

Applying Genetic Algorithms to Validate a Conjecture in Graph Theory: The Minimum Dominating Set Problem

Jorge Cervantes-Ojeda, María C. Gómez-Fuentes, Julian A. Fresán-Figueroa

Universidad Autónoma Metropolitana,
Mexico

{jcervantes, mgomez, jfresan}@cua.uam.mx

Abstract. This paper presents a case study where the interdisciplinary approach between artificial intelligence, specifically genetic algorithms, and discrete mathematics has been instrumental in verifying a conjecture in graph theory. The focus of our research lies in the Minimum Dominating Set (MDS) problem, which involves identifying a dominating set with the minimum cardinality for a given graph. While previous works have primarily aimed at utilizing genetic algorithms to efficiently find satisfactory solutions to the MDS problem, our study aims to discover the optimal solution. The Rank GA algorithm employed in our work possesses the ability to escape local optima while simultaneously conducting local search to refine the best available solution. Since the MDS problem is known to be NP-hard, the characteristics of Rank GA, coupled with the identification of regularities in one of the solutions, enabled us to verify the conjecture for graphs comprising 30, 80, 312, and 800 vertices. This research serves as a testament to the versatility of genetic algorithms, showcasing their utility not only in practical problem-solving but also in tackling theoretical challenges.

Keywords: GA for theoretical problems, rank GA, genetic algorithms, minimum dominating set.

1 Introduction

Genetic Algorithms (GAs) have gained significant popularity for solving practical problems; however, their application in pure mathematics remains relatively limited. In this study, we present a compelling case where the Rank Genetic Algorithm [1] has facilitated the verification of a conjecture within the realm of discrete mathematics, particularly in the field of graph theory.

The Minimum Dominating Set (MDS) problem holds great significance in algorithmic graph

theory, finding applications in various domains and prompting extensive research endeavors [2]. A vast array of literature, encompassing articles, books, and surveys, has been dedicated to the MDS problem [3, 4]. Notably, the MDS problem and its variants find prominent applications in communication networks [5], and social networks [6, 7, 8]. For further exploration of the MDS problem's application areas, pertinent works can be referred to in [9].

The MDS problem is widely acknowledged to be NP-hard, prompting significant research efforts in developing heuristics to attain high-quality solutions within a reasonable timeframe. Noteworthy works in this domain can be found in [10, 11]. Numerous studies have been undertaken to devise efficient algorithms for solving the MDS problem, with the aim of improving upon the trivial bound of $O\left(\frac{2^n}{\sqrt{n}}\right)$ [12].

Consequently, most works employing heuristic methods for MDS focus on finding good solutions as quickly as possible. In contrast, our objective is to uncover the best possible solution through the utilization of a genetic algorithm. Specifically, we seek real solutions that aid in verifying the validity of a theoretical conjecture.

To accomplish this, we employed the Rank GA due to its reliability in escaping local optima while conducting a local search to refine the best solution obtained thus far. Through the utilization of Rank GA, we generated empirical data that aids in determining an upper bound for the cardinality of the minimal dominating set in incidence graphs of classical generalized quadrangles.

The paper follows the following structure: In Section 2, we provide an overview of previous

works where Genetic Algorithms (GAs) have been successfully employed to solve mathematical problems. Section 3 presents a detailed explanation of the conjecture that is to be verified in our study. In Section 4, we describe the Rank GA and elaborate on its utilization in determining the minimal dominating set of a graph. The experimental results are presented in Section 5, while the conclusions drawn from our research are outlined in Section 6.

2 Background

Genetic Algorithms (GAs) are widely recognized for their utility in engineering problems, but it is important to highlight their applicability in advancing mathematical research as well [13].

For instance, Jong and Spears [14] demonstrated that GAs can be effectively employed to solve NP-Complete problems. Additionally, Jakobs [15] applied a GA to tackle a geometry problem, while Pourrajabian et al. [16] successfully utilized a GA to solve nonlinear algebraic equations. In more recent studies, Cervantes et al. [17] utilized a Rank GA to enhance an upper bound that governs the rainbow k -connectivity of $(k; 6)$ -cages.

In traditional usage, Genetic Algorithms (GAs) are typically employed to optimize a given fitness function by exploring a solution space and aiming to find the optimal solution. However, in the study by García-Altamirano et al. [18], the Rank GA underwent a unique adaptation: it was designed to exclusively perform Hajós operators with the objective of finding a path to reach a predetermined resulting directed graph. This represents a notable departure from the conventional usage of GAs.

The algorithm started from an initial state where each individual represented a starting digraph, while the goal digraph that the algorithm aimed to reach was known in advance. The intriguing aspect was to uncover the method by which the Rank GA traversed the solution space to achieve the final state. By closely analyzing the algorithm's steps in the specific case, a general approach applicable to all cases was discovered.

Indeed, showcasing the potential contributions of Genetic Algorithms (GAs) in the realm of

mathematics is not only valuable but also inspiring. Demonstrating how GAs can be effectively employed in mathematical problems can serve as a catalyst for progress in other areas where scientific advancements are primarily theoretical. By highlighting the application of GAs in mathematics, researchers can explore new avenues and innovative approaches to solve complex theoretical problems.

This interdisciplinary approach can lead to novel insights, methodologies, and solutions, bridging the gap between theoretical and practical domains. Thus, presenting such works can provide motivation and inspiration for researchers in various fields to explore the potential of GAs beyond their traditional applications.

3 The Conjecture to be Verified

In this section, we introduce key concepts that provide the necessary context for the conjecture that was verified using the Rank GA.

3.1 Generalized Quadrangle

Let k and r be positive integers greater than 2. A Generalized Quadrangle $CG(k-1, r-1)$ refers to a system comprising lines and points, where an incidence relation is established based on the following axioms:

1. Two lines intersect at most in one point.
2. At most one line passes through any two points.
3. Exactly r lines pass through each point.
4. Each line contains exactly k points.
5. If a point p is not contained in a line l , there exists precisely one line that passes through p and intersects l .

These axioms define the fundamental properties of a Generalized Quadrangle, providing the framework for studying and analyzing their characteristics.

Fig. 1 illustrates the classical Generalized Quadrangle $G(q, q)$ for the specific case of $q = 2$. In this case, the values of r and k are both equal to 3.

Considering a Generalized Quadrangle $G(q, q)$ with q as a prime number, the incidence relation between points and lines can be depicted

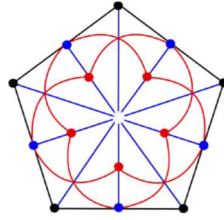


Fig. 1. Classical generalized quadrangle $CG(q, q)$ $q = 2$

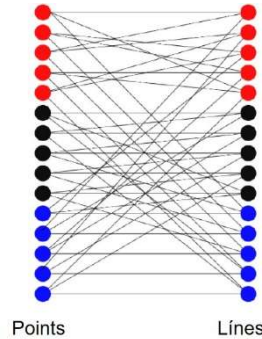


Fig. 2. Incidence graph of the classical generalized quadrangle $CG(q, q)$ $q = 2$

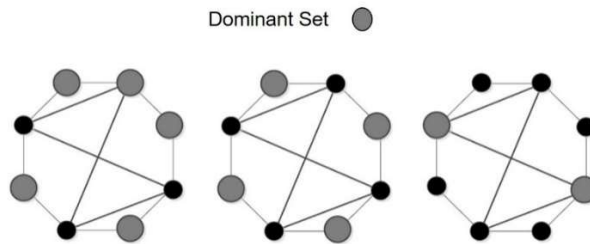


Fig. 3. Examples of Dominant Sets in a Graph

as a bipartite graph, where the points and lines correspond to vertices of the graph and a point is adjacent to a line if they are incident. The number of vertices in each part of the bipartite graph can be determined using expression (1):

$$q^3 + q^2 + q + 1. \tag{1}$$

Fig. 2 shows the incidence bipartite graph for the classical Generalized Quadrangle $CG(q, q)$ $q = 2$.

3.2 The Minimum Dominating Set Problem

The Minimum Dominating Set (MDS) problem revolves around identifying a dominant set in a given graph G . A graph G is defined as a set of

vertices V and a set of edges E , denoted as $G = (V, E)$. In the context of the MDS problem, a dominant set is a subset D of V , where every vertex that does not belong to D is connected to (at least) one vertex in D through an edge in E . To illustrate this concept, Fig. 3 showcases examples of various dominant sets for a given graph.

Dominating Set (Mathematical Definition): Given a simple undirected graph $G = (V, E)$, where V represents the set of vertices and E represents the set of edges, a subset D of vertices $D \subseteq V$ is a dominating set of G if, for every vertex u in $V \setminus D$, there exists a vertex x in D that is adjacent to u . In other words, every vertex not in D is adjacent to at least one vertex in D .

Vertex Domination: A vertex u is said to dominate a vertex v if either $u = v$ or u is adjacent to v . This concept signifies the relationship between vertices within the context of domination in the graph.

Minimal Dominating Set (Mathematical Definition): The minimal dominating set of a graph G is a dominating set D such that removing any element from D would render it non-dominant. In other words, for every element x in D , the set $D - \{x\}$ is not a dominating set.

Domination Number: The domination number $\gamma(G)$, of a graph G is defined as the cardinality (size) of the smallest dominating set in G . It represents the minimum number of vertices required for a set in G to be a dominating set.

3.3 The Conjecture

The conjecture proposes that for a prime number q , the incidence bipartite graph associated with the classical generalized quadrangle $CG(q, q)$ possesses an upper bound on the domination number $\gamma(G)$. The domination number, denoted as $\gamma(G)$, represents the minimum number of vertices required to form a dominating set in the graph G .

According to the conjecture, the upper bound for the domination number $\gamma(G)$ in the incidence graph associated with $CG(q, q)$ can be determined using Equation (2):

$$\gamma(G) = 2q^2 + 1. \quad (2)$$

Equation (2) provides an expression that bounds the value of the domination number in the incidence bipartite graph associated with the classical generalized quadrangle $CG(q, q)$, where q is a prime number. The conjecture suggests that the domination number does not exceed this upper bound. Further investigation and verification are required to confirm the validity of this conjecture.

4 Rank Genetic Algorithm for Finding Minimal Dominating Set

In the field of evolutionary computation, there is a fundamental trade-off between exploration and exploitation. If exploitation dominates, the population converges prematurely, restricting the search to a small subspace of the search space.

This can lead to the algorithm getting stuck in a local optimum, resulting in suboptimal solutions. On the other hand, excessive exploration leads to searching in a large region, making it difficult for the algorithm to converge and obtain a solution.

The Rank Genetic Algorithm (Rank GA) [1] is a heuristic approach designed to strike a balance between exploration and exploitation. It achieves this by applying genetic operators that allow for both local and global search simultaneously. In the Rank GA, individuals within the population are evaluated and ranked from best to worst based on their fitness just before applying selection, recombination, and mutation operators to each individual.

The operators are applied to individuals based on their current rank. The highly ranked individuals tend to remain unchanged and primarily recombine with better-performing individuals, promoting the exploitation of favorable genetic traits.

On the other hand, lower-ranked individuals tend to recombine with worse-performing individuals, facilitating the exploration of distant genotypes in the search space, thus enabling the escape from local optima in the fitness landscape.

The probability of mutation assigned to each individual follows a monotonously increasing pattern according to their rank in the population, which is determined by their fitness. This means that individuals with higher ranks have a lower probability of mutation, while individuals with lower ranks have a higher probability.

In [19], it has been demonstrated that the Rank GA outperforms a simple GA in challenging fitness landscapes, especially those with binary string genotypes, where a good balance between exploration and exploitation is crucial. This balance is particularly necessary when the fitness landscape contains numerous local optima, and a modular solution can be constructed through evolution.

In such cases, finding each module's solution may be challenging, requiring high exploration levels. Simultaneously, the integration of already found modules' solutions necessitates high exploitation levels within the algorithm.

By effectively balancing exploration and exploitation, the Rank GA offers superior performance in difficult fitness landscapes, providing a robust approach for problems that

require a good trade-off between exploration and exploitation in evolutionary computation.

4.1 Solution Representation

The Rank GA adopts a binary representation for solutions in this case. Each solution X in the population is represented as an array of Booleans, where the size of the array corresponds to the number of vertices in the graph. The order of the array follows a fixed arrangement that aligns with the vertices of the graph.

In this representation, if the i -th vertex of the graph is included in the solution X , then the corresponding element x_i in the array is set to 1. Conversely, if the i -th vertex is not part of the solution, x_i is assigned a value of 0.

4.2 Fitness Function

To evaluate a solution X , we employ two metrics. Firstly, we use the metric k to evaluate the number of vertices in $V(G)$ that are dominated by X . In this context, a larger value of k is desirable. Secondly, we consider the cardinality c of X , which represents the number of elements in X . Here, a smaller value of c is preferred.

The fitness function to be maximized is defined by equation (3) as follows:

$$fitness(X) = k + c. \quad (3)$$

The fitness function combines the metrics of the number of dominated vertices (k) and the cardinality of the solution (c) into a single value. By maximizing the fitness function, we want to achieve solutions that dominate a larger number of vertices while minimizing the cardinality of the solution. This allows for the identification of solutions that strike a balance between domination and the size of the minimal dominating set.

Note that with this fitness function, there is an element of neutrality that can be advantageous for the search process. For example, consider the scenario where the best individual has a fitness value of 50, with $k = 80$ and $c = 30$. In this case, another solution with $k = 81$ and $c = 31$ would be just as favorable for the algorithm.

This implies that either solution could occupy the top position in the rank of solutions. At the top, they would be protected from being lost during the

application of genetic operators, as per the Rank GA scheme, and the search would be “located” around that top solution.

This neutrality aspect in the fitness function allows for a broader exploration of the search space. Rather than being strictly limited to search from a single current best solution, the search can shift its “center” between different neutral solutions, thus expanding the exploration to encompass nearby solutions to any of them.

4.3 The Rank GA Operators

The Rank GA follows a specific procedure, beginning with the random initialization of the population. Individuals are then evaluated and sorted from best to worst based on their fitness. The rank of the i -th individual in the sorted population, denoted as r_i , is determined using equation (4):

$$r_i = \frac{i}{N-1}, \quad (4)$$

where i ranges from 0 to $N-1$, and N represents the number of individuals in the population.

The Rank GA then applies the following operations iteratively until a stopping criterion is met:

1. Rank Selection
2. Sort and rank
3. Rank Recombination
4. Evaluation, sort and rank
5. Rank Mutation
6. Evaluation, sort and rank

Rank Selection: Rank selection involves cloning individuals according to a two-step procedure.

First, the desired number of clones, $cloneNbr_i$, is calculated for each individual i using equation (5):

$$cloneNbr_i = S * (1 - r_i)^{S-1}. \quad (5)$$

Here, r_i represents the rank of individual i , ranging from 0 to 1, and S corresponds to the selective pressure of the Rank Selection operator. The floor of $cloneNbr_i$ is taken, resulting in the number of clones to generate for each individual i .

Secondly, additional clones are produced as follows. The procedure initializes $i=0$. While the total number of clones is less than the original

number of individuals N , the fractional part of $cloneNbr_i$ is used as the probability of producing an extra clone of individual i . If a random number in the range $(0,1)$ is lower than that probability, a clone of individual i is generated. The value of i is incremented modulo N to ensure it cycles through the individuals. The main idea in this step is to increase the likelihood of producing extra clones for individuals who lost a larger fractional part when the floor of $cloneNbr_i$ was taken in the first step.

The selective pressure (S) determines the intensity with which the selection operator concentrates the population towards a single dominant genotype. Lower selective pressure allows for more genetic diversity to be preserved in the population after each application of the selection operator.

By implementing Rank Selection, the Rank GA can effectively guide the cloning process, favoring individuals with higher ranks and promoting the retention and reproduction of genetic traits associated with higher fitness while maintaining some genetic diversity in the population.

The adjustable selective pressure provides control over the concentration of individuals, influencing the balance between exploitation and exploration during the evolutionary process.

Rank Recombination: The Rank Recombination operator forces mating between individuals with indexes i and $i+1$ in the sorted population, where i increases by 2. This strategy ensures that individuals mate with others that are nearby in terms of rank. The purpose is to prevent individuals from mating with those that are significantly different in rank.

By doing so, the best individuals are safeguarded from being recombined with poorly performing ones, as such combinations could disrupt the advantageous gene value combinations present in the best individuals. In contrast, recombination between two high-performing individuals can result in the fusion of their respective already found building blocks of the optimal solution, potentially leading to the creation of improved offspring.

Rank Recombination is done as uniform crossover with parent substitution. This means that all genes have an equal probability of being switched between parents, and the offspring

always replaces the parents. The probability of switching genes, referred to as the switch probability (p_{switc}), is typically set to 0.5 as the recommended value.

In the selection operator, a selective pressure value of $S = 3$ is commonly used. With $S = 3$, three clones of the fittest individual are generated. This approach ensures that the best-ranked clone has the opportunity to recombine with a clone of itself (the second-best ranked), thereby preserving the advantageous genetic traits present in the best individual.

The third-ranked clone would recombine with the second-best individual (ranked fourth), which is expected to possess different chromosomes, resulting in the production of diverse offspring. This way, the Rank GA not only preserves the best individual but also allows for recombination with another high-performing individual.

By employing Rank Recombination with a selective pressure value of $S = 3$, the Rank GA strikes a balance between preserving the advantageous genes of the best individual and introducing local search through recombination with other good individuals. This approach enhances the algorithm's ability to explore the search space effectively while retaining and propagating beneficial genetic traits.

Rank Mutation: The probability of mutation assigned to each individual, denoted as p_i is determined by a monotonic increasing function of their rank. The function is defined as follows:

$$p_i = p_{max} * r_i^{\left(\frac{\ln(p_{max}*G)}{\ln(N-1)}\right)}. \quad (6)$$

In this equation, p_{max} represents the maximum probability of mutation that can be assigned, r_i is the rank of individual i , G is the size of the genotype, and N is the population size.

The function assigns a mutation probability of 0 to the best individual, $1/G$ to the second-best individual, and p_{max} to the worst individual. As Rank Mutation is applied after Rank Recombination, it is highly likely that the first two individuals in the ranked list are identical.

This means that the best individual remains unchanged (by mutation), while the second-best individual is likely to undergo exactly one mutation. This mechanism promotes exploration in the

immediate vicinity of that second individual which is likely to be a clone of the best.

Conversely, poor-performing individuals undergo significant mutations to facilitate their escape from the basins of attraction of local optima. The larger mutations applied to these individuals ensure they can explore alternative regions in the search space and break free from suboptimal solutions.

By employing Rank Mutation with a probability assigned based on the rank, the Rank GA combines the preservation of advantageous traits in the best individual with exploratory mutations in the neighborhood of the second individual. Simultaneously, it encourages significant mutations in poorly performing individuals to facilitate their exploration of alternative regions in the search space.

This approach enhances the algorithm's ability to strike a balance between exploitation and exploration, promoting the search for optimal solutions in complex fitness landscapes.

5 Results

To verify the conjecture presented in equation (2), we conducted experiments on four bipartite graphs of generalized quadrangles with 30, 80, 312, and 800 vertices. These cases come from the Incidence graph of the classical generalized quadrangle $CG(q, q)$ with $q = 2, 3, 5$ and 7 respectively. Using the Rank GA algorithm, we searched for a minimal dominating set in each of these incidence bipartite graphs of Classical Generalized Quadrangles.

By determining the cardinality of the minimal dominating set, we were able to verify the conjecture stated in equation (2).

For the graphs with 30 and 80 vertices, the Rank GA algorithm quickly obtained solutions with $\gamma(G)$ equal to the conjectured result. However, for the graph with 312 vertices, although the solutions found were very close to the conjectured result, they were not exactly equal to it.

Upon analyzing the solutions for 30 and 80 vertices, we observed a particular pattern illustrated in Fig. 4. Specifically, on the left side of the bipartite graph, only the last bits were set to 1, while on the right side, only the first few bits were

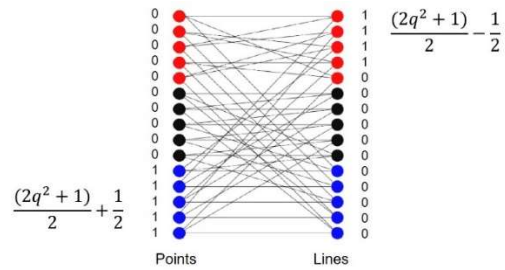


Fig. 4. Regularity in one of the solutions

Table 1. Comparison of $\gamma(G)$ predicted by the conjecture and $\gamma(G)$ obtained through Rank GA

q	# Vertices	$\gamma(G)$ Conjecture	$\gamma(G)$ Rank GA
2	30	9	9
3	80	19	19
5	312	51	51
7	800	99	99

set to 1. This led us to consider the possibility of similar solutions existing in the case of 312 vertices. We tested a solution where the number of bits set to 1 on the left side was determined by equation (7):

$$\frac{(2q^2+1)}{2} + \frac{1}{2} \tag{7}$$

And the number of bits set to 1 on the right side was given by equation (8):

$$\frac{(2q^2 + 1)}{2} - \frac{1}{2} \tag{8}$$

This solution turned out to be a valid dominating set when applied to the graph with 312 vertices. Encouraged by these findings, we further tested the case of 800 vertices, which also yielded a dominating set. These results strongly support the conjecture presented in equation (2).

As shown in Table 1, the domination number $\gamma(G)$ predicted by the conjecture was found to be equal to the $\gamma(G)$ obtained through the Rank GA algorithm.

6 Conclusions

In this research article, we explored the application of the Rank Genetic Algorithm (Rank GA) in

verifying a conjecture in the field of graph theory. Specifically, we focused on the Minimum Dominating Set (MDS) problem and its relationship with incidence graphs of classical Generalized Quadrangles.

Through our investigation, we demonstrated that the Rank GA algorithm is capable of effectively searching for minimal dominating sets in some bipartite graphs. By evaluating the domination number $\gamma(G)$ of the obtained solutions, we were able to verify the validity of the conjecture proposed for the incidence graphs of generalized quadrangles.

Moreover, we were able to construct specific sets attaining the conjectured bound. This is an important result because many conjectures in mathematics do not have explicit constructions.

Our results showed that the Rank GA algorithm performed very well in finding solutions that aligned with the conjectured domination numbers. For graphs with 30 and 80 vertices, the algorithm quickly identified solutions that matched the expected values.

Although the solutions for the graph with 312 vertices were very close to the conjectured result, they were not exact. However, further analysis revealed that the pattern observed in the smaller graphs also held true for the larger graphs, supporting the conjecture.

The use of the Rank GA algorithm allowed us to strike a balance between exploration and exploitation, ensuring efficient search capabilities in the fitness landscapes of the MDS problem. The rank-based selection, recombination, and mutation operators employed in the Rank GA algorithm provided a robust framework for navigating the search space and escaping local optima.

The experimental results presented in this study, along with the consistent agreement between the conjectured domination numbers and those obtained through the Rank GA algorithm, validate the effectiveness of using genetic algorithms for theoretical problems in graph theory.

The Rank GA algorithm demonstrated its ability to contribute not only to practical problem-solving but also to the exploration and verification of a mathematical conjecture and the construction of objects satisfying the conjecture.

Overall, this research highlights the potential of genetic algorithms, specifically the Rank GA, in the

field of graph theory. The successful verification of the conjecture in the context of the Minimum Dominating Set problem for incidence graphs of classical Generalized Quadrangles emphasizes the potential of genetic algorithms to advance theoretical understanding in mathematics.

Future studies could further explore the capabilities of genetic algorithms in other areas of theoretical research and extend their application to different problem domains.

By bridging the gap between artificial intelligence and discrete mathematics, this research opens up new possibilities for leveraging computational methods in solving complex theoretical problems and advancing scientific knowledge.

References

1. **Cervantes, J., Stephens, C. R. (2008).** Rank based variation operators for genetic algorithms. Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp. 905–912. DOI: 10.1145/1389095.1389271.
2. **Hedar, A. R., Ismail, R. (2010).** Hybrid genetic algorithm for minimum dominating set problem. In: **Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.)**, Computational Science and Its Applications – ICCSA 2010, Lecture Notes in Computer Science, vol 6019, pp. 457–467. DOI: 10.1007/978-3-642-12189-0_40.
3. **Couturier, J. F., Heggernes, P., Van't-Hof, P., Kratsch, D. (2013).** Minimal dominating sets in graph classes: combinatorial bounds and enumeration. Theoretical Computer Science, Vol. 487, pp. 82–94. DOI: 10.1016/j.tcs.2013.03.026.
4. **Liu, Z., Wang, B., Guo, L. (2010).** A survey on connected dominating set construction algorithm for wireless sensor networks. Information Technology Journal, Vol. 9, No. 6, pp. 1081–1092.
5. **Bai, X., Zhao, D., Bai, S., Wang, Q., Li, W., Mu, D. (2020).** Minimum connected dominating sets in heterogeneous 3D wireless ad hoc

- networks. *Ad Hoc Networks*, Vol. 97 DOI: 10.1016/j.adhoc.2019.102023.
6. **Bouamama, S., Blum, C. (2021)**. An improved greedy heuristic for the minimum positive influence dominating set problem in social networks. *Algorithms*, Vol. 14, No. 3, pp. 79. DOI: 10.3390/a14030079.
 7. **Chalupa, D. (2018)**. An order-based algorithm for minimum dominating set with application in graph mining. *Information Sciences*, Vol. 426, pp. 101–116. DOI: 10.1016/j.ins.2017.10.033.
 8. **Dinh, T. N., Shen, Y., Nguyen, D. T., Thai, M. T. (2014)**. On the approximability of positive influence dominating set in social networks. *Journal of Combinatorial Optimization*, Vol. 27, No. 3, pp. 487–503. DOI: 10.1007/s10878-012-9530-7.
 9. **Levin, M. S. (2021)**. Note on dominating set problems. *Journal of Communications Technology and Electronics*, Vol. 66, Suppl. 1, pp. S8–S22. DOI: 10.1134/S1064226921130040.
 10. **Li, B., Zhang, X., Cai, S., Lin, J., Wang, Y., Blum, C. (2021)**. NuCDS: An efficient local search algorithm for minimum connected dominating set. *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 1503–1510.
 11. **Pinacho-Davidson, P., Blum, C. (2020)**. Barrakuda: A hybrid evolutionary algorithm for minimum capacitated dominating set problem. *Mathematics*, Vol. 8, No. 11, pp. 1858. DOI: 10.3390/math8111858.
 12. **Fomin, F. V., Grandoni, F., Pyatkin, A. V., Stepanov, A. A. (2008)**. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Transactions on Algorithms (TALG)*, Vol. 5, No. 1, pp. 1–17. DOI: 10.1145/1435375.1435384.
 13. **Mitchell, M. (1996)**. *Introduction to genetic algorithms*. MIT Press.
 14. **De Jong, K. A., Spears, W. M. (1989)**. Using genetic algorithms to solve NP-complete problems. *ICGA*, pp. 124–132.
 15. **Jakobs, S. (1996)**. On genetic algorithms for the packing of polygons. *European journal of operational research*, Vol. 88, No. 1, pp. 165–181.
 16. **Pourrajabian, A., Ebrahimi, R., Mirzaei, M., Shams, M. (2013)**. Applying genetic algorithms for solving nonlinear algebraic equations. *Applied Mathematics and Computation*, Vol. 219, No. 24, pp.11483–11494. DOI: 10.1016/j.amc.2013.05.057.
 17. **Cervantes-Ojeda, J., Gómez-Fuentes, M., González-Moreno, D., Olsen, M. (2019)**. Rainbow connectivity using a rank genetic algorithm: Moore cages with girth six. *Journal of Applied Mathematics*, Vol. 2019. DOI: 10.1155/2019/4073905.
 18. **García-Altamirano, J. C., Olsen, M., Cervantes-Ojeda, J. (2023)**. How to construct the symmetric cycle of length 5 using Hajós construction with an adapted rank genetic algorithm. *Discrete Mathematics & Theoretical Computer Science*, Vol. 25, No. 1. DOI: 10.46298/dmtcs.10189.
 19. **Cervantes, J., Stephens, C. R. (2008)**. Limitations of existing mutation rate heuristics and how a rank GA overcomes them. *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, pp. 369–397. DOI: 10.1109/TEVC.2008.927707.

Article received on 15/06/2023; accepted on 20/09/2023.

Corresponding author is Jorge Cervantes-Ojeda.