Journal of Innovations in Computer Science and Engineering
Shahid Beheshti Unversity

*May 2023, Volume 1, Number 1*

# Efficient Heuristic Algorithms for unweighted minimum vertex cover problem

**Vida Barzegaran1, Zeinab Torabi1** Code Orcid: 0000-0002-2526-688X**, Sahar Kianian1** Code Orcid: 0000-0001-9367-7456**,**
computer arithmetic and optimal operation of integrated energy systems
1Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.
Assistant professor of computer engineering at Shahid Rajaee University
vidi@aut.ac.ir, z.torabi@sru.ac.ir, sahar.kianian@sru.ac.ir

**Abstract**
Covering all edges in a graph with a small set of vertices is one of the most fundamental graph problems which is called the minimum vertex cover problem. In the literature different strategies have been employed to find near-optimal minimum vertex cover set in different kinds of graphs.
In this work, two efficient algorithms (i.e., MAxA and MAxAR) are introduced to find the minimum vertex cover set in any unweighted undirected graph. The proposed construction algorithms have two main steps in each iteration which explore neighborhoods of minimum degree vertices to find and select appropriate vertices for the cover set. Until all of the edges are removed or selected in the algorithms, these two steps are performed iteratively. The proposed algorithms have been implemented on DIMACS, BHOSLIB, and other benchmarks where experimental results show that the proposed algorithms outperform other relevant methods in terms of time and cardinality of vertex cover set.
Keywords: NP problem, Minimum vertex cover, Heuristic algorithm.



## 1. Introduction

There are many reasons for using surveillance cameras on city roads. In a road graph, the edges and nodes represent the roads and intersections, respectively. What we are required to do is to put cameras in the intersections to see the city in its entirety. In addition, as few cameras as possible should be used for cost efficiency. For a simple graph $G$, $C$ is a subset of vertices such that at least one ends of each edge is in $C$. Therefore, the goal is to find minimum vertex cover in the road graph. Below, some of real-world applications for minimum vertex cover are listed [1]:

In a wireless sensor network, the minimum vertex cover can be used for monitoring links, routing, data aggregation, clustering [2, 3], and so on. For example, in reference [4], the authors have proposed algorithms, which are used to increase the lifespan of a wireless sensor network. By selecting a minimum number of nodes, the mentioned algorithms reduce energy consumption.

In biochemistry calculations [5], the problem of minimum vertex cover has been used for the structure of the phylogenetic tree as well as for analyzing multiple sequence levels to infer the evolutionary relationships between genes and proteins. In the conflict graph, in which the vertices and edges represent the sequence of samples and the conflict between the corresponding sequences respectively, the minimum vertex cover is used to eliminate all of the paradoxes by removing the smallest sequence.

To protect large networks against propagation of hidden worms in real-time, the minimum vertex cover has recently been used in a graph in which the vertices and edges are the routing servers and connections between routing servers, respectively [6].

In other computer fields such as scheduling, VLSI design, and signal transmission, the minimum vertex cover is used to find the closest solution to the perfect solution [7]. An efficient algorithm for minimum vertex cover can improve parameters of time, cost, and resource allocation in different applications.

The potential applications of minimum vertex cover reveal its significance. In addition, due to the NP nature of the problem, it still draws the attention of researchers. Several exact algorithms [8-11] such as Branch and Bound, LP-Based Branch and Cut as well as approximate algorithms have been proposed to solve the problem. On the other hand, to find the optimal solution for the minimum vertex cover, any factor smaller than 1.3606 [12] is considered an NP-hard problem. Many algorithms have been proposed for the minimum vertex cover problem, including the maximum degree greedy algorithm (MDG) [13], vertex support algorithm (VSA) [14], modified vertex support algorithm (MVSA) [15], and the maximum degree adjacent to the minimum degree algorithm (MAMA) [16]. The construction algorithm for minimum vertex cover problem is based on finding a solution through expanding a partial vertex set. Primarily, in these algorithms, the vertex set is empty. Then, until it becomes a cover set, the vertices are repeatedly added to the set. MDG and VSA are greedy algorithms. Each one of them selects the maximum vertex degree and maximum vertex support in each period and then updates the graph. MVSA is a modified version of VSA. MAMA selects the vertex with the maximum degree in the neighborhood of the vertex with the minimum degree in each round.

The proposed algorithms similar to MAMA search the neighborhood of minimum degree vertices. However, in the proposed algorithms for each minimum degree vertices, one of the neighbors is selected in each period. The effectiveness of our proposed algorithms has been observed on the benchmark of small and large graphs. The simulation results have also shown that our proposed algorithms find the vertex cover set with a smaller number of vertices. In addition, it takes less time compared to MAMA.

The rest of the paper is organized as follows. Section 2 briefly addresses a selected number of more efficient MVC algorithms. In Section 3, the proposed algorithms have been introduced while experimental evaluations are described in Section 4. Finally, we draw our conclusions in Section 5.

## 2. Review of Literature

Because finding the best solution to the problem is impossible. This problem can be studied to find better solutions in the future. Several new methods, such as quantum computers, are now used to find the appropriate solution to the MVC problem [42]. But there are two challenges to mention:
1. As technology advances, the amount of data also increases. MVC is one of the few problems where kernelization (reducing input size by iterative data reduction rule) is known to be really very efficient. As a result, data management and big data issues can become important in the field. 2. The above MVC solutions use the local search method. Therefore, it is necessary to find different solutions to avoid getting stuck in local optimization. These solutions should also be pretty fast. In other words, it will find several suboptimal solutions in a short time and choose the best one among them. Every day, the application of the MVC problem manifests itself more in all types of networks, such as recently the problem was used to monitor surveillance cameras in the transport network in Russia [43]. In addition, in a paper made to predict and diagnose COVID-19; Graph theory and MVC were used to provide a suitable solution [44]. In the past six decades, a number of methods have been proposed to solve the minimum vertex cover. Most of these methods are approximate solutions. In reference [17], Anton investigated the difficulty of an algorithm on a graph that is e-dense everywhere; he showed that if we want to solve MVC problem with a factor less than $\frac{7+\varepsilon}{6+2\varepsilon}$, it turns into an NP-hard problem. In the present section, we review the algorithms for the minimum vertex cover with polynomial time complexity. Evolutionary [18 –22] and local search [23-35] algorithms are the two most common approaches for solving minimum vertex cover problem.

The most popular heuristic method for NP-hard combinatorial optimization problems is local search where, numerous important algorithms of this type are applicable to complicated problems such as satisfiability, coloring, and routing problems. Cai et al. proposed NuMVC, FastVC, NuMVC2+p, and FastVC2+p which have desirable performance in massive graphs. NuMVC includes two-stage exchange and uses "edge weighting with forgetting" to deal with and fix the problems of local search methods. In FastVC, using two heuristic functions leads to less time complexity and better performance. Improving the previous algorithms on the one hand and adding a pre-processing step to them, on the other hand, lead to NuMVC2+p and Fast VC2+p algorithms in finding the minimum vertex cover for large sparse graphs.

Several heuristic algorithms including 'Evolutionary Algorithm', 'Simulated Annealing', and 'Branch and Bound' algorithms are used to solve the problem of minimum vertex cover in a random graph that is modeled with $G(n, c/n)$. The efficiency of these algorithms has been investigated for different values of $c$ [36, 37].

Time complexity of MDG is $O(E^2)$. In the VSA, there is an adjacency matrix that is used to calculate the sum of degrees for all vertices. This value is called vertex support. Afterwards, a vertex with the maximum support is added to the cover set in each period. This takes $O(V^2)$.

MVSA calculates the vertex support in the same way as VSA. Then, in each period, it finds the vertices with minimum support and then adds the minimum neighbor to the cover set. The time complexity of MVSA is $O(EV^2 \log V^2)$. MAMA adds the largest neighbor of the minimum degree vertices in each period to the cover set. The time complexity of MAMA equals to $O(EV^2)$.

## 3. Proposed Algorithm

In this section, two algorithms are proposed to solve the minimum vertex cover problem, both of which are based on the neighbors of minimum degree vertices. The goal is to select nodes that can cover the largest number of uncovered edges. Considering MDG and MAMA algorithms in exploring the minimum degree neighbors, MaxA and MaxAR algorithms are introduced here where the time complexity of both algorithms is $O(EV^2)$. Primarily, the algorithms start with an empty cover set, and two steps are repeated to cover all edges of the graph. The first step in both algorithms is to find the vertices with minimum degree (i.e., $Min$ set). In the second step, the neighbors of the $Min$ set are searched to find the best candidates.

In MaxA, for each vertex in the $Min$ set, its largest neighbor (the neighbor with the largest degree) is selected and added to the cover set.

MaxAR, which is an extension of MaxA, selects the best candidate for each vertex in the $Min$ set between a maximum degree vertex and a random vertex among neighbors. The probability of choosing between each of these two choices is equal to 0.5. In fact, MaxAR is a combination of MaxA algorithm and random walk (RW) algorithm. RW is a preferable technique for heuristic approaches because it is simple to use and it can easily escape complex boundaries and local optima and cause extensive exploration in the state space. As mentioned earlier, MVC is an NP-hard problem and it needs effective heuristics to achieve admissible performance. The evaluation of the proposed algorithm shows that this goal is achieved with the help of the RW algorithm. Based on these observations, we have presented our two algorithms below, note that here $Z$ refers to node(s) with no edges:

**Algorithm1:**

Algorithm MaxA: Input $G(V, E)$, Output : Cover

1. Cover = {}
2. While $G$ is not empty
3. $Z$ = node(s) with degree 0
4. Remove $Z$ from $G(V, E)$
5. $Min$ = node(s) with minimum degree
6. $Max_{adjacent}$ = find a maximum degree adjacent to each node in $Min$
7. Cover = Cover ∪ $Max_{adjacent}$
8. Delete Min and its vertices
9. End while

**Algorithm 2:**

Algorithm MaxAR: Input: $G(V,E)$, Output : Cover

1. Cover = {}
2. While $G$ is not empty
3. $Z$= node(s) with degree 0
4. Remove $Z$ from $G(V,E)$
5. $Min$ = node(s) with minimum degree
6. $Max_{adjacent}$= find a maximum degree adjacent to each node in $Min$
7. $Random$ = Select a random adjacent to each node in $Min$
8. $Selected$= for each node in $Min$ select one vertex between its $Max_{adjacent}$ and $Random$ with probability ½.
9. Cover = Cover ∪ $Selected$
10. Delete Selected
11. End while

In Table 1, we have checked the time complexity of MAMA and MDG algorithms and our two proposed algorithms. As it is evident, the time complexity of our algorithms and MAMA are equal.

Table1. Time complexity of simulated algorithms

| MDG | MAMA | MaxA | MaxAR |
|---|---|---|---|
| $O(E^2)$ | $O(EV^2)$ | $O(EV^2)$ | $O(EV^2)$ |

## 4. Experimental Results

A series of experiments were conducted to evaluate the performance of our proposed algorithms and compared their results with MAMA and MDG. The experimental results in [16] showed that MAMA has better performance in comparison to VSA, MDG, and MVSA in the benchmarks of small and large graphs. Therefore, if we show the superiority of the proposed algorithms over MAMA, their superiority over MDG, VSA, and MVSA methods is also confirmed.

In our research, algorithms were coded in Pycharm 2020.3.1 x64. Simulations were also performed on an Intel core i5 computer with a Windows 10 operating system.

Tables 2 and 3 show the efficiency of the proposed algorithms in different benchmarks. DIAMCS and BHOSLIB are the machine-learning benchmarks mentioned in the related articles. Other datasets such as biological networks, social networks, web graphs, technological networks, etc. [39] are related to different networks. The results of the DIAMCS [40] and BHOSLIB [41] benchmarks are compiled in Table 2 and the results of other benchmarks are shown in Table 3. Here, $C^*$ and $|V|$ refer to the optimal solutions and the number of vertices, respectively. As shown in Tables 2 and 3, our proposed algorithms outperform MAMA, and thus outperform the others.

Table 2. The results of applying MDG, MAMA, and proposed algorithms on DIAMCS and BHOSLIB

| Benchmark | $\|V\|$ | $C^*$ | Cardinality of the vertex cover | | | | Approximation ratio | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MDG | MAMA | MaxA | MaxAR | MDG | MAMA | MaxA | MaxAR |
| Brock200-1 | 200 | 187 | 188 | 196 | 188 | **187** | 1.005 | 1.048 | 1.005 | 1.000 |
| Brock200-4 | 200 | 185 | 187 | 189 | **185** | **185** | 1.010 | 1.021 | 1.000 | 1.000 |
| C125 | 125 | 114 | 115 | 121 | **114** | 114.7 | 1.008 | 1.061 | 1.000 | 1.006 |
| C250.9 | 250 | 245 | 245 | 259 | 246 | 246 | 1.000 | 1.050 | 1.004 | 1.004 |
| C500.9 | 500 | 485.4 | 487 | 488 | 486 | **485.4** | 1.003 | 1.005 | 1.001 | 1.000 |
| c-fat200-1 | 200 | 128 | 139 | 180 | **128** | **128** | 1.080 | 1.406 | 1.000 | 1.000 |
| c-fat200-2 | 200 | 144 | 145 | 154 | **144** | 144.2 | 1.006 | 1.069 | 1.000 | 1.001 |
| c-fat500-1 | 200 | 341 | 345 | 352 | **341** | **341** | 1.011 | 1.032 | 1.000 | 1.000 |
| c-fat500-10 | 500 | 372 | 372 | 376 | **372** | 373 | 1.000 | 1.010 | 1.000 | 1.002 |
| c-fat500-2 | 500 | 361 | 363 | 372 | **361** | 361.4 | 1.005 | 1.030 | 1.000 | 1.001 |
| c-fat500-5 | 500 | 369 | 370 | 391 | 385 | **369** | 1.002 | 1.059 | 1.043 | 1.000 |
| Dsjc-500 | 500 | 488.9 | 491 | 495 | 490 | **488.9** | 1.004 | 1.012 | 1.002 | 1.000 |
| Fbr-30-15-2 | 450 | 423 | 423 | 432 | 429 | 429.5 | 1.000 | 1.021 | 1.014 | 1.015 |
| Fbr35-17-2 | 595 | 567 | 568 | 574 | **567** | **567** | 1.001 | 1.012 | 1.000 | 1.000 |
| Hamming10-2 | 1024 | 966 | 966 | 1007 | 982 | 982 | 1.000 | 1.042 | 1.016 | 1.016 |
| Hamming6-2 | 64 | 50 | 50 | 53 | 53 | 52 | 1.000 | 1.06 | 1.060 | 1.040 |
| Hamming6-4 | 64 | 46 | 47 | 50 | **46** | 46.1 | 1.020 | 1.086 | 1.000 | 1.002 |
| Hamming8-2 | 256 | 230 | 230 | 250 | 234 | 233 | 1.000 | 1.086 | 1.017 | 1.013 |
| Hamming8-4 | 256 | 230 | 230 | 246 | 234 | 234.9 | 1.000 | 1.069 | 1.017 | 1.021 |
| Johnson16-2-4 | 120 | 104 | 104 | 109 | 107 | 107 | 1.000 | 1.048 | 1.028 | 1.028 |
| Johnson32-2-4 | 496 | 463 | 463 | 475 | 469 | 469.6 | 1.000 | 1.025 | 1.012 | 1.014 |
| Johnson8-2-4 | 28 | 19 | 20 | 24 | **19** | **19** | 1.052 | 1.263 | 1.000 | 1.000 |
| Johnson8-4-4 | 70 | 58 | 58 | 70 | **58** | 58.1 | 1.000 | 1.206 | 1.000 | 1.001 |
| Keller4 | 171 | 151 | 151 | 158 | **151** | **151** | 1.000 | 1.046 | 1.000 | 1.000 |
| Keller5 | 776 | 741 | 745 | 750 | 742 | **741** | 1.005 | 1.012 | 1.001 | 1.000 |
| phat300-1 | 300 | 262.7 | 264 | 273 | 264 | **262.7** | 1.004 | 1.039 | 1.004 | 1.000 |
| phat300-2 | 300 | 273 | 275 | 278 | **273** | 274 | 1.007 | 1.018 | 1.000 | 1.003 |
| phat300-3 | 300 | 292 | 293 | 293 | **292** | **292** | 1.003 | 1.003 | 1.000 | 1.000 |
| phat700-1 | 700 | 640.9 | 641 | 645 | 642 | **640.9** | 1.000 | 1.017 | 1.004 | 1.000 |
| phat700-2 | 700 | 654 | 654 | 672 | 656 | 655.1 | 1.000 | 1.027 | 1.003 | 1.001 |
| phat700-3 | 700 | 692 | 693 | 705 | **692** | 692.3 | 1.001 | 1.018 | 1.000 | 1.000 |
| Sanr200-0.7 | 200 | 186 | 187 | 191 | **186** | 186.2 | 1.005 | 1.026 | 1.000 | 1.001 |
| Sanr200-0.9 | 200 | 187 | 187 | 190 | **187** | **187** | 1.000 | 1.016 | 1.000 | 1.000 |
| Sanr400-0.5 | 400 | 378 | 382 | 385 | 379 | **378** | 1.010 | 1.018 | 1.002 | 1.000 |
| Sanr400-0.7 | 400 | 384 | 386 | 391 | **384** | **384** | 1.005 | 1.018 | 1.000 | 1.000 |

Table 3. The results of applying MDG, MAMA, and proposed algorithms on benchmark instances [39]

| Benchmark | $\|V\|$ | $C^*$ | Cardinality of the vertex cover | | | | Approximation ratio | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MDG | MAMA | MaxA | MaxAR | MDG | MAMA | MaxA | MaxAR |
| bio-celegans | 453 | 255 | 259 | 285 | **255** | 255.2 | 1.015 | 1.117 | 1.000 | 1.000 |
| bio-diseasome | 516 | 285 | 285 | 371 | **285** | **285** | 1.000 | 1.301 | 1.000 | 1.000 |
| bio-dmela | 7393 | 2658 | 2666 | 2968 | **2658** | 2663.8 | 1.003 | 1.116 | 1.000 | 1.002 |
| bio-yeast | 1458 | 456 | 463 | 557 | **456** | 457 | 1.015 | 1.221 | 1.000 | 1.002 |
| ca-CSphd | 1882 | 550 | 556 | 739 | **550** | 551 | 1.010 | 1.343 | 1.000 | 1.001 |
| ca-Erdos992 | 6100 | 461 | 461 | 474 | **461** | **461** | 1.000 | 1.028 | 1.000 | 1.000 |
| ca-GrQc | 4158 | 2210.3 | 2219 | 2719 | 2211 | **2210.3** | 1.003 | 1.023 | 1.000 | 1.000 |
| ca-HepPh | 11204 | 6557.6 | 6574 | ٧٦١٦ | ٦٥٥٨ | **6557.6** | 1.002 | 1.161 | 1.000 | 1.000 |
| ca-netscience | 379 | 214 | 214 | 275 | **214** | **214** | 1.000 | 1.285 | 1.000 | 1.000 |
| Frb30-15-1 | 450 | 424 | 429 | 424 | 425 | 424.7 | 1.011 | 1.000 | 1.002 | 1.001 |
| Frb30-15-2 | 450 | 425 | 431 | 426 | **425** | **425** | 1.014 | 1.002 | 1.000 | 1.000 |
| Frb30-15-3 | 450 | 425 | 429 | 425 | 425 | 425.6 | 1.009 | 1.000 | 1.000 | 1.001 |
| ia-email-EU | 32430 | 820 | 820 | ٨٧٤ | **٨٢٠** | **820** | 1.000 | 1.065 | 1.000 | 1.000 |
| ia-email-univ | 1133 | 604.2 | 608 | ٦١٨ | ٦٠٥ | **604.2** | 1.006 | 1.022 | 1.001 | 1.000 |
| ia-enron-only | 143 | 87 | 87 | ٨٨ | ٩١ | 90.6 | 1.000 | 1.011 | 1.045 | 1.041 |
| ia-fb-essages | 1266 | 589.4 | 595 | ٥٩٣ | ٥٩٠ | **589.4** | 1.009 | 1.006 | 1.001 | 1.000 |
| ia-infect-dublin | 410 | 297 | 297 | ٣٠٥ | ٢٩٨ | 298.2 | 1.000 | 1.026 | 1.003 | 1.004 |
| ia-infect-hyper | 113 | 92 | 93 | ٩٣ | **٩٢** | **92** | 1.010 | 1.010 | 1.000 | 1.000 |
| ia-reality | 6809 | 81 | 81 | ٨١ | ٨١ | 81 | 1.000 | 1.000 | 1.000 | 1.000 |
| inf-power | 4941 | 2215 | 2275 | ٢٤٩٨ | **٢٢١٥** | 2221.8 | 1.027 | 1.127 | 1.000 | 1.003 |
| scc_enron-only | 151 | 138 | 139 | ١٣٨ | ١٣٨ | 138 | 1.007 | 1.000 | 1.000 | 1.000 |
| scc_fb-forum | 897 | 372 | 372 | ٣٧٣ | ٣٧٢ | 372 | 1.000 | 1.000 | 1.000 | 1.000 |
| scc_infect-hyper | 113 | 110 | 110 | ١١٠ | ١١٠ | 110 | 1.000 | 1.000 | 1.000 | 1.000 |
| scc_retweet | 18469 | 562 | 562 | ٥٩٦ | ٥٦٣ | 563.3 | 1.000 | 1.060 | 1.001 | 1.002 |
| scc_rt_lolgop | 9742 | 103 | 103 | ١١٤ | **١٠٣** | **103** | 1.000 | 1.106 | 1.000 | 1.000 |
| tech-routers-rf | 2113 | 796 | 805 | 939 | **٧٩٦** | **796** | 1.011 | 1.179 | 1.000 | 1.000 |
| tech-WHOIS | 7476 | 2288 | 2298 | ٢٦٥٣ | **٢٢٨٨** | 2288.7 | 1.004 | 1.159 | 1.000 | 1.000 |
| web-erkStan | 12305 | 5492 | 5492 | ٦٧٦٤ | ٥٦٦٨ | 5640.3 | 1.000 | 1.231 | 1.032 | 1.027 |
| web-edu | 3031 | 1451 | 1587 | ١٥٨٤ | **١٤٥١** | **1451** | 1.093 | 1.091 | 1.000 | 1.000 |
| web-indochina-2004 | 11358 | 7300 | 7424 | ٨٢٢٤ | **٧٣٠٠** | **7300** | 1.016 | 1.126 | 1.000 | 1.000 |
| web-spam | 4767 | 2321 | 2346 | ٢٤٩٧ | **٢٣٢١** | 2322.6 | 1.010 | 1.075 | 1.000 | 1.000 |
| web-webbase-2001 | 16062 | 2667 | 2687 | ٣٣٤٣ | **٢٦٦٧** | 2668 | 1.007 | 1.253 | 1.000 | 1.000 |

As shown in Table 3, the proposed algorithms performed better on the same benchmarks, but in general both algorithms showed better coverage rates on sparse graphs compared to the other methods where in dense graphs, their response is often the same as the other methods. The average and the best approximation ratio for MAMA, MDG, and our proposed algorithm are shown in Table 4 where the approximation ratio is equal to $\rho = \frac{\text{Cardinality of the vertex cover}}{C^*}$. The superiority of the proposed algorithms is evident from the ratios in Table 4.

Table 4. Comparison of approximation ratio based on experimental result

| Algorithms | Worst $\rho$ | Average $\rho$ |
|---|---|---|
| MDG | 1.093 | 1.040 |
| MAMA | 1.343 | 1.098 |
| MaxA | 1.045 | 1.002 |
| MaxAR | 1.041 | 1.002 |

In most of the benchmarks instances, MAMA, MaxA, and MaxAR algorithms take the same execution time. However, experimental results have shown that in seven benchmarks, MAMA required more time than other algorithms to run. These results are shown in Table 5.

Table 5. Comparison of execution time

| Benchmark | MDG $\|ms\|$ | MAMA $\|ms\|$ | MaxA $\|ms\|$ | MaxAR $\|ms\|$ |
|---|---|---|---|---|
| tech-routers-rf | 0.48 | 15.89 | 0.73 | 0.81 |
| tech-WHOIS | 9.46 | 1140.76 | 18.36 | 18.73 |
| web-BerkStan | 25.38 | 271.34 | 25.65 | **24.34** |
| web-edu | 1.44 | 7.95 | 1.29 | **1.12** |
| web-indochina-2004 | 49.06 | 3131.18 | 87.17 | 89.13 |
| web-spam | 7.34 | 611.51 | 17.33 | 16.16 |
| web-webbase-2001 | 13.47 | 1168.00 | 27.35 | 24.73 |

# 5. Conclusion

The well-known minimum vertex cover problem cannot be solved in polynomial time. In this article, two heuristic algorithms MaxA and MaxAR were introduced both of which can be used for any large and small graph. The results revealed that the proposed algorithms had better performance in sparse graphs, that is, as the number of vertices increased, the efficiency of the proposed algorithms became more apparent. It is expected that the proposed algorithms will be more efficient in large sparse graphs.

**References**

[1] Sachdeva, K., 2012. Applications of Graphs in Real-Life. *International Journal of Advanced Research in Computer Science*, *3*(1).
[2] Silberstein, A. and Yang, J., 2007, April. Many-to-many aggregation for sensor networks. In *2007 IEEE 23rd International Conference on Data Engineering* (pp. 986-995). IEEE.
[3] Yigit, Y., Akram, V.K. and Dagdeviren, O., 2021. Breadth-first search tree integrated vertex cover algorithms for link monitoring and routing in wireless sensor networks. *Computer Networks*, *194*, p.108144.
[4] Khamayseh, Y., Mardini, W. and Shatnawi, A., 2012. An approximation algorithm for vertex cover problem. In *International Conference on Computer Networks and Communication Systems*.
[5] Cheetham, J., Dehne, F., Rau-Chaplin, A., Stege, U. and Taillon, P.J., 2003. Solving large FPT problems on coarse-grained parallel machines. *Journal of Computer and System Sciences*, *67*(4), pp.691-706.
[6] Yigit, Y., Dagdeviren, Z.A., Dagdeviren, O. and Challenger, M., 2021, October. Performance Evaluation of Capacitated Vertex Cover Algorithms for Security Applications in Wireless Sensor Networks. In *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)* (pp. 619-624). IEEE.
[7] Quan, C. and Guo, P., 2021. A local search method based on edge age strategy for minimum vertex cover problem in massive graphs. *Expert Systems with Applications*, *182*, p.115185.
[8] Pattabiraman, B., Patwary, M., Ali, M., Gebremedhin, A.H., Liao, W.K. and Choudhary, A., 2013, December. Fast algorithms for the maximum clique problem on massive sparse graphs. In *International Workshop on Algorithms and Models for the Web-Graph* (pp. 156-169). Springer, Cham.
[9] Pattabiraman, B., Patwary, M.M.A., Gebremedhin, A.H., Liao, W.K. and Choudhary, A., 2015. Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection. *Internet Mathematics*, *11*(4-5), pp.421-448.
[10] San Segundo, P., Lopez, A. and Pardalos, P.M., 2016. A new exact maximum clique algorithm for large and massive sparse graphs. *Computers & Operations Research*, *66*, pp.81-94.
[11] Verma, A., Buchanan, A. and Butenko, S., 2015. Solving the maximum clique and vertex coloring problems on very large sparse networks. *INFORMS Journal on computing*, *27*(1), pp.164-177.
[12] Dinur, I. and Safra, S., 2005. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pp.439-485.
[13] Papadimitriou, C.H. and Yannakakis, M., 1991. Optimization, approximation, and complexity classes. *Journal of computer and system sciences*, *43*(3), pp.425-440.
[14] Balaji, S., Swaminathan, V. and Kannan, K., 2010. Optimization of unweighted minimum vertex cover. *World Academy of Science, Engineering and Technology*, *43*, pp.716-729.
[15] Fayaz, M. and Arshad, S., 2015, December. Clever Steady Strategy Algorithm: A Simple and efficient approximation algorithm for minimum vertex cover problem. In *2015 13th International Conference on Frontiers of Information Technology (FIT)* (pp. 277-282). IEEE.
[16] Fayaz, M., Arshad, S., Zaman, U. and Ahmad, A., 2016, December. A simple, fast and near optimal approximation algorithm for optimization of un-weighted minimum vertex cover. In *2016 International Conference on Frontiers of Information Technology (FIT)* (pp. 176-180). IEEE.
[17] Eremeev, A.V., 2000. On some approximation algorithms for dense vertex cover problem. In *Operations Research Proceedings 1999* (pp. 58-62). Springer, Berlin, Heidelberg.
[18] Pelikan, M., Kalapala, R. and Hartmann, A.K., 2007, July. Hybrid evolutionary algorithms on minimum vertex cover for random graphs. In Proceedings of the 9th annual conference on Genetic and evolutionary computation (pp. 547-554).
[19] Back, T. and Khuri, S., 1994, June. An evolutionary heuristic for the maximum independent set problem. In Proceedings of the first IEEE conference on evolutionary computation. IEEE World Congress on Computational Intelligence (pp. 531-535). IEEE.
[20] Evans, I.K., 1998, March. Evolutionary algorithms for vertex cover. In International Conference on Evolutionary Programming (pp. 377-386). Springer, Berlin, Heidelberg.
[21] Guo, P., Quan, C. and Chen, H., 2019. MEAMVC: A membrane evolutionary algorithm for solving minimum vertex cover problem. IEEE Access, 7, pp.60774-60784.
[22] Karci, A. and Arslan, A., 2003. Bidirectional evolutionary heuristic for the minimum vertex-cover problem. Computers & Electrical Engineering, 29(1), pp.111-120.
[23] Cai, S., Su, K., Luo, C. and Sattar, A., 2013. NuMVC: An efficient local search algorithm for minimum vertex cover. Journal of Artificial Intelligence Research, 46, pp.687-716.
[24] Cai, S., 2015, June. Balance between complexity and quality: Local search for minimum vertex cover in massive graphs. In Twenty-Fourth International Joint Conference on Artificial Intelligence.
[25] Cai, S., Su, K. and Chen, Q., 2010, July. EWLS: A new local search for minimum vertex cover. In Twenty-Fourth AAAI Conference on Artificial Intelligence.

[26] Cai, S., Su, K. and Sattar, A., 2011. Local search with edge weighting and configuration checking heuristics for minimum vertex cover. Artificial Intelligence, 175(9-10), pp.1672-1696.
[27] Cai, S., Lin, J. and Su, K., 2015, February. Two weighting local search for minimum vertex cover. In Twenty-ninth AAAI conference on artificial intelligence.
[28] Cai, S., Lin, J. and Luo, C., 2017. Finding a small vertex cover in massive sparse graphs: Construct, local search, and preprocess. Journal of Artificial Intelligence Research, 59, pp.463-494.
[29] Fang, Z., Chu, Y., Qiao, K., Feng, X. and Xu, K., 2014, June. Combining edge weight and vertex weight for minimum vertex cover problem. In International Workshop on Frontiers in Algorithmics (pp. 71-81). Springer, Cham.
[30] Katzmann, M. and Komusiewicz, C., 2017, February. Systematic exploration of larger local search neighborhoods for the minimum vertex cover problem. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).
[31] Luo, C., Hoos, H.H., Cai, S., Lin, Q., Zhang, H. and Zhang, D., 2019, August. Local Search with Efficient Automatic Configuration for Minimum Vertex Cover. In IJCAI (pp. 1297-1304).
[32] Pullan, W., 2009. Optimisation of unweighted/weighted maximum independent sets and minimum vertex covers. Discrete Optimization, 6(2), pp.214-219.
[33] Richter, S., Helmert, M. and Gretton, C., 2007, September. A stochastic local search approach to vertex cover. In annual conference on artificial intelligence (pp. 412-426). Springer, Berlin, Heidelberg.
[34] Wagner, M., Friedrich, T. and Lindauer, M., 2017, June. Improving local search in a minimum vertex cover solver for classes of networks. In 2017 IEEE Congress on Evolutionary Computation (CEC) (pp. 1704-1711). IEEE.
[35] Wagner, M., Friedrich, T. and Lindauer, M., 2017, June. Improving local search in a minimum vertex cover solver for classes of networks. In 2017 IEEE Congress on Evolutionary Computation (CEC) (pp. 1704-1711). IEEE.
[36] Pelikan, M., Kalapala, R. and Hartmann, A.K., 2007, July. Hybrid evolutionary algorithms on minimum vertex cover for random graphs. In Proceedings of the 9th annual conference on Genetic and evolutionary computation (pp. 547-554).
[37] Witt, C., 2012. Analysis of an iterated local search algorithm for vertex cover in sparse random graphs. Theoretical Computer Science, 425, pp.117-125.
[38] Eren Y, Küçükdemiral IB, Üstoğlu İ. Chapter 2- introduction to optimization, optimization in renewable energy system.
[39] https://networkrepository.com/networks.php
[40] https://networkrepository.com/dimacs.php
[41] https://networkrepository.com/bhoslib.php
[42] Zhang YJ, Mu XD, Liu XW, Wang XY, Zhang X, Li K, Wu TY, Zhao D, Dong C. Applying the quantum approximate optimization algorithm to the minimum vertex cover problem. Applied Soft Computing. 2022 Mar 1;118:108554.
[43] Gusev VV. The vertex cover game: Application to transport networks. Omega. 2020 Dec 1;97:102102.
[44] Wang B, Sun Y, Duong TQ, Nguyen LD, Hanzo L. Risk-aware identification of highly suspected COVID-19 cases in social IoT: a joint graph theory and reinforcement learning approach. Ieee Access. 2020 Jun 19;8:115655-61.
.

**Vida Barzegaran**, received her B.S. degree in Computer Engineering in 2021 from Shahid Rajaee University. She is currently pursuing M.S. in Computer System Architecture with Amirkabir University of Technology. Her research interests include computer arithmetic and optimal operation of integrated energy systems.
E-mail: vidi@aut.ac.ir

**Zeinab Torabi** received her Ph.D. degree in Computer System Architecture from Shahid Beheshti University, in 2016. She is currently an Assistant Professor of Computer Engineering in the faculty of computer engineering, Shahid Rajaee Teacher Training University. Her research interests include computer arithmetic and RNS.
E-mail: z.torabi@sru.ac.ir

**Sahar Kianian** received her B.Sc. degree in computer Engineering (2007) from razi University, also M.Sc. and Ph.D. degrees in computer Engineering from Isfahan University (2010 and 2016, respectively). She is an assistant professor of computer engineering at Shahid Rajaee University. Her research interests are the application of algorithms, machine learning and data science to complex networks, focuses on protein interactions, connections of neurons and relationships among people. Applications include disease prediction, drug discovery, event detection and tracking, recommendation system, web mining and social influence mining. E-mail: sahar.kianian@sru.ac.ir