human Ecology, 35, 129-138. Springer. doi: 10.1007/s10745-006-9083-4

[6] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7, 1-30. JMLR.org. doi: 10.5555/1248547.1248548

[7] Dinh, T. N., Shen, Y., Nguyen, D. T. and Thai, M. T. (2014). On the approximability of positive influence dominating set in social networks. Journal of Combinatorial Optimization, 27(3), 487-503. Springer. doi: 10.1007/s10878-012-9530-7

[8] Garey, M. R. and Johnson, D. S. (1979). Computers and intractability. Freeman, San Francisco. doi: 10.5555/574848

[9] Gandomi, A. H., Yang, X.-S. and Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with Computers, 29, 17-35. Springer. doi: 10.1007/s00366-011-0241-y

[10] Haynes, T. W., Hedetniemi, S. and Slater, P. (2013). Fundamentals of domination in graphs. CRC Press. doi: 10.1201/9781482246582

[11] Hedar, A.-R. and Ismail, R. (2010). Hybrid genetic algorithm for minimum dominating set problem. In: Computational Science and Its Applications–ICCSA 2010: International Conference, Fukuoka, Japan, March 23-26, 2010, Proceedings, Part IV, 457-467. Springer. doi: 10.1007/978-3-642-12189-0_40

[12] Hedar, A.-R. and Ismail, R. (2012). Simulated annealing with stochastic local search for minimum dominating set problem. International Journal of Machine Learning and Cybernetics, 3, 97-109. Springer. doi: 10.1007/s13042-011-0043-y

[13] Ho, C. K., Singh, Y. P. and Ewe, H. T. (2006). An enhanced ant colony optimization metaheuristic for the minimum dominating set problem. Applied Artificial Intelligence, 20(10), 881-903. Taylor & Francis. doi: 10.1080/08839510600940132

[14] Yoichi, I. (2011). A faster algorithm for dominating set analyzed by the potential method. IPEC, 2011. doi: 10.1007/978-3-642-28050-4_4

[15] Misra, R. and Mandal, C. (2009). Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks. IEEE Transactions on Parallel and Distributed Systems, 21(3), 292-302. IEEE. doi: 10.1109/TPDS.2009.78

[16] Parekh, A. K. (1991). Analysis of a greedy heuristic for finding small dominating sets in graphs. Information Processing Letters, 39(5), 237-240. Elsevier. doi: 10.1016/0020-0190(91)90021-9

[17] Payne, R. B. and Sorensen, M. D. (2005). The cuckoos. Bird Families of the World, 15. Bird Families of the World. doi: 10.1093/oso/9780198502135.001.0001

[18] Payne, R. and Kirwan, G. (2020) Chestnut-winged Cuckoo (Clamator coromandus). In Del Hoyo, J., Elliott, A., Sargatal, J., Christie, D. and De Juana, E. (Eds.) Birds Of The World, 1st Ed. doi: 10.2173/bow.chwcuc1.01

[19] Sanchis, L. A. (2002). Experimental analysis of heuristic algorithms for the dominating set problem. Algorithmica, 33, 3-18. Springer. doi: 10.1007/s00453-001-0101-z

[20] Pogorelov, K., Schroeder, D. T., Filkuková, P., Brenner, S. and Langguth, J. (2021). Wico text: a labeled dataset of conspiracy theory and 5g-corona misinformation tweets. In: Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks, 21-25. doi: 10.1145/3472720.3483617

[21] Sheskin, D. J. (2003). Handbook of parametric and nonparametric statistical procedures. Chapman and Hall/CRC. doi: 10.1201/9780429186196

[22] Takaguchi, T., Hasegawa, T. and Yoshida, Y. (2014). Suppressing epidemics on networks by exploiting observer nodes. Physical Review E, 90(1), 012807. APS. doi: 10.1103/PhysRevE.90.012807

[23] Umbarkar, A. J. and Sheth, P. D. (2015). Crossover operators in genetic algorithms: a review. ICTACT Journal on Soft Computing, 6(1). doi: 10.5120/ijca2017913370

[24] Walton, S., Hassan, O., Morgan, K. and Brown, M. R. (2011). Modified cuckoo search: a new gradient free optimisation algorithm. Chaos, Solitons & Fractals, 44(9), 710-718. Elsevier. doi: 10.1016/j.chaos.2011.06.004

[25] Wan, P.-J., Alzoubi, K. M. and Frieder, O. (2003). A simple heuristic for minimum connected dominating set in graphs. International Journal of Foundations of Computer Science, 14(02), 323-333. World Scientific. doi: 10.1142/S0129054103001753

[26] Wu, J. and Li, H. (2001). A dominating-set-based routing scheme in ad hoc wireless networks. Telecommunication Systems, 18, 13-36. Springer. doi: 10.1023/A:1016783217662

[27] Xu, Y.-Z. and Zhou, H.-J. (2016). Generalized minimum dominating set and application in automatic text summarization. Journal of Physics: Conference Series, 699(1), 012014. IOP Publishing. doi: 10.1088/1742-6596/699/1/012014

[28] Yang, X.-S. and Deb, S. (2009). Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 210-214. IEEE. doi: 10.1109/NABIC.2009.5393690

[29] Yang, X.-S. and Deb, S. (2014). Cuckoo search: recent advances and applications. Neural Computing and Applications, 24, 169-174. Springer. doi: 10.1007/s00521-013-1367-1

[30] Zhao, D., Xiao, G., Wang, Z., Wang, L. and Xu, L. (2020). Minimum dominating set of multiplex networks: Definition, application, and identification. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(12), 7823-7837. IEEE. doi: 10.1109/TSMC.2020.2987163