International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 08, ISSN No. 2455-2143, Pages 66-72
Published Online December 2023 in IJEAST (http://www.ijeast.com)

A STUDY OF ANT COLONY OPTIMIZATION APPLICATION FOR SOLVING MINIMUM VERTEX COVER PROBLEM

Akshaya Kumar Mandal
Department of Computer Science, Assam University,
A Central University of India, Assam, Silchar- 788011, Assam, India

Abstract—**This article presents a meticulous exploration of the Minimum Vertex Cover (MVC) Problem, offering a detailed and step-by-step mathematical derivation of the Ant Colony Optimization (ACO) algorithm tailored expressly for its resolution. The MVC Problem is introduced with an emphasis on its significance and inherent complexities. The principles of ACO are expounded upon, laying the foundation for a comprehensive mathematical formulation. Each facet of the ACO algorithm, from initialization to pheromone updates, solution construction, and the nuanced balance between exploration and exploitation, is dissected. Through the illustration of a practical example, the article showcases the ACO algorithm's application, providing insights into its practicality and efficacy in solving specific instances of the MVC Problem. This work contributes to an enriched understanding of ACO's potential in addressing intricate combinatorial optimization challenges, serving as a valuable guide for researchers and practitioners seeking to implement and explore this approach further application in various domains.**

Keywords—**Ant Colony Optimization, Minimum Vertex Cover, Combinatorial Optimization, Mathematical Derivation, Heuristic Algorithms**

I. INTRODUCTION

In the realm of graph theory, the Minimum Vertex Cover Problem (MVC) stands as a fundamental challenge with wide-ranging implications. Take into account an undirected, unweighted graph denoted by G = (V, E), where V stands for the vertices set, and E denotes the edges set within the graph [4-10]. The crux of the Minimum Vertex Cover Problem lies in identifying a subset of vertices, drawn from V that possesses a unique property: for every edge in E, at least one of its two endpoints must belong to this carefully curated subset. This process aims to minimize the cardinality of the chosen subset, creating what is known as the minimum vertex cover set. Formally, a non-empty vertex set S ⊆ V is deemed a vertex cover set if each edge in E finds at least one of its endpoints within S [10-22]. The quest for the minimum vertex cover set within graph G becomes a compelling endeavor, seeking to uncover the most efficient and streamlined solution within the intricate network of vertices and edges.

Lemma 1: Any graph G= (V, E), and for any subset S ⊆ V, two statements are interchangeable:

- S constitutes the MVC set in G.
- The complement of S is in V, denoted as V-S, forms the independent vertex set in G.

The MVC is known as an NP-complete problem [33], widely employed to model diverse real-life scenarios across various domains such as circuit design, telecommunications, network flow, and medical science, including applications in the detection of cancer cells. This intricate problem, renowned for its computational complexity, plays a pivotal role in decision-making processes where the objective is to determine the smallest subset of vertices in a graph. The MVC's versatility is evident in its application to practical challenges, making it a valuable tool for addressing complex issues in fields ranging from technology and communication to healthcare [22-32]. As an NP-complete problem, MVC encapsulates the inherent difficulty of finding optimal solutions within a reasonable timeframe, adding a layer of complexity that mirrors the intricacies of real-world problem-solving [34, 35].

A. Application of ACO to the MVC

The structure of the Minimum Vertex Cover (MVC) problem diverges from other problems previously tackled by ACO in the existing literature [1-3, 22]. In contrast to many ACO-solved problems, where solutions often manifest as ordered or unordered subsets of edges [8], MVC poses a distinctive challenge. The solution space for MVC comprises unordered subsets of vertices, each derived by individual ants during each cycle of the algorithm. This distinction introduces an added layer of complexity, necessitating an innovative approach in representing MVC within a graph structure. Furthermore, the development of local heuristics for the state transition rule and the subsequent update of the pheromone trail become notably intricate. The unique characteristics of MVC demand a tailored adaptation of ACO, marking this exploration as an intriguing endeavor in the intersection of graph theory and optimization algorithms.

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 08, ISSN No. 2455-2143, Pages 66-72
Published Online December 2023 in IJEAST (http://www.ijeast.com)

B. Approximation algorithm for the MVC Problem [32]
Input: Graph G = (V,E)

Output: A vertex cover C for G with size no more than twice the size of the optimal vertex cover.
begin
S← ∅
while E is not empty
Randomly select an edge (i, j)
Add i and j to S
Eliminate all edges connected to either vertex i or vertex j from the edge set E.
end
return S
end

II. GRAPH REPRESENTATION OF MVC PROBLEM

Consider G=(V,E) as the graph representing the Minimum Vertex Cover (MVC) problem, with the solution to this instance being an unordered subset of vertices denoted as S ⊆ V [5-7]. To facilitate the application of the ACO Algorithm, to create a complete graph G_c= (V, E_c) as the complement of G. Within this complete graph, the ACO Algorithm [1-3] guides ants to choose the next vertex randomly. The connectivity function for each path in the Figure 1 is precisely defined to regulate the exploration process.

C_k:E_c→ {0,1} for each edge (i, j) as:

$$C_k(i,j) = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E_c - E \end{cases} \qquad (1)$$

The adjacent value for each path, a clear distinction is established between the edges in E(G) and those in E(G_c) - E(G). Simultaneously, this connected value serves as the pheromone. For any given ant, denoted as 'k', initiating the exploration from any vertex, that ensure the ant k avoids selecting vertices it has already reached by updating the adjacent value of the edge (i, j) upon reaching the point j. Specifically, C (i, j) is set to 0 for j ∈ S, where S represents an arbitrarily chosen set of initial points for ants. This precautionary measure is essential to maintain the integrity of the exploration process and prevent ants from revisiting previously explored vertices.

A. Updating Transition Formula
when one of the vertices is visited by an ant, the cost of the edge in E is set to 0 using rule, denoted as C_k(i, j)=0, if edge(i, j)∈E and either vertex i or j is visited by ant K. Subsequently, the connectivity value of the edge undergoes further updating, following the specified protocol [5]: C(i,j) = 1/n, if edge (i, j)∈E and either vertex i or j is visited by ant K, where n represents the number of vertices in the graph. The maximum cost, denoted as:C_j^k, signifies the higher preference for selecting vertex j. Before entering the subsequent cycle, a reset the cost for all edges is imperative using equation (1) to reinstate the cost information of each path. Subsequently, the total values for the paths incident with each vertex in graph G are recalculated.

B. State transition rule
In most problems solved by ACO, solutions are derived from the power set of the edge set. However, as the solution to the MVC is obtained through subsets of vertices, each vertex is assigned a maximum value. Consequently, the state transition probability for ant k is defined to determine the likelihood of selecting the next vertex using probability:

$$P_j^k = \frac{\tau^\alpha \pi_{jk}^\beta}{\sum_{j \in S_i} \tau^\alpha \pi_{jk}^\beta} \qquad (2)$$

Where S_k represents the set of vertices accessible to ant k, τ^α and π_{jk}^β denote the global pheromone updating factor and the value of variable denoted as π_{jk}. The variable π_{jk} evaluates the local preference for vertex j for ant k according to the heuristic function [5].

$$\pi_{jk} = \frac{\sum_{(i,j) \in E_c} C(i,j)}{W(j)} \qquad (3)$$

Here, π_{jk} represents the count of edges not covered by ant k that are currently linked to vertex j, divided by the weight of vertex j. In this context, it is assumed that the vertex 'l' has the highest value. Subsequently, the state transition probability of ant k is defined and utilized to determine the selection of the next vertex.

$$P_i^k = \begin{cases} 1, i = l \\ 0, i \neq l \end{cases} \qquad (4)$$

For any given vertex 'i', if C_i^k = 0, it indicates that the connected value for each edge becomes zero in graph G_C, when at least one endpoint for each edge has been traversed by an ant in graph
G. This condition acts as a termination criterion for the ant colony optimization algorithm. In contrast, when G_C is non-zero, the connected values undergo an update, and the process of selecting the next vertex is sustained. This iterative operation continues until the connected value for every vertex reaches zero, culminating in the establishment of a distinct vertex cover set S_k.

C. Pheromone updating rule
The update of the connectivity value, denoted as C (i, j), for an edge in E occurs when one of its vertices has been visited by a specific ant, say ant k, as outlined in the following manner: C (i, j) = 1/n, if edge (i, j) ∈ E and either vertex i or j has been visited by ant k, where n represents the number of graph

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 08, ISSN No. 2455-2143, Pages 66-72
Published Online December 2023 in IJEAST (http://www.ijeast.com)

vertices (|V|). Importantly, at the conclusion of each cycle, the pheromone left on the vertices of the presently best solution, denoted as S, is addressed. To achieve this without redundancy, for each vertex i ∈ S, the pheromone is updated following the global updating rule [1-3]: $\tau_i = (1-\rho)\tau_i$

Where
$\Delta\tau_i = \frac{1}{|s_k|}$ and $\rho \in (0,1)$, represents a parameter that emulates the rate of pheromone intensity evaporation.
So the updated pheromone is: $\tau_i = \tau_i + \rho\Delta\tau_i$

D. Stopping Criterion
The ACO's termination criterion can materialize as a maximum iteration count, a fixed CPU time limit, or a specified number of consecutive iterations yielding the optimal algorithmic improvement. This paper takes a distinct route, opting for a predefined number of iterations where no progress in the solution is observed.

III. ALGORITHM OF ACO FOR MVC

Step-1: Initialization parameters: Utilizing the following formula to assign the adjacent value for each path in the graph:G_c.

$$C(i,j) = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E_c - E \end{cases}$$

andS_k=φ,i=1,k=1, Choose a vertices randomly as the starting points and position the ant on the graph.

Step-2: Apply function C(i, j)=0 to update the adjacent values of all paths incident with vertex i and compute $C_j^k = \frac{1}{n}(j \in V)$.If $C_j^k = 0$, we get the vertex cover set S_k. Else if k<H, let k = k +1 ,i = i +1, then repeat Step 1.
else
ifk=H, execute Step 4.
else execute Step3.

Step-3: Calculate C_j^k , if P_i^k =1 , Find vertex has the maximum value, so $S_k = S_k \cup \{u_i\}$, go to step-2.

Step-4: Let S= min{| S_1 |, … … … … …, | S_H |},If S constitutes an approximate minimum visited vertex set of graph.

A. The Time Complexity of Algorithm
Step-1: Involves n (n-1)/2 for constructing graph Gc.
Step-2: necessitates n-1 for updating the connected values of all paths adjacent to vertex i.
Step-3: required n-1 steps to computing C_i^k
Step-4: Execute H-1 times for getting MVC set.
Therefore, the algorithm's time complexity can be expressed as: O ($2n^2$+2n-2) = O (n^2)

B. Example of the ACO in MVC
We utilize the graph G depicted in Fig. 1, to exemplify the algorithm's concrete implementation, leading to the derivation of an approximate MVC for graph G.

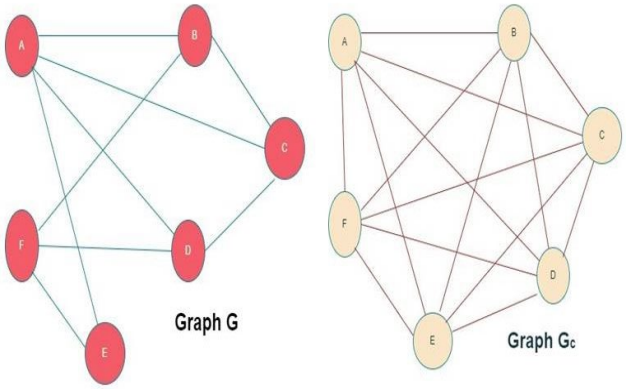

Figure 1 Graph Representation of MVC problems with 6-vertices

C. Initialization of parameters
Before initiating the step-by-step mathematical derivation of the ACO for solving the MVC Problem, it is crucial to initialize key parameters: $\alpha = 1, \beta = 1, \rho = 0$ and $\tau_{ij} = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E \end{cases}$

D. A Step by Step Illustration
Step-1 (ant-1)(iteration-1)
Assuming vertex A as the starting point for ant 1 in the exploration of the visited vertex set, the connected values for each path in graph G are as: follows: C(A,B)=C(A,C)=C(A,D)=C(A,E)=C(B,C)=C(B,F)=C(C,D)=C(D,F)=C(E,F)=1 and set other paths have the value zero. Initialized S_1= {A}.

Step-2 Assign the values of all paths to be displayed at vertex A be 0, and then find C_i^1 ,
$C_B^1 = C_C^1 = C_D^1 = \frac{1}{6}$+1+1=2.18, C_F^1=3

Step-3: C_F^1=3 is the more then S_1= {A,F},repeat step 2.

Step-2: Assign the values of all paths to be displayed at vertex F be 0, and then compute C_i^1 ,

$C_B^1 = C_D^1 = \frac{1}{5}$+1=1.02,$C_E^1 = \frac{1}{5} = 0.2$, C_C^1= 1+1=2 repeat step 3

Step-3: C_c^1=2 is the more then S_1={A,F,C},goto step 2.

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 08, ISSN No. 2455-2143, Pages 66-72
Published Online December 2023 in IJEAST (http://www.ijeast.com)

Step-4: Revise the values for all adjacent paths, and $C_A^1=C_B^1=C_c^1=C_D^1=C_E^1=C_F^1=0$
So the terminate it steps. Result is S1 = {A, F, C}.

Step-1 (ant-2)(iteration-2)
Assuming vertex B serves as the starting point for ant 2 in exploring the visited vertex set, the for each adjacent path in graph G_c are as follows:
C(B,A)=C(A,C)=C(A,D)=C(A,E)=C(B,C)=C(B,F)=C(C,D)=C(D,F)=C(E,F)=1 and set other paths have the value zero . Initialized S_2={B}.

Step-2: Set all adjacent paths incident with vertex B to 0, and subsequently calculate C_i^2 :
$C_A^2=\frac{1}{6}+1+1+1=3.16$, $C_C^2=C_F^2=\frac{1}{6}$+1+1=2.18,
C_E^1=1+1=2,C_E^1=1+1+1=3,
Step-3: C_A^2=3.16 is the maximum then S_2={B,A},goto step 2.

Step-2: Set all adjacent paths incident with vertex A to 0, and subsequently calculate:C_i^2 ,$C_E^1=C_C^1=\frac{1}{5}$+1=1.02,C_D^1 =1+1+1 = 3, C_F^1= 1+1=2 goto step 3

Step-3: C_D^2=3 is the more then S_2={B,A,D},goto step 2.

Step-2: Set all adjacent paths incident with vertex D to 0, and subsequently calculate:C_i^2,C_E^1= 1 , C_C^1=0,, C_F^1= 1+$\frac{1}{4}$=1.25 goto step 3

Step-3: C_F^2=1.25 is the more then S_2={B,A,D,F},goto step 2.

Step-4: Revise all adjacent paths, and $C_A^2=C_B^2=C_c^2=C_D^2=C_E^2=C_F^2=0$
So the terminate it steps. Output is S2 = {B, A, D, F}.

Step-1 (ant-3)(iteration-3)
Assuming vertex C serves as the starting point for ant 3 in exploring the visited vertex set, the for each adjacent path in graph G_c are as follows:
C(A,B)=C(A,C)=C(A,D)=C(A,E)=C(B,C)=C(B,F)=C(C,D)=C(D,F)=C(E,F)=1 and other edges have the value zero. Let S_3={C}.

Step-2: Set all adjacent paths incident with vertex C to 0, and subsequently calculateC_i^3 , $C_A^3=\frac{1}{6}$+1+1+1=3.16, $C_B^3=C_D^1=\frac{1}{6}$+1+1=2.18, C_F^3=3,C_E^3=2

Step-3:C_A^3=3.16 is the more then S_3={C,A},goto step 2.

Step-2: Set all adjacent paths incident with vertex A to 0, and subsequently calculate, C_i^3 , $C_B^3=C_D^3=C_E^3=\frac{1}{5}$+1=1.02,$C_F^3$= 1+1+1=3 goto step 3

Step-3: C_F^3=3 is the more then S_3={C,A,F},goto step 2.

Step-4: Revise all adjacent paths, and $C_A^3=C_B^3=C_c^3=C_D^3=C_E^3=C_F^3=0$
So the terminate it steps. Output is S3 = {C, A, F}.

Step-1 (ant-4)(iteration-4)
Assuming vertex D as the starting point for ant 4 in the exploration of the vertex cover set, the connected values for each edge in graph G_c are as follows:
C(B,A)=C(A,C)=C(A,D)=C(A,E)=C(B,C)=C(B,F)=C(C,D)=C(D,F)=C(E,F)=1 and other edges have the value zero. Let S_4={D}.

Step-2: Set all adjacent paths incident with vertex B to 0, and subsequently calculateC_i^2 ,
$C_A^4=\frac{1}{6}+1+1+1=3.16$, $C_C^4=C_F^4=\frac{1}{6}$+1+1=2.18,
C_E^4=1+1=2,C_F^4=1+1+1=3,

Step-3: C_A^4=3.16 is the more then S_4={D,A},goto step 2.

Step-2: Set all adjacent paths incident with vertex D to 0, and subsequently calculateC_i^4 ,

$C_E^4=C_C^4=\frac{1}{5}$+1=1.02,C_B^4 =$\frac{1}{5}$+1+1 = 2.2, C_F^4= 1+1=2 goto step 3

Step-3: C_B^4=2.2 is the more then S_4={D,A,B},goto step 2.

Step-2: Set all adjacent paths incident with vertex B to 0, and subsequently calculateC_i^4 ,

C_E^4= 1 , $C_C^1=\frac{1}{4}$,, C_F^1= 1+$\frac{1}{4}$=1.25 goto step 3

Step-3: C_F^4=1.25 is the more then S_4={D,A,B,F},goto step 2.

Step-4: Revise the connected values for all edges, and $C_A^4=C_B^4=C_c^{41}=C_D^{14}=C_E^4=C_F^4=0$
So the terminate it steps. Output is S4 = {D, A, B, F}.

Step-1 (ant-5)(iteration-5)
Assuming vertex E serves as the starting point for ant 5 in exploring the visited vertex set, the for each adjacent path in graph G_c are as follows:
C(B,A)=C(A,C)=C(A,D)=C(A,E)=C(B,C)=C(B,F)=C(C,D)=C(D,F)=C(E,F)=1 and other edges have the value zero. Let S_5={E}.

Step-2: Set all adjacent paths incident with vertex B to 0, and subsequently calculate C_i^2 ,
$C_A^5=\frac{1}{6}+1+1+1=3.16$, $C_B^5=C_C^5=C_D^5$=1+1+1=3,
$C_F^5=\frac{1}{6}$+1+1=2.16,

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 08, ISSN No. 2455-2143, Pages 66-72
Published Online December 2023 in IJEAST (http://www.ijeast.com)

Step-3: C_A^5=3.16 is the more then S_5={E,A},goto step 2.

Step-2: Set all adjacent paths incident with vertex D to 0, and subsequently calculateC_i^5 ,

C_B^5=C_C^5=C_D^5 =$\frac{1}{5}$+1+1 = 2.2, C_F^4= 1+1=2 goto step 3

Step-3: C_B^5=2.2 is the more then S_5= {E, A, B},goto step 2.

Step-2: Set all adjacent paths incident with vertex B to 0, and subsequently calculate, C_i^5 ,

C_D^5= 1 + 1 = 2 , C_C^5=C_F^5= 1+$\frac{1}{4}$=1.25 goto step 3

Step-3: C_D^5=2 is the more then S_5={E,A,B,D},goto step 2.

Step-4: Revise the connected values for all edges, and C_A^5=C_B^5=C_c^5=C_D^5=C_E^5=C_F^5=0
So the terminate it steps. Output is S5 = {E,A,B,D}.

Step-1 (ant-6) (iteration-6)
Assuming vertex F serves as the starting point for ant 6 in exploring the visited vertex set, the for each adjacent path in graph G_c are as follows:
C(A,B)=C(A,C)=C(A,D)=C(A,E)=C(B,C)=C(B,F)=C(C,D)=C(D,F)=C(E,F)=1 and other edges have the value zero. Let S_6={F}.

Step-2: Set all adjacent paths incident with vertex F to 0, and subsequently calculateC_i^6 ,
C_B^6=C_D^6=$\frac{1}{6}$+1+1=2.18, C_A^6=4,C_C^6=1+1+1=3

Step-3: C_A^6=4 is the more then S_6={F,A},goto step 2.

Step-2: Set all adjacent paths incident with vertex A to 0, and subsequently calculateC_i^6 ,
C_B^1=C_D^1=$\frac{1}{5}$+1=1.02,C_E^1 =$\frac{1}{5}$ = 0.2, C_C^1=$\frac{1}{5}$ + 1+1=2.2 goto step 3

Step-3: C_c^1=2.2 is the more then S_6={F,A,C},goto step 2.

Step-4: Revise the connected values for all edges, and C_A^6=C_B^6=C_c^6=C_D^6=C_E^6=C_F^6=0
So the terminate it steps. Output is S6 = {F, A, C}.

Table 1 Results of ACO-MVC Computation

Starting point	Ant	MVC
'A'	1	S_1={'A', 'F', 'C'},
'B'	2	S_2={'B', 'A', 'D', 'F'}
'C'	3	S_3 = {'C', 'A', 'F'}
'D'	4	S_4 = {'D', 'A', 'B', 'F'}.
'E'	5	S_5 = {'E', 'A', 'B', 'D'}.
'F'	6	S_6 = {'F', 'A', 'C'}.

We have S_1=S_3 = S_6 = {A, F, C} and all of them are also the MVC sets. So,
S=min {| S_1 |, … … … … …, | S_6 |},ie.| S_1 |=| S_3 |=| S_6 |=3

IV. CONCLUSION

In conclusion, this article has meticulously unveiled a step-by-step mathematical derivation of an ACO algorithm specifically designed for the MVC Problem. Demonstrating its effectiveness through the identification of optimal solutions in various vertex sets, the derived ACO algorithm showcases promise in addressing the computational complexities inherent in MVC. The carefully defined state transition probability empowers the algorithm to intelligently explore vertex cover sets, as illustrated in a practical example, highlighting its feasibility and accessibility. Future work could focus on refining the algorithm, exploring additional heuristics, and investigating its adaptability to dynamic problem instances. Moreover, the integration of parallel computing or hybrid approaches and comparative studies with alternative optimization techniques would contribute to a deeper understanding of the algorithm's capabilities and limitations, fostering advancements in solving intricate combinatorial optimization challenges.

V. REFERENCES

[1] Mandal, A. K.; Dehuri, S. (2020). A survey on ant colony optimization for solving some of the selected np-hard problem. In International Conference on Biologically Inspired Techniques in Many-Criteria Decision Making, Springer, Cham, 85--100.

[2] Mandal, A. K.; Sarma, P.K.D.; Dehuri, S. (2023). A Study of Bio-inspired Computing in Bioinformatics: A State-of-the-art Literature Survey. The Open Bioinformatics Journal, 2023 Jun 23; 16(1). https://doi.org/10.2174/18750362-v16-e230517-2022-17

[3] Mandal, A. K.; Sarma, P.K.D. (2022). Novel Applications of Ant Colony Optimization with the Traveling Salesman Problem in DNA Sequence Optimization. In2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC) 2022 Dec 15 (pp. 1-6). IEEE.https://doi.org/10.1109/iSSSC56467.2022.10051206

[4] Mehrabi, A. D., Mehrabi, S., &Mehrabi, A. (2009, October). A pruning based ant colony algorithm for minimum vertex cover problem. In International Conference on Evolutionary Computation (Vol. 2, pp. 281-286). SCITEPRESS.

[5] Shyu, S. J., Yin, P. Y., & Lin, B. M. (2004). An ant colony optimization algorithm for the minimum weight vertex cover problem. Annals of Operations Research, 131, 283-304.

[6] Andrade, D. V., Resende, M. G., &Werneck, R. F. (2012). Fast local search for the maximum independent set problem. Journal of Heuristics, 18, 525-547.

[7] Cai, S., & Lin, J. (2016, July). Fast Solving Maximum Weight Clique Problem in Massive Graphs. In IJCAI (pp. 568-574).

[8] Cai, S., Su, K., Luo, C., &Sattar, A. (2013). NuMVC: An efficient local search algorithm for minimum vertex cover. Journal of Artificial Intelligence Research, 46, 687-716.

[9] 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.

[10] Chen, J., Kou, L., & Cui, X. (2016). An approximation algorithm for the minimum vertex cover problem. Procedia engineering, 137, 180-185.

[11] Garey, M. R., & Johnson, D. S. (1978). ``strong"np-completeness results: Motivation, examples, and implications. Journal of the ACM (JACM), 25(3), 499-508.

[12] Chlebík, M., &Chlebíková, J. (2008). Crown reductions for the minimum weighted vertex cover problem. Discrete Applied Mathematics, 156(3), 292-312.

[13] Chen, J., &Kanj, I. A. (2005). On approximating minimum vertex cover for graphs with perfect matching. Theoretical Computer Science, 337(1-3), 305-318.

[14] Shyu, S. J., Yin, P. Y., & Lin, B. M. (2004). An ant colony optimization algorithm for the minimum weight vertex cover problem. Annals of Operations Research, 131, 283-304.

[15] Chen, J., Kanj, I. A., & Xia, G. (2006). Improved parameterized upper bounds for vertex cover. In Mathematical Foundations of Computer Science 2006: 31st International Symposium, MFCS 2006, StaráLesná, Slovakia, August 28-September 1, 2006. Proceedings 31 (pp. 238-249). Springer Berlin Heidelberg.

[16] Cygan, M., Kowalik, Ł., &Wykurz, M. (2009). Exponential-time approximation of weighted set cover. Information Processing Letters, 109(16), 957-961.

[17] Chvatal, V. (1979). A greedy heuristic for the set-covering problem. Mathematics of operations research, 4(3), 233-235.

[18] Pitt, L. B. (1985). A simple probabilistic approximation algorithm for vertex cover. Yale University, Department of Computer Science.

[19] Singh, A., & Gupta, A. K. (2006). A hybrid heuristic for the minimum weight vertex cover problem. Asia-Pacific Journal of Operational Research, 23(02), 273-285.

[20] Shyu, S. J., Yin, P. Y., & Lin, B. M. (2004). An ant colony optimization algorithm for the minimum weight vertex cover problem. Annals of Operations Research, 131, 283-304.

[21] Gilmour, S., &Dras, M. (2005, December). Understanding the pheromone system within ant colony optimization. In Australasian Joint Conference on Artificial Intelligence (pp. 786-789). Berlin, Heidelberg: Springer Berlin Heidelberg.

[22] Jovanovic, R., & Tuba, M. (2011). An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem. Applied Soft Computing, 11(8), 5360-5366.

[23] Costa, D., & Hertz, A. (1997). Ants can colour graphs. Journal of the operational research society, 48(3), 295-305.

[24] Wang, C., Zhou, Y. R., &Tu, W. P. (2007). Hybrid genetic algorithm for vertex cover problem. JisuanjiGongchengyuYingyong(Computer Engineering and Applications), 43(14), 27-29.

[25] Kotecha, K., &Gambhava, N. (2003). A Hybrid Genetic Algorithm for Minimum Vertex Cover Problem. In IICAI (pp. 904-913).

[26] Mehrabi, A. D., Mehrabi, S., &Mehrabi, A. (2009, October). A pruning based ant colony algorithm for minimum vertex cover problem. In International Conference on Evolutionary Computation (Vol. 2, pp. 281-286). SCITEPRESS.

[27] Brankovic, L., &Fernau, H. (2013). A novel parameterised approximation algorithm for minimum vertex cover. Theoretical Computer Science, 511, 85-108.

[28] Papadimitriou, C. H. (1994). Computational complexity addison-wesley reading. Massachusetts Google Scholar.

[29] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., &Protasi, M. (2012). Complexity and approximation: Combinatorial optimization problems and their approximability properties. Springer Science & Business Media.

[30] Solnon, C., & Bridge, D. (2006). An ant colony optimization meta-heuristic for subset selection problems.

[31] Lee, K., Joo, J., Yang, J., &Honavar, V. (2006, August). Experimental comparison of feature subset selection using GA and ACO algorithm. In International Conference on Advanced Data Mining and Applications (pp. 465-472). Berlin, Heidelberg: Springer Berlin Heidelberg.

[32] Chen, J., Kou, L., & Cui, X. (2016). An approximation algorithm for the minimum vertex cover problem. Procedia engineering, 137, 180-185.

[33] Asif, M., &Baig, R. (2009, October).Solving NP-complete problem using ACO algorithm; In 2009

International conference on emerging technologies (pp. 13-16). IEEE.

[34] Mandal, A.K., Sarma, P.K.D., Dehuri S. (2023). Image-based Skin Disease Detection and Classification through Bioinspired Machine Learning Approaches. International Journal on Recent and Innovation Trends in Computing and Communication. 12, 1 (Sep. 2023), 85–94. DOI: https://doi.org/10.17762/ijritcc.v12i1.7914.

[35] Mandal, A.K., Sarma, P.K.D., Dehuri S. (2023). Machine Learning Approaches and Particle Swarm Optimization Based Clustering for the Human Monkeypox Viruses: A Study. In Innovations in Intelligent Computing and Communication: First International Conference, ICIICC 2022, Bhubaneswar, Odisha, India, December 16-17, 2022, Proceedings 2023 Jan 1 (pp. 313-332). Cham: Springer International Publishing.