

Cellular Learning Automata Based Algorithm for Solving Minimum Vertex Cover Problem

Aylin Mousavian

Faculty of Computer and
Information Technology

Engineering, Qazvin Branch, Islamic
Azad University, Qazvin, Iran
a.mousavian@qiau.ac.ir

Alireza Rezvanian

Computer Engineering &
Information Technology

Department, Amirkabir University of
Technology (Tehran Polytechnic),
Tehran, Iran
a.rezvanian@aut.ac.ir

Mohammad Reza Meybodi

Computer Engineering &
Information Technology

Department, Amirkabir University of
Technology (Tehran Ploytechnic),
Tehran, Iran
mmeybodi@aut.ac.ir

Abstract— The minimum vertex cover of a given graph G is a set of vertices such that every vertex in G belongs either to the set or adjacent to vertices of the covering set. Finding the minimum vertex cover in an arbitrary graph is NP-Complete and several approximation algorithms have been proposed for solving this problem in graphs. In this paper, cellular learning automata based algorithm is proposed for solving the minimum vertex cover problem. In this algorithm each vertex as a cell is equipped with a learning automaton that has been influenced by adjacent cells and learning rule. This approach gradually reaches near optimal solution for minimum vertex cover and reduces the number of cells as covering set. Taking advantage of parallel implementation, proposed algorithm reduces running time in most of instances. The proposed algorithm is tested on DIMACS benchmark graphs and is compared with other well-known algorithms. Experimental results show that proposed algorithm uniformly performs efficiently and gives better results over other methods.

Keywords- cellular learning automata; minimum vertex cover problem; NP-Complete problem; irregular cellular learning automata

I. INTRODUCTION

This paper proposes an algorithm for minimum vertex cover problem based on cellular learning automata (CLAVC). Vertex cover problem in undirected graph $G=(V,E)$ is a subset $S \subseteq V$, such that every vertex $v \in V$ is either in S or adjacent to a vertex of S . In particular, vertex cover is set of vertices that cover all vertices in arbitrary graph [1].

Minimum vertex cover problem is NP-Complete problem [2]. All well-known exact algorithms for solving NP-Complete problems have exponential time complexity [3]. Many researchers have been working on designing approximate algorithms in order to solve the problem in reasonably short time [4]. Vertex cover problem has been employed to model many issues in various real-life-

applications, such as network flow [5], circuit design [6], transportation [7] and telecommunication [8]. Since this problem has many important practical applications, many methods have been presented to solve the problems such as evolutionary heuristic [9, 10], approximate algorithm [11]. Back and Khuri [9] have shown experimentally that a genetic algorithm performance is in running time of Cn^2 .

Another heuristic solution for vertex cover problem is greedy algorithm which is removed vertex of maximum degree repeatedly until all vertices are covered [12] but the algorithm actually performs poorly on many classes of graphs and has no fixed performance bounds [12]. A popular approximation technique removes vertices of edges that have been chosen randomly from set. In this algorithm the obtained result is always lower than twice the optimum solution, but provides near optimal solution.

In recent years several learning automata based algorithms are presented to solve graph problems and successful results have been reported in literature, such as shortest path problem [13], minimum spanning tree problem [14], minimum dominating set problem [15] and maximum clique problem [16]. Another successful method for solving vertex cover is using learning automata which find minimum number of nodes in less running time [17].

In this paper cellular learning automata is applied to solve minimum vertex cover problem. Cellular learning automata (CLA) [18], a combination of cellular automata (CA) [19] and learning automata (LA) [20], could be a powerful mathematical model for many decentralized problems and phenomena. A CLA is a CA in which a learning automaton is assigned to every cell. The learning automaton residing in a particular cell determines its action (state) on the basis of its action probability vector. Like CA there is a rule that the CLA operates in accordance with. The rule of CLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell [21, 22]. Cellular learning automata (CLA) called irregular cellular learning automata (ICLA) which removes the restriction of rectangular grid structure in

traditional CLA. This generalization is expected because there are applications which cannot be adequately modeled with rectangular grids; one category of such applications is graph of networks [23]. Our purpose is to solve minimum vertex cover problem by using irregular cellular automata. Network of cells with learning automata is created corresponding given graph, the rule of CLA and neighbor's action make the proposed algorithm able to obtain optimal solution. The rest of paper is organized as follows. In section 2 irregular learning automata is presented. The proposed algorithm based on ICLA is described in section 3. Simulation results are in section 4 and Section 5 is the conclusion.

II. LEARNING AUTOMATA

Learning automaton (LA) [17, 20] is an abstract model of finite state machine so that it can perform finite actions which are randomly chosen and evaluated by a stochastic environment and response is given as reward or penalty to LA. LA uses feedback of environment and select next action. During this process, LA learns how to choose the best action from a set of allowed actions. Figure 1 illustrates relationship between the LA and its random environment.

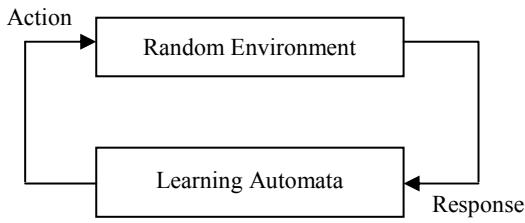


Fig. 1. The relationship between the LA and its random environment

LA with variable structure can be described with quadruple $\{\alpha, \beta, p, T\}$ where $\alpha=\{\alpha_1, \dots, \alpha_r\}$ denotes the finite set of actions and $\beta=\{\beta_1, \dots, \beta_m\}$ is the set of input values that can be taken by reinforcement signal and $p=\{p_1, \dots, p_r\}$ is probability vector of each action. $P(n+1)=T[\alpha(n), \beta(n), p(n)]$ is learning algorithm where is updated in each iteration.

Equations (1) and (2) illustrate changes in probability vector according to performance evaluation of α_i in each step. Automaton updates its action probability set based on equation (1) for favorable responses as:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1-p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (1)$$

And equation (2) for unfavorable ones:

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \left(\frac{b}{r-1} \right) + (1-b)p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (2)$$

Where $P(n)$ is the action probability vector at instant n . r is the number of actions that can be taken by the LA. a and b denote reward and penalty parameters and determine the amount of increases and decreases of the action probabilities respectively[22, 23].

III. IRREGULAR CELLULAR LEARNING AUTOMATA

The idea of Irregular Cellular Automata (ICA) was suggested in mid 80s [22, 23], but it has been receiving less attention since then. In an informal way, ICA is a configuration of points in the space with no prior restriction. Each point has a number of other points as its neighbors [23]. Irregular cellular learning automata are combination of irregular cellular automata and learning automata [23, 24]. We define ICLA as an undirected graph which, each vertex represent a cell which is equipped with learning automaton. The learning automata residing in a particular cell specify its action on action probability vector like CLA, there is a rule that the ICLA operate in accordance with [22, 23, 24]. Actions (states) of neighbors influence on particular cell to update its action probability vector. An ICLA is formally defined below.

An ICLA is a structure where $A=(G< V, E>, \Phi, A, F)$

- G is an undirected graph, V as set of vertices, E as set of edges.
- Φ is a finite set of states (actions).
- A is set of LA whereas each of them is assigned to one cell of the ICLA.
- $F: \Phi_j \rightarrow \beta$ is the local rule of ICLA in each vertex j .
- Where $\Phi_j = \{\Phi_i(i, j) \sqcup E\} + \{\Phi_j\}$ is the set of states of all neighbors of j and β is the set of values that the reinforcement signal can take. β compute influence of LA_j based on actions selected by neighboring LA. More details and information about irregular cellular learning automata obtained in [21, 23, 24].

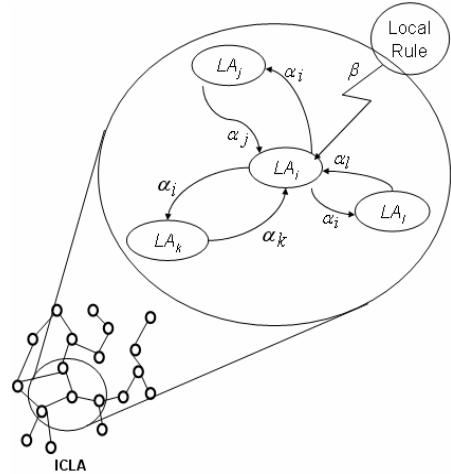


Fig. 2. Irregular Cellular learning Automata [23]

IV. PROPOSED ALGORITHM

In this section, proposed algorithm for covering vertices in graph based on irregular cellular learning automata is explained. Undirected graph vertices is considered as cells which is assigned with learning automata, since in graph structure, rectangular grid model for cellular learning automata is not used, then irregular cellular learning automata is assumed to model mentioned problem. At first iteration of algorithm, each vertex as a cell has 0.5 probability vector value.

Unlike previous solutions that the covering vertices have been selected sequentially, in this algorithm all candidate cells for covering vertices are chosen together at the same time. The Number of chosen cells at first iteration is computed based on average degree of graph and is selected randomly. Chosen cells in each iteration are placed in a separate category as cover set; all neighbors of chosen automata are added to the set as neighborhood. For all cells that are not covered by selected cells with learning automata, does as below.

In each iteration if a cell has been had neighbors with length larger than value ($1.1 * \text{Average degree}$) and has not been covered with any cell then is penalized and probability vector is updated. The value ($1.1 * \text{average degree}$) is normal value that has best result in algorithm. This value has been found after more than 1000 times test in different instances. For all covering cellular learning automata, numbers of neighbors are computed in parallel with current covering cells. If count of neighbors is larger than value ($1.1 * \text{average degree}$) and covering cells as neighbor lower than 3, current cell is rewarded, otherwise is penalized. This process is continued until the cardinality of covering set is equal to predefined threshold. Therefore CLA learns how to choose best cell in successive iterations. In next section experimental simulation is demonstrated to evaluate proposed algorithm. The probability vector value for the best covering cells is updated at each iteration in compared with the previous iteration. Therefore in successive iterations, the proposed algorithm selects intelligently covering cells by learning from neighborhood and local rules.

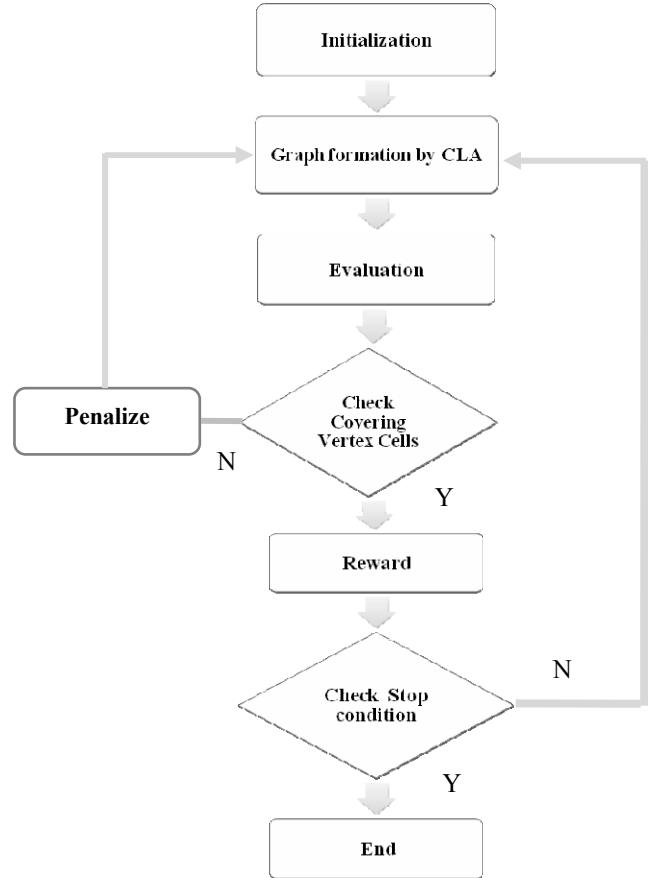


Fig. 3. Structure of proposed algorithm based on CLA for Vertex Cover problem

V. EXPERIMENTAL RESULTS

In order to evaluate the performance and behavior of proposed algorithm (CLAVC), computational experiments are performed on DIMACS benchmark instances. Results of CLAVC are compared with PLS, EWCC, EWLS [25]. Table 1 demonstrated effectiveness of proposed algorithm in compare with above mentioned algorithms. Experimental results show that this approach greatly increases the covering size in the DIMACS benchmark instances. Figures 4 to 7 as bar chart diagrams indicate that coverage of CLAVC has exceeded in compare with PLS, EWCC, EWLS. Simulation results show that the new CLAVC can yield better solutions for DIMACS benchmark graph and led to give near optimal solutions for most of the test instances. In table 1, CN indicates covering size and STP shows all steps of the algorithm which have been implemented. The results in bold indicate the best performance for an instance.

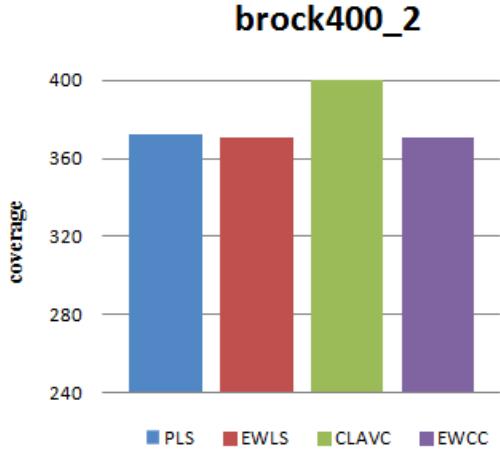


Fig. 4. Comparison of CLAVC in brock400_2 benchmark with EWCC, EWLS, PLS [25]

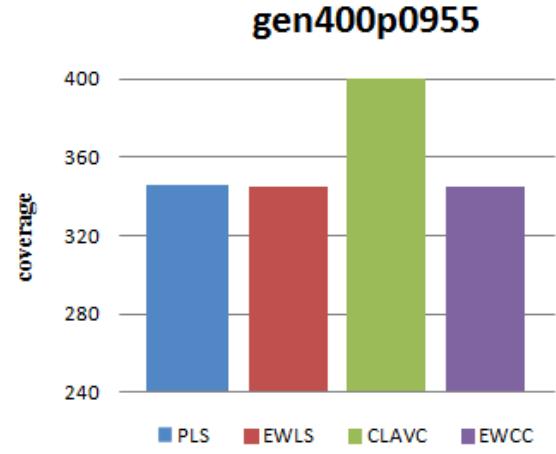


Fig. 7. Comparison of CLAVC in gen400p0955 benchmark with EWCC, EWLS, PLS [25]

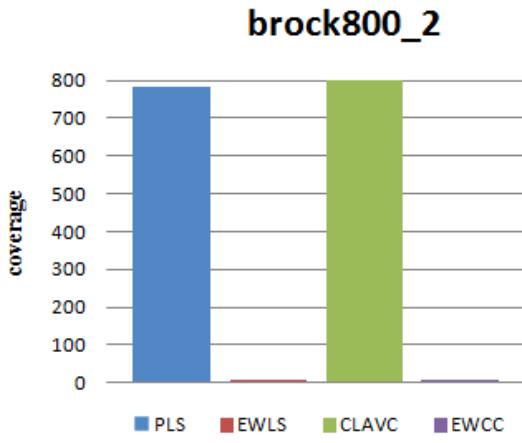


Fig. 5. Comparison of CLAVC in brock800_2 benchmark with EWCC, EWLS, PLS [25]

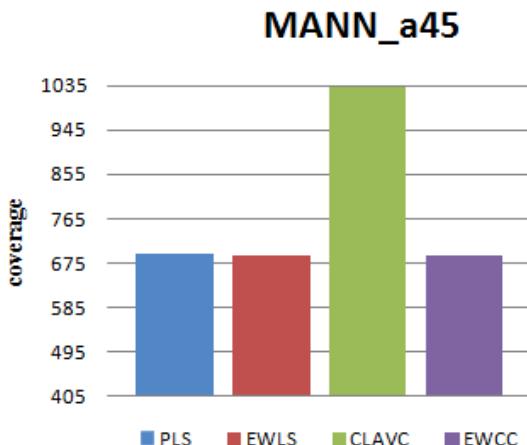


Fig. 6. Comparison of CLAVC in MANN_a45 benchmark with EWCC, EWLS, PLS [25]

TABLE I. COMPARISON OF COVERAGE AND STEPS FOR PROPOSED ALGORITHM WITH OTHER ALGORITHMS

Instances	#	EWLS		EWCC		CLAVC	
		CS	STP	CS	STP	CS	STP
Brock200_2	200	188	114476	188	117800	200	100
Brock400_2	400	371	12253257	371	185029672	400	50
Brock800_2	800	0	n/a	0	n/a	800	100
C125	125	91	70	91	73	125	50
C250	250	206	1541	206	1743	250	100
C500	500	443	115481	443	99203	500	70
gen200p0944	200	156	2484	156	1296	200	171
Gen400p0955	400	345	41906	345	29450	400	130
P_hat7002	700	656	222	656	212	700	79
MANN_a45	1035	690	124004605	690	404529167	1033	200

VI. CONCLUSION

In this paper, a new algorithm for minimum vertex cover problem is proposed using cellular learning automata. As mentioned above, most of the methods starting from a random set of optimal search, whereas in proposed algorithm in each step, vertex set is defined and response is improved using learning mechanism due to attention to neighboring state, then good result have been obtained for minimum vertex cover problem. Due to exponential complexity of minimum vertex cover problem, the algorithm is not supposed to reach optimal solution in reasonable time, but the results show that proposed algorithm optimizes solutions compared to other well-known algorithms in minimum vertex cover problem.

REFERENCES

- [1] Z. Zhang, X. Gao and W. Wu, "Algorithm for connected set cover problem and fault-tolerant connected set cover problem," *Theoretical Computer Science*, vol. 410, pp. 812-817, 2009.
- [2] F. C. Gomes, C. N. Meneses, P. M. Pardalos and G. V. Viana, "Experimental analysis of approximation algorithms for the vertex cover and set covering problems," *Computers & Operations Research*, vol. 33, pp. 3520-3534, 2006.
- [3] R. M. Karp, *Reducibility among combinational problem*, Newyork: Springer, 1972.
- [4] C. Xing, G. Zhang and M. Chen, "Research on universal network performance testing model," in *Proceeding of International Symposium of Communication and Information Technologies*, Sydney, 2007.
- [5] Y. Ni, "Minimum weight covering problem in stochastic environment," *Information Science*, vol. 214, pp. 91-104, 2012.
- [6] W. T. Shiue, "Novel state minimization and state assignment in finite state machine design for low- power portable devices," *Integration the VLSI Journal*, vol. 38, pp. 549-570, 2005.
- [7] Y. Guo, R. Rodrigues and Z. Y, "Carrier assignment models in Transportation procurement," *Operational Research Society*, vol. 57, pp. 1472-1481, 2006.
- [8] A. Wiese and E. Kranakis, "Local PTAS for independent set and vertex cover in location aware unit disk graphs," *Distributed Computing in Sensor Systems*, vol. 5067, pp. 415-431, 2008.
- [9] S. Khuri and T. Back, "An evolutionary heuristic for the minimum vertex cover problem," in *Proceeding of 18th Annual Conference on Artifical Intelligence*, 1994.
- [10] X. Xu and M. Jun, "An efficient simulated annealing algorithm for the minimum vertex cover problem," *Neurocomputing*, vol. 69, pp. 913-916, 2006.
- [11] W. Pullan, "Optimisation of unweighted/weighted maximum independent sets and minimum vertex covers," *Discrete Optimization*, vol. 6, pp. 214-219, 2009.
- [12] R. B. Yehuda and S. Even, "A linear time approximation algorithm for weighted vertex cover problem," *Journal of Algorithm*, vol. II, pp. 198-203, 1981.
- [13] H. Beigy and M. R. Meybodi, "Utilizing distributed learning automata to solve stochastic shortest path problem," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 14, pp. 591-617, 2006.
- [14] J. A. Torkestani and M. R. Meybodi, "Learning automata based algorithms for solving stochastic minimum spanning tree problem," *Journal of Applied Soft Computing*, vol. 11, pp. 4064-4077, 2011.
- [15] J. A. Torkestani and M. R. Meybodi, "Finding minimum weight connected dominating set in stochastic graph based on learning automata," *Information Sciences*, pp. 57-77, 2012.
- [16] M. Soleimani-Pouri and A. Rezvanian, "Solving maximum clique problem in stochastic graphs using learning automata," in *Proceedings of 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, 2012.
- [17] A. Mousavian, A. Rezvanian and M. R. Meybodi, "Solving minimum vertex cover problem using learning automata," in *Proceeding in 13th Iranian Conference on Fuzzy System*, Qazvin, 2013.
- [18] H. Beigy and M. R. Meybodi, "A mathematical framework for cellular learning automata," *Advances in Complex Systems*, vol. 7, pp. 295-319, 2004.
- [19] E. Fredkin, "Digital machine: A informational process based on reversible cellular automata," *Physica*, vol. D45, pp. 245-270, 1990.
- [20] A. Rezvanian, M. Rahmati, M. R. Meybodi, "Sampling from complex networks using distributed learning automata", *Physica A: Statistical Mechanics and its Applications*, vol. 396, pp. 224-234, 2014.
- [21] M. R. Meybodi, H. Beigy and M. Taherkhani, "Cellular Learning Automata and its Applications," *Journal of Science Tech*, pp. 54-77, 2004.
- [22] D. Stevens, "Integration of an irregular cellular automata approach and geographic information systems for high-resolution modeling of urban growth," in *Msc. Thesis*, Department of Geography,University of Toronto, 2003.
- [23] M. Esnaashari and M. R. meybodi, "Irregular cellular learning automata and its application to clustiring in sensor network," in *Proceedings of 15th Conference on Electrical Engineering (15th ICEE)*, Tehran, 2007.
- [24] A. E. Eraghi, J. A. Torkestani, M. R. Meybodi and A. H. Fathy, "Cellular learning automata-based channel assignment algorithms for wireless mobile Ad Hoc networks," in *Proceedings of 2009 International Conference on Machine Learning and Computing*, Perth, 2009.
- [25] S. Cai, K. Su and A. Sattar, "Local search with edge weighting and configuration checking heuristics for minimum vertex cover," *Artificial Intelligence*, vol. 175, p. 1672–1696, 2011.