International Journal of Research Publication and Reviews
Journal homepage: www.ijrpr.com ISSN 2582-7421

Analysis of Recent Trends in Solving *NP* Problems with New Research Directions Using Evolutionary Methods

Balakrishnan S[1], Dr.Tamilarasi Suresh[2], Raja Marappan[3*]

[1] Research scholar, Dr MGR Educational and Research Institute, Maduravoyal, Chennai - 600 095,
e-mail: balakrishnanit@drmgrdu.ac.in, sbalasharva@gmail.com
[2] Professor, Department of Information Technology, St. Peter's Institute of Higher Education and Research, Avadi, Chennai – 54,
email: tamilarasisu@gmail.com
[3] Senior Assistant Professor, School of Computing, SASTRA Deemed University, Thanjavur-613401
[*] Corresponding Author Email: raja_csmath@cse.sastra.edu

ABSTRACT

Nowadays, there are different *NP* problems that are being applied to various applications. These problems are not solved in polynomial time. A wide variety of soft computing and approximation strategies are applied in solving *NP* problems to find near-optimal solutions. The recent trends in solving some of the classical *NP*problems are analyzed in this paper. The design of effective algorithms which produces a better solution with minimal computational time and space complexity is required in solving these complex real-world problems. The recent approximation and soft computing strategies applied in solving these *NP* problems are analyzed with future research directions.

Keywords : NP, maximum clique, N-queen, vertex cover, traveling salesman,0-1 knapsack, graph coloring, genetic algorithm

INTRODUCTION

The solutions for a lot of real-world problems are not obtained in polynomial time. An algorithm executes in a polynomial time if its complexity is in $\Theta(n^k)$ for an input size n with $k > 0$, an integer. The non-deterministic polynomial (*NP*) class problems are solved on a non-deterministic Turing machine in $\Theta(n^k)$ complexity [1]. The class *P*solves the problems in a polynomial complexity using deterministic Turing machine. The theoretical foundations of *NP*-completeness are presented in [2]. *NP* problem is converted into the satisfiability problem in $\Theta(n^k)$ complexity. A problem *R* is in *NP-complete* if *R*is in *NP*, and some other problem from *NP* is transferred into *R* in $\Theta(n^k)$ complexity. *NP*-hardness represents a characteristic "at least as hard as the hardest problems in *NP*".

Section 2 focuses on some of the classical *NP* problems and applications. Section 3 focuses on the analysis of recent methods applied in solving some of the *NP* problems. The critiques in some of the recent well known evolutionary methods in solving *NP* problems are analyzed in section 4 with some of the new research directions. Conclusions are highlighted in section 5.

NP PROBLEMS

Let $G = (V, E)$ be the undirected graph with n vertices, where $V(G)$ &$E(G)$ represents the vertex & edge sets respectively.

2.1 Maximum Clique

Clique in G is defined as a complete subgraph of *G*. Then maximum clique finds the maximum complete subgraph of *G*. This *NP*-hard instance is applied in register allocation and job scheduling applications.

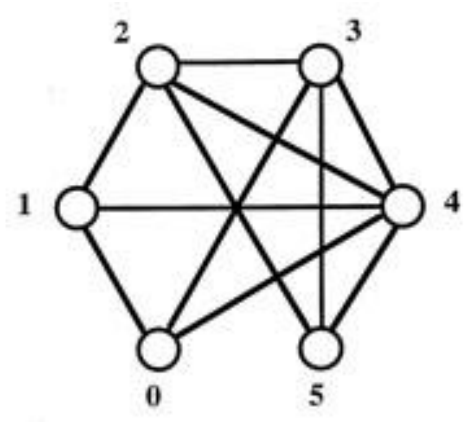


Fig 1: G_1

The maximal clique is the largest subset of V(G) in which every v in V(G) is incident to every other vertex in the subset. The count of total cliques becomes double when a new vertex is added every time. Thus the problem becomes exponential growth. For the given graph shown in figure 1, the vertex set covered in the maximum clique is {2, 3, 4, 5} and hence the maximum clique size is 4.

2.2 N-Queen Problem

This *NP*-complete problemassigns *N* queens on an $N \times N$ chessboard in which two queens should not be assigned on the same diagonal, row or column is the *N*-Queen problem [3]. One solution for $N = 8$ is shown in figure 2.

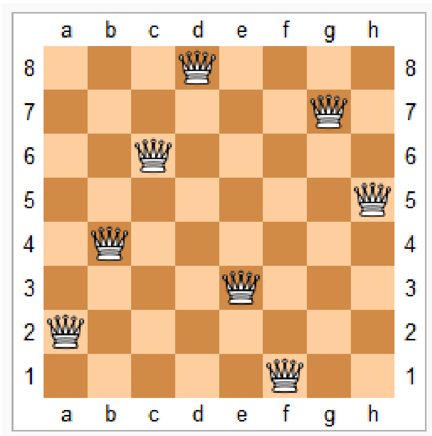


Fig 2: One solution to 8-Queens problem

2.3 Vertex Cover Problem

For G, the vertex cover finds a vertex set $S\in V$(G) in which each edge $e\in E$(G) in *G* has at least one end vertex in *S* [4]. The partial vertex cover problemcovers at least *k* edges of *E*(*G*)with a subset *S'*in *V*(G). The minimum partial vertex cover determines a minimum cardinality subset of vertices $S'\in V$(G) in which at least *k* edges in *G* cover the vertices in $S'\in V$(G). This problem is applied in different optimization and design problems.

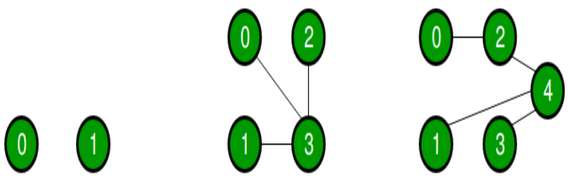


Fig 3: Graphs - G_1, G_2, G_3

For the graphs shown in figure 3, the minimum vertex cover is given by $G_1 = \{\emptyset\}$, $G_2 = \{3\}$, and $G_3 = \{2, 4\}$ or $\{0, 4\}$.

2.4 Traveling Salesman Problem (TSP)

TSP finds the shortest distance for a traveling salesman to traverse *n* cities, and it returns again to the same starting city. The total cost of the tour should be minimized.

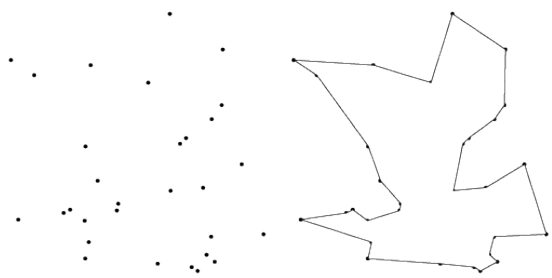


Fig 4: TSP – Graph G_4 with solution

Figure 4 shows the solution for the graph G_4 of the TSP.

$$\begin{pmatrix} \infty & 20 & 17 & 16 \\ 18 & \infty & 25 & 12 \\ 15 & 10 & \infty & 22 \\ 11 & 26 & 19 & \infty \end{pmatrix}$$

Consider the above edge length matrix for four cities directed graph. The optimal tour length is 50 and the optimal tour is given by City (vertex) 1⇨ City (vertex) 3: City (vertex) 3⇨ City (vertex) 2: City (vertex) 2⇨ City (vertex) 4: City (vertex) 4⇨ City (vertex) 1.

2.5 0-1 Knapsack Problem

Knapsack problem, *NP*-complete combinatorial optimization, is applied in material cutting, prediction of demands, cargo loading [6]. This problem maximizes the value of objects under the constraints of backpack capacity. In this decision problem, the knapsack capacity is fixed. Every object is unique and has profit and weight attributes. Given there are *n* unique objects with weight w_i and profit $p_{i,}$ the total weight of all objects loaded with the knapsack cannot exceed its total capacity *c*.

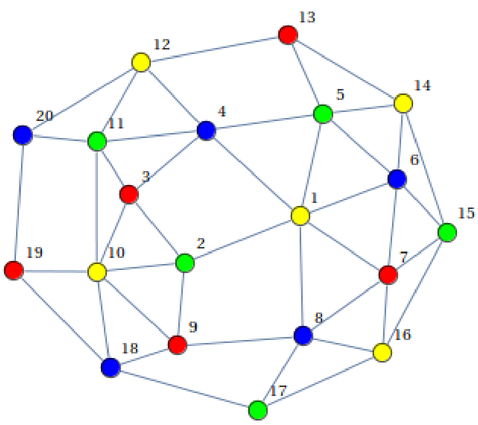
If j^{th} object is placed into the knapsack then x_j =1 and the profit earned is p_j. If j^{th} object is not placed into the knapsack then x_j =0 and the profit earned is 0.The optimization problem is to fill a knapsack that maximizes the total profit such that x_j=0 or 1, $x_j$and $w_j$are ≥ 0 for *j*=1, 2, …n.

For example, the optimal solution to the knapsack instance for *n*=3 with (p_1, p_2, p_3) = (12, 13, 15), (w_1, w_2, w_3) = (7, 5, 12) and *c*= 12 is given byx_1=1, x_2=1, x_3=0. The maximum profit is 25.

2.6 Graph Coloring Problem

For the simple graph *G*, the adjacency matrix of *G*is *A*(*G*), an *n* × *n* symmetric matrix with *A*(*k*,*j*) = 1 (1 ≤ *k*,*j* ≤ *n*) if (v_j, v_k) ϵ E(G); and A(*j*, *k*) = 0 (1 ≤ *k*, *j* ≤ *n*) if (v_j, v_k) is not in E(G).

χ(*G*) is the minimal color integer of *G* that defines the least number of integers needed for *V*(*G*) in such a way that no two adjoining vertices v_j and v_k are assigned the same integer [7-18]. It is used in different engineering applications such as channel assignment, allocation of registers, resource utilization, and scheduling. The color assignments of graph G_5 are highlighted and are shown in figure 5. χ(G_5) = 4.



s

Fig 5: Graph G_5 with color assignments

3 ANALYSIS OF *NP* PROBLEMS

This section focuses on the analysis of recent methods applied in solving some of the *NP* problems.

3.1 Maximum Clique Problem

The artificial intelligence and meta-heuristic algorithms are applied to find the solution of maximum clique [1]. The computational time needed to determine the maximum clique depends on the size of a clique. The methods are tested on randomly generated graphs to find an incomplete solution.The branch and bound with a pruning strategy is applied to some DIMACS benchmark and random graphs [20]. The vertex order affects output performance measurements. The challenging clique benchmarks to be solved are shown in [19]. The performance of these methods should be improved further to reduce the complexity in the online training process so that a better near-optimal solution can be achieved. The heuristic with different membrane operators is developed in solving clique problem [24]. The experimentation for more large data sets in finding the maximum cliques are further expected.

3.2 *N*-Queen Problem

There are several variations in *N*-Queen such as excluded diagonals and the blocked *N*-Queen problems that are shown as *NP*-complete [3]. The generators are presented for hard random instances to find the solution. It is expected to determine a straightforward generator for random instances of hard categories so that the computations of reductions can be eliminated.Constraint programming with propagation and backtracking searches is applied in solving the N-Queens problem [21]. *N*-Queens computation is both #P-Complete and *NP*-Complete. Counting the number of solutions using the approximation methods is computationally expensive for $n \geq 20$. *N*-queen problem has fewer solutions for $n = 6$ than $n = 5$. The asymptotic behavior of solving the *N*-Queen problem is not yet analyzed and currently no known expression is obtained to count the exact solutions. The count of possible exact solutions for $n = 27$ exceeds 2.34×10^{17}.

3.3 Vertex Cover Problem

The memetic algorithm, a population-based meta-heuristic evolutionary framework embedded with local search methods, exploration and exploitation of an evolutionary process is applied in solving the minimum partial vertex cover problem [4]. The powerful backbone recombination and adaptive mutation are applied to skip the local optimum. An approximation with the shortest path calculation is designed to solve the GCP [22]. The new approximation method still is expected for solving weighted graphs.The quality of solutions for the different benchmark instances can further be improved using evolutionary algorithms so that it can be processedin real-world applications.

3.4 Traveling Salesman Problem

The Physarum-inspired computational model with hill climbing strategies is applied in solving multi-objective TSP. Exact techniques such as linear programming, branch and bound, heuristic, branch and cut, and approximation methods are applied in solving TSP [30].Even evolutionary algorithms are applied in solving optimization problems, but it suffers insufficient population diversity, premature convergence, and non-uniform distribution of solutions. The efficiency of the model can be further improved by reducing the complexity of the most critical computational operations. New insights should be gained and developed using the evolutionary model in balancing the heterogeneous features of multilayer networks to fine-tune the solution quality in community detection [5].

Clustering method is applied to maximize the performance of the evolutionary method in solving symmetric TSP [29]. This method applied clustering so that the cities are partitioned into minimal number of partitions and then every individual cluster is solved using GA. It computes the shortest Hamiltonian tour to visit every city and it consecutively visits the cities of each cluster. Minimum path lengths are applied using GA with cycle crossover and mutation such as reverse sequence, swap and slide. Cluster size, the number of clusters is not determined. New chromosome from the cluster paths are selected without any heuristics and applied for smaller problem instances. Population size N is set in between 160 and 240 which may increase computation cost for symmetric problem instances.

3.5 0-1 Knapsack Problem

The accurate and approximate algorithms based on dynamic programming, genetic algorithm, ant colony optimization, differential evolution, membrane computing is designed in solving knapsack instances [6]. The solutions obtained by the approximation algorithms cannot completely solve some practical applications. The ant colony system with pheromone matrix optimization is developed to solve TSP and 0-1 knapsack problems [26].

3.6 Graph Coloring Problem

There are different approximations and evolutionary approaches applied to solve GCP [8-18]. These approaches reduce the minimum color used for some of the benchmark instances. For some of the benchmark instances, the minimal color has not yet obtained using the approximation and soft computing strategies. The challenging graph coloring instances are shown in [23]. The memetic method to solve GCP is developed to solve large benchmarks [25]. Furthermore, it is necessary to explore problem-based additional features so that effective operators and search strategies will be designed for large scale benchmark instances. Since GCP occupies the vast solution space, it is still necessary to design effective approximation and evolutionary methods to find the solution for large benchmark and random instances.

4 CRITIQUES IN THE EVOLUTIONARY METHODS AND NEW RESEARCH DIRECTIONS

The critiques in some of the recent well known evolutionary methods in solving *NP* problems are analyzed in this section. New research directions in solving *NP* problems using the evolutionary methods are also focused.

The flow diagram of the general evolutionary technique to solve *NP* problems is depicted in figure 6.Several mutations are defined in solving some of the *NP*-hard problems and this research require further to obtain new mutations by combining existing operators in solving other *NP* problems [27].

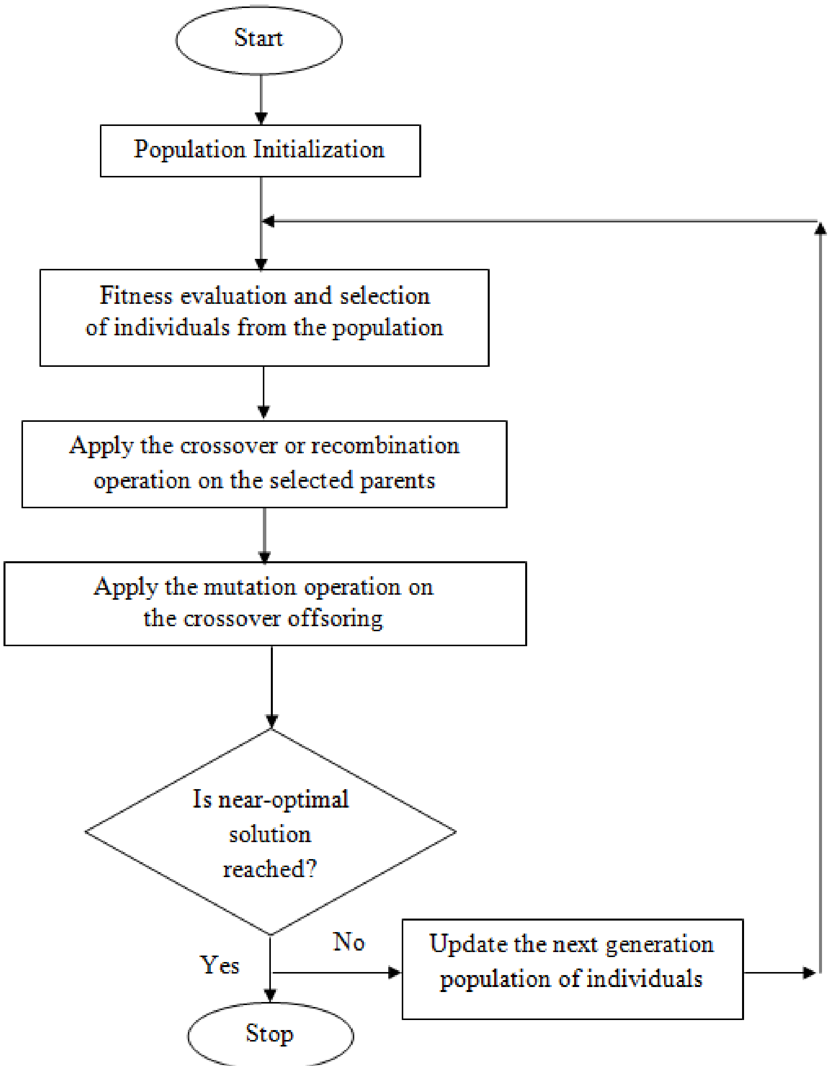


Fig 6: Flowchart of the general evolutionary method to solve *NP* problems

Some of the existing methods require a large population size which increases the complexity of the evolutionary model. These methods require more exploration of search space and hence require more space requirements. Some of the operators result in premature convergence over a smaller number of generations. These operators result in a near-optimal solution which is significant from better near-optimal solutions [28]. Some of the evolutionary algorithms require a large number of runs, mutations, generations, and crossovers that are increasing the computational complexities of real-world problems. It has also been noticed that some methods produce a lower proportion of successful runs of the evolutionary algorithm [7, 31].

To overcome the limitations of these existing methods, the following new research directions are focused on solving these classical *NP* problems using evolutionary methods:

1. Evolutionary methods should maintain sufficient population diversity over the generations.

2. New strategies should be devised to keep away from the premature convergence of the optimization problem.
3. During the progress of an evolutionary algorithm, good individuals in the population are to be maintained.
4. New operators and search strategies should be designed and to be chosen rightly to obtain multiple solutions.
5. During the generations of the algorithm, the distribution of the solutions should be enhanced.
6. The imbalance between quick convergence and diversity preservation should be analyzed mathematically.
7. A new nature-inspired evolutionary model is expected to reduce the exploration space of the optimization problem.
8. The elitism strategies and selection of individuals should be chosen based on the parameters of the optimization problem.
9. For multi-objective problems, different problem parameters should be considered in evaluating the general algorithm performance.
10. The average generations, crossovers, and mutations are expected to smaller todecrease the computational complexity of the optimization.
11. The number of successful runs is expected to a high number.
12. The usage of additional memory resources should be minimized.
13. Setting a smaller size of population is expected to achieve a sufficient convergence rate.
14. Quick stochastic convergence is expected.
15. New advanced strategies are expected to embed with evolutionary operators in obtaining better near-optimal solutions.

5 CONCLUSIONS

Some of the classical *NP* problems and applications are discussed in this research. The recent methods applied in solving some of the classical *NP* problems are analyzed. Some of the existing methods require a large population size which increases the complexity of the evolutionary algorithm. These methods require more exploration of search space and hence require more space complexity. Some of the operators result in premature convergence over a smaller number of generations. These operators result in a near-optimal solution which is significant from better near-optimal solutions. Some of the evolutionary algorithms require a large number of runs, mutations, generations, and crossovers that are further increasing the computational complexities of real-world problems. It has also been noticed that some methods produce a lower proportion of successful runs of the evolutionary algorithm. To overcome these critiques in these existing methods, the new research directions are focused on solving these classical *NP* problems using evolutionary methods.

Conflict of Interests:

The authors of this paper declare that there is no conflict of interests regarding the publication of this paper.

REFERENCES

[1] Deniss Kumalander, Applying AI and Incomplete Solution Principles to Solve NP-hard Problems in the Real-Time Systems. Proceedings of the 10th WSEAS International Conference on Computers, Vouliagmeni, Athens, Greece, July 13-15, 2006 (pp807-811)

[2] S.A. Cook, The complexity of theorem proving procedures, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, 1971, pp. 151-158

[3] Ian P. Gent, Christopher Jefferson, Peter Nightingale: Complexity of n-Queens Completion. Journal of Artificial Intelligence Research 59 (2017) 815 – 848

[4] Yupeng Zhou; Changze Qiu; Yiyuan Wang; Mingjie Fan; Minghao Yin: An Improved Memetic Algorithm for the Partial Vertex Cover Problem. IEEE Access. 2019

[5] Xuejiao Chen; Yuxin Liu; Xianghua Li; Zhen Wang; Songxin Wang; Chao Gao: A New Evolutionary Multiobjective Model for Traveling Salesman Problem. IEEE Access. 2019

[6] Lian Ye; Jinhang Zheng; Ping Guo; Mario J. Pérez-Jiménez: Solving the 0-1 Knapsack Problem by Using Tissue P System With Cell Division. IEEE Access 2019.

[7] Marappan, R.; Sethumadhavan, G.:SolutiontoGraphColoringusingGeneticandTabuSearchProcedures. *Arab. J. Sci. Eng.* 2018,43, 525–542.

[8] Marappan, R.; Sethumadhavan, G.: A new genetic algorithm for graph coloring. 5th International Conference on Computational Intelligence, Modelling and Simulation 49-54 (2013)

[9] Sethumadhavan, G.; Marappan, R.: A Genetic Algorithm for Graph Coloring using Single Parent Conflict Gene Crossover and Mutation with Conflict Gene Removal Procedure. IEEE International Conference on Computational Intelligence and Computing Research 350-355 (2013)

[10] Marappan, R.; Sethumadhavan, G.: Solution to Graph Coloring Problem using Evolutionary Optimization through Symmetry-Breaking Approach. International Journal of Applied Engineering Research 26573-26580 (2015)

[11] Marappan, R.; Sethumadhavan, G.: Solving Graph Coloring Problem for Large Graphs. Global Journal of Pure and Applied Mathematics 2487-2495 (2015)

[12] Marappan, R.; Sethumadhavan, G.: Solution to Graph Coloring Problem using Divide and Conquer based Genetic Method. International Conference on Information Communication and Embedded System (ICICES 2016)

[13] Raja Marappan; Gopalakrishnan Sethumadhavan: Solution to Graph Coloring Problem Using Heuristics and Recursive Backtracking. International Journal of Applied Engineering Research 10 (10) 25939-25944 (2015)

[14] Marappan, R.; Sethumadhavan, G.; R.K. Srihari: New approximation algorithms for solving graph coloring problem - An

experimental approach. Elsevier, Perspectives in Science 8, 384 - 387 (2016)
[15] Marappan, R.; Sethumadhavan, G.; Harimoorty U: Solving Channel Allocation Problem using New Genetic Operators - An Experimental Approach. Elsevier, Perspectives in Science 8, 409-411 (2016)
[16] Marappan, R.; Sethumadhavan, G.: Divide and Conquer based Genetic Method for Solving Channel Allocation. ICICES 2016 (2016)
[17] Marappan, R.; Sethumadhavan, G.: Solving Fixed Channel Allocation using Hybrid Evolutionary Method. ICAET, MATEC Web of Conferences 57, 02015 (2016)
[18] Marappan, R.; Sethumadhavan, G.: Solving Channel Allocation Problem using New Genetic Algorithm with Clique Partitioning Method. IEEE ICCIC 2016.
[19] Challenging clique benchmark graphs https://turing.cs.hbg.psu.edu/txn131/clique.html
[20] Patric R. J. Ostergard: A fast algorithm for the maximum clique problem. Discrete Applied Mathematics 120 (2002) 197–207
[21] Ian P. Gent, Christopher Jefferson, Peter Nightingale: Complexity of n-Queens Completion (Extended Abstract). Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)
[22] Jingrong Chen; Lei Kou, Xiaochuan Cui: An Approximation Algorithm for the Minimum Vertex Cover Problem. Procedia Engineering 137 (2016) 180 – 185.
[23] Graph coloring instances: https://turing.cs.hbg.psu.edu/txn131/graphcoloring.html
[24] Ping Guo; Xuekun Wang; Y*i* Zeng; Haizhu Chen: MEAMCP: A Membrane Evolutionary Algorithm for Solving Maximum Clique Problem. IEEE Access. 2019
[25] Yan Jin; Jin-Kao Hao: Solving the Latin Square Completion Problem by Memetic Graph Coloring. IEEE Transactions on Evolutionary Computation, VOL. 23, NO. 6, DECEMBER 2019
[26] Yuxin Liu; Chao Gao; Zili Zhang; Yuxiao Lu; Shi Chen; Mingxin Liang; Li Tao: Solving *NP*-Hard Problems with Physarum-Based Ant Colony System. IEEE/ACM transactions on computational biology and bioinformatics, VOL. 14, NO. 1, January/February 2017
[27] Basima Hani Hasan; Moutaz Saleh Mustafa: Comparative Study of Mutation Operators on the Behavior of Genetic Algorithms Applied to Non-deterministic Polynomial (*NP*) Problems. Second International Conference on Intelligent Systems, Modelling and Simulation 2011
[28] Marappan, R.; Sethumadhavan, G.: Complexity Analysis and Stochastic Convergence of Some Well-known Evolutionary Operators for Solving Graph Coloring Problem. Mathematics 2020, 8(3), 303; https://doi.org/10.3390/math8030303
[29] Ahmad Fouad El-Samak; Wesam Ashour: Optimization of Traveling Salesman Problem using Affinity Propagation Clustering and Genetic Algorithm. JAISCR, 2015, Vol. 5, No. 4, pp. 239-245
[30] Tomoko Sakiyama; Ikuo Arizono:Coordination of Pheromone Deposition Might Solve Time-Constrained Travelling Salesman Problem. Hindawi, Complexity, Volume 2018, Article ID 6498218, 5 pages. https://doi.org/10.1155/2018/6498218
[31] Marappan, R.; Sethumadhavan, G. Complexity analysis and stochastic convergence of some well-known evolutionary operators for solving graph coloring problem. Mathematics **2020**, *8*, 303. doi: https://doi.org/10.3390/math8030303.