

New Adaptive Parallel Evolutionary Approach for Graph Coloring and Channel Allocation

S. Balakrishnan^{*1}, Tamarasi Suresh²

¹Research Scholar, Department of Computer Science and Engineering, St. Peter's Institute of Higher Education and Research, Tamilnadu, India.

Mail Id: sbalasharva@gmail.com

²Professor & Head, Department of Information Technology, St. Peter's Institute of Higher Education and Research, Tamilnadu, India

Mail Id: tamarasisu@gmail.com

^{*}Corresponding Author

Research Scholar, Department of Computer Science and Engineering, St. Peter's Institute of Higher Education and Research, Tamilnadu, India.

Mail Id: sbalasharva@gmail.com

ABSTRACT

Graph coloring is a Nondeterministic Polynomial-time (NP)-hard problem with wide applications; existing evolutionary methods suffer from fixed operators, premature convergence, and prolonged computation on large graphs. This research aims to minimize the number of colors and conflicting edges while reducing generations and runtime for large-scale graph coloring and channel allocation. Research proposes a decentralized island-based memetic framework in which independent-set extraction reduces the original graph, while a clique-split strategy lowers interference in channel allocation, and greedy initialization seeds diverse subpopulations. This framework evolves subpopulations concurrently on multiple processors, by exchanging elite individuals through periodic migration to preserve diversity. Adaptive probability matching selects crossover and mutation operators according to conflict-reduction success, and tabu search refines offspring. A fixed-k strategy iteratively decrements k after each conflict-free coloring. In graph coloring, vertices are partitioned into minimal independent color classes; in channel allocation, wireless nodes form a conflict graph in which colors represent frequencies, and cliques determine the minimal interference-free channel requirement. This approach combines class-preserving crossover, conflict-directed mutation, and statistics-driven operator selection, thereby reducing the search space while sustaining exploration across islands and accelerating convergence toward feasible colorings. Experiments on Discrete Mathematics and Theoretical Computer Science (DIMACS) graph-coloring benchmarks and wireless Channel Assignment Problem (CAP) instances achieve competitive color counts of 5–84, computation times of 0.09-15.72 s, and success rates of 78-96%, while demonstrating effective conflict minimization and convergence across varying allocation scenarios in CAP. Overall, it provides a scalable and efficient solution for large-scale graph coloring and channel allocation, also extensible to other engineering applications.

Keywords: Graph Coloring, Channel Allocation, Evolutionary Algorithm, Parallel Island Model, Adaptive Operator Selection, Combinatorial Optimization.

1. INTRODUCTION

Graph Coloring is a basic combinatorial optimization problem used in many areas including wireless communications, scheduling, frequency allocation, and resource management, where the goal is to color the vertices of a graph such that no two adjacent vertices have the same color, thereby satisfying essential constraints efficiently [1]. Channel allocation is an important problem in today's wireless communications because there is limited frequency bandwidth to be used for multiple users and communication devices. Channel allocation reduce interference and also make efficient use of the limited bandwidth, and ensures stable communication within a growingly complicated environment [2]. Classic graph coloring algorithms like greedy graph coloring and optimization have definite solutions for simple problems, their efficiency is heavily compromised by complicated graphs and the requirement to solve large channel allocation problems within a timely manner [3]. Evolutionary algorithms make better alternate techniques which helps to use the searching of several candidates' solutions through selection, crossover, and mutation techniques. These techniques help in examining various colorings and prevent getting into local optima to solve allocation problems [4]. Despite these advantages, evolutionary algorithms have often been linked with slow convergence speed, expensive computation cost, and lack of adaptability when it comes to solving large graph coloring problems. The use of sequential population update and

rigid search parameters hinders exploration capability, particularly for dynamic graphs [5]. However, parallel evolutionary computation addresses these limitations by splitting the process of evaluating and evolving the candidates among multiple processors. Each search process is able to investigate various regions within the solution space simultaneously, which prove to be more efficient in solving complex graph coloring problems [6]. Moreover, adaptive optimization complements the evolutionary paradigm by adapting the parameters according to the diversity, progress, and violation of the population. Adaptation of mutation probability, selection, and search intensity results in more optimized process according to changing problem condition [7]. Adaptive methods incorporated with parallel evolutionary search can be applied to make more efficient processes of graph coloring and channel allocation. It helps to control the process of generation of solutions, as well as interference constraints and color choice, while being efficient in computation [8]. Graph coloring finds many applications in wireless communications, scheduling, and frequency allocation. But complicated structures of the graph, restrictions due to interference and computational overhead limit the efficiency of optimization performance [9]. Limitations of previous research on graph coloring and Channel Assignment Problem (CAP) include the inability to reduce the number of channel conflicts, decrease the number of colors used, and reach faster convergence. The conventional approach has proven to be inflexible, computationally expensive, and scalable [10]. The aim of this research is to develop an adaptive parallel evolutionary approach that minimizes colors and conflicting edges in graph coloring and wireless channel allocation while improving computational efficiency, and scalability.

2. RELATED WORKS

Related research reviewed about improving graph coloring, minimizing colors, accelerating convergence, and reducing wireless interference using several algorithms which achieved promising results but retained with some limitations. Designed a highly efficient and robust graph coloring algorithm applicable to different kinds of graphs in [11] which used Malatya Vertex Coloring Algorithm that calculates the importance of each vertex via Malatya Centrality and colorizes nodes depending on colors of their neighbors. The method proved to be more efficient and robust than Welsh-Powell, Greedy Coloring, Degree of Saturation (DSatur), and RLF on benchmark, social, and random graphs, but efficiency and optimality depend on the structure of the graph.

Comparison between two algorithms, such as Sequential Coloring Algorithm (SCA) and Welsh-Powell Algorithms (WPA) were evaluated in [12] using graphs of various sizes by comparing solution quality and average processing time for four trials each. WPA generated better solutions more often when the number of nodes was greater than 1,000; however, the evaluation was limited to two heuristics and the graphs used. Research [13] was developed for designing a particle swarm optimization algorithm that would minimize the search space and number of generations required for graph coloring. Techniques like divide-and-conquer decomposition, neighborhood search, turbulence modifications, repeated fitness evaluation, and lazy-particle search were used. The technique performed better in terms of near-optimal solution finding compared to state-of-the-art approaches on selected benchmark graphs.

For enhancing graph coloring performance and computational efficiency in complex graphs, research [14] recommended a Minimal Cost Graph Neural Networks (MCGNN), which use minimal cost algorithms alongside graph neural networks for structure analysis and feature extraction of nodes. Numerical experiments have demonstrated superior performance in terms of graph coloring and computation efficiency compared to existing methods but information about the datasets and benchmarks is unknown.

Optimization of channel assignment, minimize interference, and increase energy efficiency and reliability in the healthcare sector using wireless sensor networks was done in [15]. The greedy algorithm and the Recursive Largest First (RLF) heuristic algorithms have been employed in conjunction with simulated annealing and genetic algorithms to adaptively allocate resources in a local fashion. The hybrid algorithm that used the genetic algorithm turned out to be most effective in interference minimization and data reliability. Research [16] increased throughput and energy efficiency in heterogeneous wireless sensor networks using multiple channels allocation. A two-phase genetic algorithm was used for super node spanning tree generation and channel assignment. The second phase consisted of cluster head identification and channel allocation to normal sensors. Results showed a decrease of 21.6% in total energy consumption and 48.3% in energy dissipation per transmitted bit as opposed to earlier methods.

For the enhancement of device-to-device communication in Wireless Personal Area Network (WPAN) using effective channel assignment and interference mitigation. The suggested Load Based Dynamic Channel Allocation (LB-DCA) model in [17] was based on distributed coordination, cell splitting, interference calculation, and hop- and load-balancing of channels. The LB-DCA model produced very high Critical Success Index (CSI),

Matthews Correlation Coefficient (MCC), Fowlkes Mallows index (FMI), and bookmaker informedness (BMI), in addition to improvements in spectral efficiency, scalability, signaling quality, power control, and energy efficiency. It does not state any benchmarking experiments or network conditions.

Research gap: Despite promising results, existing approaches exhibit several limitations, including graph-structure-dependent performance [11], restricted heuristic comparisons [12], benchmark-specific optimization [13], insufficient dataset transparency [14], and the absence of validation [15]. Wireless \channel allocation methods [16-18] demonstrate improvements in interference, energy efficiency, and throughput but lack comprehensive benchmarking, standardized network conditions, or adaptive parallel optimization for scalable, conflict-free channel assignment. The proposed adaptive parallel evolutionary approach overcomes the limitations of conventional graph-coloring and channel-allocation methods by combining graph reduction, parallel evolution using islands, operator adaptation, elite migration, and tabu search to reduce the number of colors and interferences while enhancing performance and scalability.

3. PROPOSED MODEL

The model presents graph coloring and channel allocation formulations, graph reduction through independent-set extraction and clique-based lower bounds, and greedy initial coloring. It details adaptive probability matching for crossover and mutation selection, parallel island-based evolutionary search, elite migration, and tabu search for conflict reduction. Finally, fixed- k optimization and channel allocation constraints guide the search toward feasible solutions.

The Adaptive Parallel Evolutionary Algorithm represents the problem of graph coloring as an optimization process, which seeks to use the least possible number of colors without conflicts between the vertices. The graph $G = (V, E)$ represents vertices and edges, while the coloring function assigns a color to each vertex. The conflict function $F(X)$ evaluates adjacent vertices sharing identical colors. A solution is considered valid when $F(X) = 0$, guiding evolutionary optimization toward efficient coloring and wireless channel allocation. The objective is to minimize the number of colors while avoiding conflicts between adjacent vertices illustrates in Equations (1-3):

$$c: V \rightarrow \{1, 2, \dots, k\} \quad (1)$$

$$\min k \quad (2)$$

$$c(u) \neq c(v), \forall (u, v) \in E. \quad (3)$$

The coloring function $c: V \rightarrow \{1, 2, \dots, k\}$ assigns one color to each vertex in the graph, where $\forall$ denotes for all, V is the set of vertices, and k is the total number of available colors. The objective $\min k$ minimizes the number of colors required. The constraint $c(u) \neq c(v)$, for all $(u, v) \in E$, ensures that every pair of adjacent vertices receives different colors, where u and v are vertices and E is the set of edges representing conflicts. During evolutionary optimization, the number of conflicts is evaluated using Equation (4).

$$F(X) = \sum_{(u,v) \in E} 1[c(u) = c(v)] \quad (4)$$

where X is a candidate coloring and $1[\cdot]$ equals 1 when the condition is true and 0 otherwise, $c(u)$ and $c(v)$ are color assigned, a solution is feasible when $F(X) = 0$. Equation (2) minimizes the color count, while Equation (4) guides the evolutionary search toward conflict-free solutions for a fixed k .

3.1. Independent-Set Extraction and Clique-Based Lower Bound

Independent set extraction and clique detection in the suggested approach are applied to minimize conflicts during the initialization phase and help the search to reach the minimal number of colors. These operations support the research objective of minimizing the number of colors and conflicting edges while improving search efficiency. Equation (5) shows Independent-set extraction.

$$I \subseteq V, (u, v) \notin E, \forall u, v \in I, u \neq v \quad (5)$$

Where I represents an independent set containing vertices that share the same color, V is the complete set of vertices in the input graph, u, v are two distinct vertices selected, E is the set of edges representing conflicts between vertices. Since no two vertices in I are connected by an edge, it receives the same color without creating conflicts. The proposed approach uses independent sets to construct color classes and generate promising initial solutions. This helps reduce the number of conflicting edges and guides the search toward a smaller color count. Equation (6) demonstrates Clique identification.

$$(u, v) \in E, \forall u, v \in Q, u \neq v \quad (6)$$

Every vertex in a clique is connected to every other vertex in that clique Q . Therefore, each vertex must receive a different color u, v . The proposed approach uses clique detection to identify highly constrained vertices and guide the coloring process E , particularly when assigning channels to mutually interfering wireless nodes.

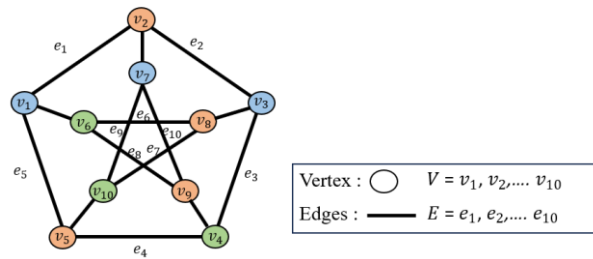


Figure 1. Representation Illustrating Vertex Connectivity and Color Assignment for Graph Coloring

Figure 1 illustrates graph coloring, where vertices ($V = v_1, v_2, \dots, v_{10}$) are connected by edges ($E = e_1, e_2, \dots, e_{10}$), and different colors are assigned to adjacent vertices to avoid conflicts. This ensures valid coloring while minimizing the total number of colors required.

$$\omega(G) \leq \chi(G) \quad (7)$$

Equation (7) shows Clique-based lower bound where $\omega(G)$ is the size of the largest clique in graph G , $\chi(G)$ is chromatic number, representing the minimum number of colors required for a valid coloring and $G = (V, E)$ is the input graph consisting of vertices V and edges E .

3.2 Greedy Initial Coloring Construction

The suggested algorithm uses a greedy construction approach to obtain the initial feasible or low-conflicting color assignment before applying the evolutionary computation technique. In this instance, in each step, one of the non-colored vertices with a maximum degree or saturation level is chosen and assigned the color with the minimum number of conflicting-colored neighbors.

$$X^* = \arg \min_{X \in \mathcal{P}_g} J(X) \quad (8)$$

In Equation (8), X^* denotes the best initial coloring solution, $\mathcal{P}_g$ represents the set of candidate colorings generated by the greedy construction, and $J(X)$ is the objective function that evaluates the number of conflicting edges and colors used. This equation identifies the candidate coloring with the minimum objective value for the proposed approach.

$$J_T(X_i) = \frac{1}{T} \sum_{t=1}^T [\alpha F(X_i^{(t)}) + \beta K(X_i^{(t)})] \quad (9)$$

Equation (9) shows $J_T(X_i)$ denotes the finite-iteration mean objective value of candidate coloring X_i , T is the number of construction iterations, $F(X_i^{(t)})$ represents the number of conflicting edges at iteration t , $K(X_i^{(t)})$ is the number of colors used, and α and β are weighting parameters for conflict minimization and color reduction, respectively.

3.3. Adaptive probability matching for operator selection

Adaptive Probability Matching (APM) is a technique that modifies the probability of choosing operators like crossover and mutation depending on performed in the past. The suggested approach of Adaptive Parallel Evolutionary Algorithm uses APM to choose operators that have shown greater conflict resolution in graph coloring. Let $\mathcal{O} = \{O_1, O_2, \dots, O_q\}$ represent the available crossover and mutation operators. The selection probability of reward obtained by operator O_j during generation t .

$$p_j(t) = p_{\min} + (1 - qp_{\min}) \frac{w_j(t)}{\sum_{\ell=1}^q w_{\ell}(t)} \quad (10)$$

Equation (10) illustrates $p_j(t)$ is the probability of selecting operator O_j , $w_j(t)$ is its estimated performance, q is the number of operators, and $p_{\min}$ is the minimum selection probability, satisfying $0 \leq p_{\min} \leq 1/q$. For graph-coloring objective, define the reward in Equation (11).

$$r_j(t) = \max(0, F(X_{\text{parent}}) - F(X_{\text{offspring}})) \quad (11)$$

where $r_j(t)$ measures the reduction in conflicting edges after applying operator O_j . The operator performance estimate is updated using Equation (12).

$$w_j(t+1) = (1 - \alpha)w_j(t) + \alpha r_j(t) \quad (12)$$

The operator performance weight $w_j(t+1)$ is updated using the previous weight $w_j(t)$ and the reward $r_j(t)$ obtained by operator O_j during generation t . Here, α is the learning rate, t denotes the current generation, and j identifies the selected crossover or mutation operator. The reward reflects the reduction in conflicting edges, helping approach prioritize operators that improve graph coloring and support the objective of minimizing conflicts and the number of colors.

3.4 Tabu Search-Based Conflict Reduction

In the proposed Adaptive Parallel Evolutionary Approach, tabu search is employed to refine candidate graph colorings by reducing conflicting edges. The method iteratively changes the color of a conflicting vertex and selects a promising neighboring solution while using tabu restrictions to prevent repeated moves. Equation (13) shows Conflict objective function.

$$C(F) = |C(E)| = \sum_{\{u,v\} \in E} 1 [F(u) = F(v)] \quad (13)$$

The term F is a candidate coloring, $C(E)$ is the set of conflicting edges, C is the edge set of graphs $G = (V, E)$, $F(u)$ and $F(v)$ are the colors assigned to adjacent vertices, and $1[\cdot]$ equals 1 when the adjacent vertices have the same color and 0 otherwise. The objective is to minimize $C(F)$, achieving a conflict-free coloring when $C(F) = 0$.

$$d_c(v) = \sum_{u \in N(v)} 1 [F(u) = F(v)] \quad (14)$$

Equation (14) Conflicting degree of a vertex, where $d_c(v)$ is the conflicting degree of vertex v , and $N(v)$ is the set of vertices adjacent to v . A higher value identifies a vertex that contributes more conflicting edges and is therefore a suitable candidate for recoloring.

$$\Delta C(v, a) = C(F_{v \rightarrow a}) - C(F) \quad (15)$$

Equation (15) demonstrates illustrates Tabu-search move evaluation. Where $F_{v \rightarrow a}$ denotes the new coloring obtained by assigning color a to vertex v , and $\Delta C(v, a)$ represents the change in the number of conflicting edges. A negative value indicates conflict reduction, while a positive value indicates an increase in conflicts.

$$(v^*, a^*) = \arg \min_{(v,a) \in \mathcal{N}(F) \setminus \mathcal{T}} C(F_{v \rightarrow a}) \quad (16)$$

Equation (16) shows Tabu-restricted move selection, where $\mathcal{N}(F)$ is the neighborhood of candidate coloring F , consisting of possible single-vertex recoloring moves, and $\mathcal{T}$ is the set of currently tabu moves. The pair (v^*, a^*) identifies the selected vertex and its new color. Tabu restrictions prevent recently reversed moves, helping the search avoid cycling. An aspiration rule permits a tabu move if it produces a solution better than the best solution found so far.

3.6 Fixed-k Graph Coloring Optimization

The fixed-k graph coloring optimization in the proposed approach aims to minimize the number of colors while eliminating conflicting edges. It iteratively evaluates candidate colorings using k colors and reduces k whenever a conflict-free solution is obtained, improving search efficiency and identifying compact, valid color assignments for graph coloring and wireless channel allocation. Equation (17) shows Residual-inspired k-color optimization

$$F^*(k) = \arg \min_c \{C(F) : F(v) \in \{1, \dots, k\} \forall v \in V\} \quad (17)$$

where $F^*(k)$ denotes the best coloring found using k colors, F is a candidate coloring, $C(F)$ is the number of conflicting edges, V is the vertex set, and k is the number of available colors. The objective is to minimize conflicts while restricting the coloring to k colors.

$$C(F) = \sum_{\{u,v\} \in E} 1 [F(u) = F(v)] \quad (18)$$

Equation (18) illustrates where E is the edge set, u and v are adjacent vertices, and $1[F(u) = F(v)]$ equals 1 when both vertices have the same color and 0 otherwise. A coloring is valid when $C(F) = 0$.

$$k_{t+1} = \begin{cases} k_t - 1, & \text{if } C(F^*(k_t)) = 0, \\ k_t, & \text{otherwise.} \end{cases} \quad (19)$$

Equation (19) evaluate the Fixed-k reduction where k_t is the current number of colors at iteration t , and k_{t+1} is the number used in the next search. The algorithm reduces k only after finding a conflict-free coloring; otherwise, it retains the best feasible coloring found previously. The provided equations (20-23) are Channel demand constraint, channel assignment constraint, Interference constraint and CAP objective function.

$$\sum_{q=1}^m F_{iq} - d_i = 0, i = 1, \dots, n \quad (20)$$

This equation ensures that the number of channels assigned to each cell satisfies its demand d_i . Here, F_{iq} represents the binary assignment of channel q to cell i , and m denotes the total number of available channels.

$$F_{iq} \in \{0, 1\}, i = 1, \dots, n, q = 1, \dots, m \quad (21)$$

This equation restricts channel assignments to binary values, where $F_{iq} = 1$ indicates an assigned channel and $F_{iq} = 0$ indicates an unassigned channel. This representation enables the proposed approach to optimize channel allocation systematically.

$$\sum_{\substack{j=1 \\ j \neq i, c_{ij} > 0}}^n \sum_{\substack{q=p-(c_{ij}-1) \\ q \neq p \\ 1 \leq q \leq m}}^{p+(c_{ij}-1)} F_{jq} > 0 \quad (22)$$

Here, n is the Total number of vertices with index j & i , c_{ij} is the required channel separation, p is reference channel index, q is the index of the channel being evaluated, m is the total number of available channels and F_{jq} denotes Binary variable indicating whether channel q is assigned to cell j . This represents neighboring-cell channel assignments within the required separation range. The summation limits must be restricted to valid channel indices $1 \leq q \leq m$.

$$\begin{aligned} C(F) = & \sum_{i=1}^n \left(\sum_{q=1}^m F_{iq} - d_i \right) \\ & + \sum_{i=1}^n \sum_{p=1}^m \left(\sum_{\substack{q=p-(c_{ii}-1) \\ q \neq p \\ 1 \leq q \leq m}}^{p+(c_{ii}-1)} F_{iq} \right) F_{ip} \\ & + \sum_{i=1}^n \sum_{p=1}^m \left(\sum_{\substack{j=1 \\ j \neq i, c_{ij} > 0}}^n \sum_{\substack{q=p-(c_{ij}-1) \\ 1 \leq q \leq m}}^{p+(c_{ij}-1)} F_{jq} \right) F_{ip} \end{aligned} \quad (23)$$

Here, $C(F)$ represents Total conflict, d_i is the required number of channels for cell i , c_{ii} is the co-channel separation parameter, F_{ip} indicates whether channel p is assigned to cell i . The Adaptive Parallel Evolutionary Approach integrates graph reduction, greedy initialization, parallel evolutionary search, adaptive operator application, and tabu search. Independent set extraction and clique detection used for coloring the graph, while crossover and mutation operators are used to examine the solution of candidate. Elite migration maintains population diversity, and conflict-directed recoloring diminishes conflicts. The fixed-(k) strategy iteratively minimizes colors, while channel-demand and interference constraints ensure feasible wireless channel allocation. Algorithm 1 summarizes the integrated optimization procedure.

Algorithm 1: Pseudocode of the Adaptive Parallel Evolutionary Approach for Graph Coloring and Wireless Channel Allocation

Begin

1. Construct graph $G = (V, E)$.
2. Extract independent sets and cliques; compute $\omega(G)$.
3. Generate greedy initial colorings.
4. Evaluate candidates using conflict $F(X)$ and fitness $J(X)$.
5. Initialize parallel islands, populations, and operator weights.
6. While the generation limit is not reached:

For each island in parallel:

- a. Select crossover/mutation using adaptive probabilities.
- b. Generate and evaluate offspring.
- c. Apply tabu search to reduce conflicts.
- d. Update operator rewards and weights.
- e. Replace inferior candidates.

end for

Migrate elite candidates between islands.

Update the best coloring C_best .

```

if  $F(C\_best) = 0$ :
    Store the valid coloring and update  $k\_best$ .
    if  $k\_best = \omega(G)$ , terminate.
    Otherwise, reduce  $k$  and restart the search.
end if
7. Return  $C\_best$ ,  $k\_best$ , and  $F(C\_best)$ .
end

```

4. RESULT AND DISCUSSION

The suggested adaptive parallel evolutionary approach was tested for the assessment of its efficiency in color and conflict minimization and increase in computational efficiency during graph coloring and channel allocation. Table 1 provides information about software used and evolutionary parameters.

Table 1. Experimental Configuration and Parameter Settings

Parameter	Value
Population size per island	20
Number of parallel islands	4
Maximum generations	100
Crossover and mutation	Adaptive probability matching
Local search	Tabu search
Elite migration interval	Every 10 generations
Initial solution	Greedy coloring with degree/saturation-based selection
Optimization strategy	Fixed-k color reduction
Stopping criterion	Maximum generations or unsuccessful reduced- k search

The Discrete Mathematics and Theoretical Computer Science (DIMACS) benchmarks are used in this research to evaluate the proposed algorithm's coloring quality, computational time, scalability, and success rate across graphs of varying sizes and structures and wireless Channel Assignment Problem (CAP) instances. The DIMACS instances, including queen5_5.col, myciel4.col, myciel5.col, anna.col, zero.in.i.1.col, DSJC125.5.col, DSJC500.5.col, DSJC1000.5.col, R1000.1.col, flat1000_50_0.col, flat1000_76_0.col are used to assess graph-coloring performance across different graph sizes and structures. The CAP instances (1-10) are used to evaluate channel allocation under interference constraints. The results are analyzed in terms of the number of colors obtained, computation time, and successful-run rate to determine the quality of the color and the computational efficiency. The analysis of the CAP instances is done in terms of the cells and channel counts, run length, fitness threshold, Frequency of Convergence, and conflict function $C(F)$ to examine allocation feasibility and convergence reliability.

Table 2. Performance measures of the proposed approach on small DIMACS graph ($V < 100$)

Graph instance	Vertices (V)	Edges (E)	$\chi(G)$	Colors obtained	Computation time (s)	Successful runs (%)
queen5_5.col	25	320	5	5	0.12	96
myciel4.col	23	71	5	5	0.09	94
myciel5.col	47	236	6	6	0.18	93

Table 2 evaluates the proposed approach on small DIMACS graphs of vertex value less than 100 using vertex count, edge count, chromatic number ($\chi(G)$), colors obtained, computation time, and successful-run rate. Among the evaluated instances, myciel4.col is the fastest, requiring 0.09 seconds, while queen5_5.col achieves the highest successful-run rate of 96%. Figure 2 outlines the graphical representation.

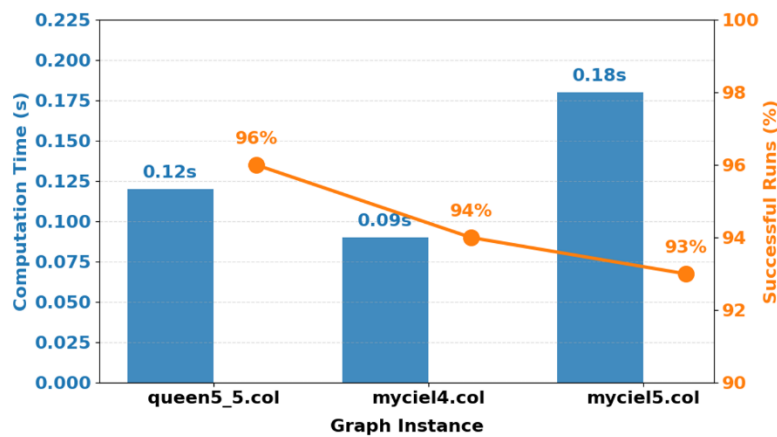


Figure 2. Visualization of the comparison of Computation time (s) and Successful runs (%) for smaller graphs

Table 3. Performance measures of the proposed approach on intermediate DIMACS graph ($100 \leq V < 500$)

Graph instance	Vertices (V)	Edges (E)	$\chi(G)$	Colors obtained	Computation time (s)	Successful runs (%)
anna.col	138	493	11	11	0.42	92
zeroin.i.1.col	211	4100	49	49	1.26	89
DSJC125.5.col	125	3891	17	17	0.86	87

The proposed approach's performances on intermediate graphs (vertex value with 100 and in between 100 to 500) in terms of their varying connectivity and coloring complexity are illustrated in Table 3. anna.col is the fastest in terms of computational time, taking 0.42 seconds to execute with the success rate of 92%. The results for anna.col, zeroin.i.1.col, and DSJC125.5.col demonstrate how the approach works on graphs with different vertices and edges. The computation time comparison of intermediate graphs is depicted in Figure 3.

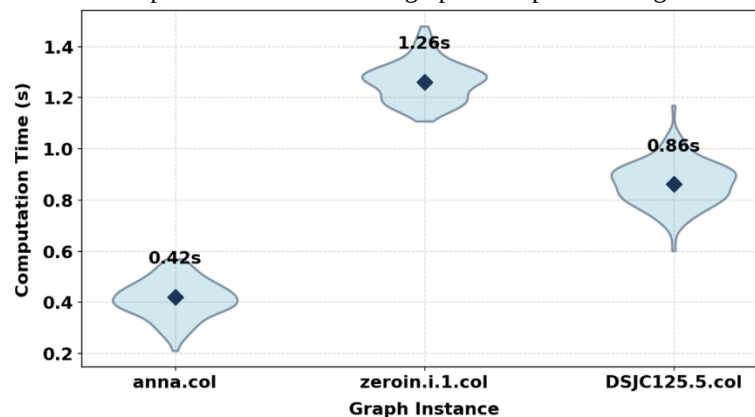


Figure 3. Visualization of Computation time (s) for intermediate graphs instances

Table 4. Performance measures and scalability of the proposed approach on large DIMACS graph ($V \geq 500$)

Graph instance	Vertices (V)	Edges (E)	$\chi(G)$	Colors obtained	Computation time (s)	Successful runs (%)
DSJC500.5.col	500	62624	48	49	4.85	84
DSJC1000.5.col	1000	249826	83	84	15.72	81
R1000.1.col	1000	14378	20	21	7.64	85
flat1000_50_0.col	1000	245000	50	50	12.48	82
flat1000_76_0.col	1000	246708	76	78	14.36	78

Table 4 shows the scalability analysis of the proposed approach on large DIMACS graph (vertex value larger than or equals to 500) coloring instances. DSJC500.5.col is found to take the minimum computational time of 4.85

seconds, whereas R1000.1.col attains the highest successful run rate of 85%. Figure 4 indicates the computational effectiveness and reliability for large graphs of various vertices and edges.

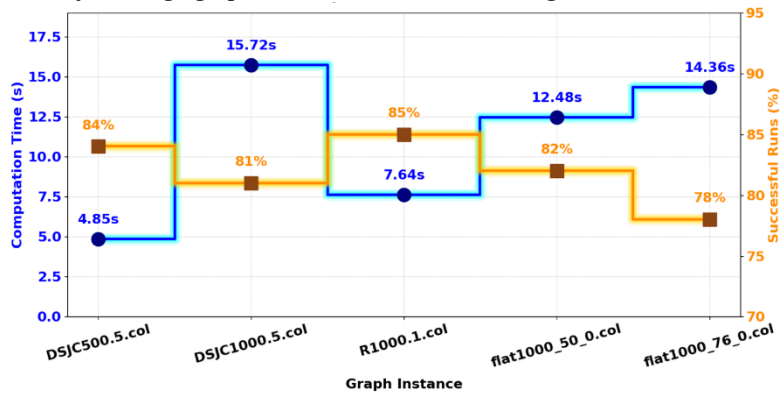


Figure 4. Illustration of Computation time and successful runs for larger graphs instances

Table 5. Performance measures of the proposed approach on wireless channel assignment problem instances

CAP instance	No. of Cells (n)	No. of Channels (m)	Run Length	Fitness Threshold	Frequency of Convergence (FoC)	C(F)
1	8	20	500	0.010	100	0
2	10	30	500	0.010	100	0
3	12	40	750	0.010	98	0
4	14	50	750	0.010	98	1
5	16	60	1,000	0.010	96	0
6	18	70	1,000	0.010	96	1
7	20	80	1,250	0.010	94	0
8	22	90	1,250	0.010	92	1
9	24	100	1,500	0.010	90	0
10	26	120	1,500	0.010	88	2

Table 5 presents the performance of the proposed Approach across ten wireless Channel Assignment Problem instances. Instance 1 achieves the highest frequency of convergence (100%) with zero conflicts, while instances 2 and 3 also maintain 100% and 98% convergence, respectively. Lower C(F) values indicate better conflict reduction and allocation feasibility.

4.1 Discussion

Research focused on reducing the colors and conflicting edges while achieving higher computational efficiency in graph coloring and wireless channel allocation. Some of the limitations of existing work include graph-coloring performance dependency on graph structure in [11], while [12] confined itself to comparing only two heuristics, restricting broader evaluation. Research [14] developed a graph neural network model with inadequate information about the dataset and benchmarks for replicability and generalization purposes to other graphs. The proposed approach overcomes such limitations through a combination of independent set extraction, clique identification, parallel island-based evolution, adaptive operator selection, elite migration, and tabu search. These techniques allow for different explorations of graphs, dynamic prioritization of better operators, reduce coloring conflicts, and gradual reduction of colors usage while incorporating demand on the channels and interference constraints for wireless allocation. In practice, this approach can allow for effective frequency allocation and resource scheduling in wireless systems by reducing interference conflicts and computational overhead.

5. CONCLUSION

Development of adaptive parallel evolutionary approach for effective graph coloring and wireless channel allocation through color reduction, conflict reduction and faster computation was achieved in this research. The suggested approach use independent set extraction, clique-based lower bound generation, greedy initialization, parallel island-based evolutionary search, adaptative probability matching, elite migration, tabu search refinement, and fixed k optimization to produce color assignments under graph and channel-allocation constraints. Results were performed effectively across different DIMACS graph instances. For small graphs, myciel4.col achieves the

shortest computation time of 0.09 s, while queen5_5.col records the highest success rate of 96%. For intermediate graphs, anna.col requires 0.42 s with a 92% success rate. Among large graphs, DSJC500.5.col records the shortest computation time of 4.85 s, while R1000.1.col achieves the highest success rate of 85%. In terms of channel allocation, the CAP instances 1 and 2 have attained 100% convergence with no conflicts, which shows good allocation without any interference. However, the approach has not yet been validated in real-world dynamic wireless networks, and its performance depends on graph structure and optimization settings. Future research can explore real-world applications of wireless deployment, dynamic channel demands, large graph instances, and comparison with evolutionary and optimization techniques.

REFERENCES

1. Gimenez-Guzman, J.M., Marsa-Maestre, I., de la Hoz, E., Orden, D. and Herranz-Oliveros, D., 2023. Channel selection in uncoordinated IEEE 802.11 networks using graph coloring. *Sensors*, 23(13), p.5932. <https://doi.org/10.3390/s23135932>
2. Terci, G.S. and Boz, B., 2025. Coloring dynamic graphs with a similarity and pool-based evolutionary algorithm. *IEEE Access*, 13, pp.38054-38075. <https://doi.org/10.1109/ACCESS.2025.3546108>
3. Ge, C., Xia, S., Chen, Q. and Adachi, F., 2022. 2-layer interference coordination framework based on graph coloring algorithm for a cellular system with distributed MU-MIMO. *IEEE Transactions on Vehicular Technology*, 72(3), pp.3557-3568. <https://doi.org/10.1109/TVT.2022.3219411>
4. Švarcmajer, M., Ivanović, D., Rudec, T. and Lukić, I., 2025. Application of graph theory and variants of greedy graph coloring algorithms for optimization of distributed peer-to-peer blockchain networks. *Technologies*, 13(1), p.33. <https://doi.org/10.3390/technologies13010033>
5. Sun, K., Zhang, J., Gao, X., Huang, W., Zhang, H. and Leung, V.C., 2024. Dynamic channel allocation scheme based on traffic prediction in dense wireless networks. *IEEE Transactions on Wireless Communications*, 23(9), pp.12331-12342. <https://doi.org/10.1109/TWC.2024.3391426>
6. Lee, W. and Seo, J.B., 2023. Deep learning-aided channel allocation scheme for wlan. *IEEE Wireless Communications Letters*, 12(6), pp.1007-1011. <https://doi.org/10.1109/LWC.2023.3257128>
7. Ascoli, A., Weiher, M., Herzig, M., Slesazek, S., Mikolajick, T. and Tetzlaff, R., 2022. Graph coloring via locally-active memristor oscillatory networks. *Journal of Low Power Electronics and Applications*, 12(2), p.22. <https://doi.org/10.3390/jlpea12020022>
8. Zhou, B., Li, S., Zi, B., Chen, B. and Zhu, W., 2023. Multi-objective optimal design of a cable-driven parallel robot based on an adaptive adjustment inertia weight particle swarm optimization algorithm. *Journal of Mechanical Design*, 145(8), p.083301. <https://doi.org/10.1115/1.4062458>
9. Mansour, R.F., Abdel-Khalek, S., Basha, S.M., Khayyat, M.M., Elnaim, B.M. and Shankar, V., 2022. Adaptive Parallel Seeker Optimization-based Route Planning for clustered WSN in smart cities. *Computers and Electrical Engineering*, 102, p.108289. <https://doi.org/10.1016/j.compeleceng.2022.108289>
10. Mosavi, S.A.A., Erfanian, A. and Sabeghi, N., 2026. Design and implementation of a graph-coloring algorithm for optimizing flight-level allocation in air traffic management in Iran. *Iranian Journal of Numerical Analysis & Optimization*, 16(2), p.590. <https://doi.org/10.22067/ijnao.2026.95911.1749>
11. Yakut, S., 2025. Efficient and robust graph coloring algorithm for graph coloring problem: Malatya vertex coloring algorithm. *IEEE Access*, 13, pp.168512 - 168526. <https://doi.org/10.1109/ACCESS.2025.3614296>
12. KraleV, V. and KraleV, R., 2023. A comparative analysis between two heuristic algorithms for the graph vertex coloring problem. *International Journal of Electrical and Computer Engineering (IJECE)*, 13(3), p.2981. <https://doi.org/10.11591/ijece.v13i3.pp2981-2989>
13. Marappan, R. and Sethumadhavan, G., 2022. Solving graph coloring problem using divide and conquer-based turbulent particle swarm optimization. *Arabian Journal for Science and Engineering*, 47(8), pp.9695-9712. <https://doi.org/10.1007/s13369-021-06323-x>
14. Gao, M. and Hu, J., 2024. Graph coloring algorithm based on minimal cost graph neural network. *IEEE Access*, 12, pp.168000-168009. <https://doi.org/10.1109/ACCESS.2024.3439352>
15. Sano, Y., Norimoto, M. and Ishikawa, N., 2023. Qubit reduction and quantum speedup for wireless channel assignment problem. *IEEE Transactions on Quantum Engineering*, 4, pp.1-12. <https://doi.org/10.1109/TQE.2023.3293452>

16. Shahryari, M.S., Farzinvas, L., Feizi-Derakhshi, M.R. and Taherkordi, A., 2023. High-throughput and energy-efficient data gathering in heterogeneous multi-channel wireless sensor networks using genetic algorithm. *Ad Hoc Networks*, 139, p.103041. <https://doi.org/10.1016/j.adhoc.2022.103041>
17. Logeshwaran, J., Shanmugasundaram, R.N. and Lloret, J., 2024. Load based dynamic channel allocation model to enhance the performance of device-to-device communication in WPAN. *Wireless Networks*, 30(4), pp.2477-2509. <https://doi.org/10.1007/s11276-024-03680-x>