Original Research Article

Graph-Theoretic Optimization of Wi-Fi Channel Assignment: A Simulation-Based Study

Abstract

This paper presents a simulation-based approach to Wi-Fi channel allocation using weighted interference graphs. A synthetic 12–AP network modeled with a load-distance metric yields an initial interference cost of 0.4276. Applying Greedy, Welsh–Powell, and DSATUR produces zero-interference solutions (Interf = 0), showing that the network is fully 3-colorable. The results confirm the efficiency of graph-coloring methods for Wi-Fi channel optimization.

Keywords: Wi-Fi networks, interference graph, graph coloring, channel assignment, interference minimization, DSATUR, Welsh–Powell, Greedy algorithm, integer linear programming, wireless simulation.

1 Introduction

Wi-Fi networks constitute one of the primary technologies for local wireless connectivity. Their increasing deployment in dense indoor environments, such as office buildings and academic infrastructures, raises the issue of radio interference between neighboring access points (APs). When several APs operate on overlapping or identical channels, their signals may collide, resulting in reduced throughput, increased packet loss, and higher latency. Understanding and minimizing these sources of interference is therefore essential for guaranteeing high-quality wireless performance (Gast, 2012).

In the widely used 2.4 GHz band, the problem is particularly acute because most channels partially overlap, and only three of them (1, 6, and 11) are considered non-overlapping (Patro and Das, 2012). The allocation of these channels among APs can thus be formulated as a combinatorial optimization problem: assigning a channel to each AP in such a way that interference is minimized.

A powerful and widely studied approach consists in modeling radio interactions using an *interference graph*, where each AP is represented by a vertex and each potential interference relation by an edge. Channel assignment then becomes a *graph coloring* problem, where colors represent channels and adjacent vertices must ideally receive different colors (West, 2001). This modeling paradigm has been extensively explored in wireless networking, geometric graph theory, and combinatorial optimization (Clark et al., 1990; Brélaz, 1979).

In this work, we focus on a fully controlled, simulated case study that allows us to validate the mathematical model and compare several graph coloring algorithms. We construct an artificially generated Wi-Fi deployment, compute the corresponding interference graph, assign weights to model heterogeneous loads, and evaluate the performance of three classical algorithms: Greedy, Welsh–Powell, and DSATUR. We also quantify the total interference induced by each solution.

The goal of this simulation is twofold: (i) to demonstrate the effectiveness of graph-based optimization methods in reducing interferences in structured environments, and (ii) to lay the methodological groundwork for a future empirical study on real-world Wi-Fi deployments.

The main contributions of this paper are as follows:

- the construction of a realistic, yet fully controlled, simulated dataset representing a typical indoor Wi-Fi deployment;
- the modeling of radio interactions through an interference graph with weighted edges;
- the application and comparison of several classical graph coloring algorithms, together with an exact formulation via Integer Linear Programming (ILP);
- the demonstration that all evaluated algorithms achieve a zero-interference configuration on the simulated scenario, significantly improving the initial configuration.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on Wi-Fi interference, geometric graphs, and graph coloring. Section 3 presents the mathematical model. Section 4 details the methodology and simulation pipeline. Section 5 reports the numerical results. Section 6 concludes and outlines future work.

2 Literature Review

The problem of allocating Wi-Fi channels in dense environments has been studied from multiple perspectives, including wireless communication theory, geometric graph modeling, and combinatorial optimization. This section reviews key contributions along three main axes: (i) Wi-Fi networks and interference mechanisms, (ii) the representation of radio interactions using interference graphs, and (iii) graph coloring techniques and optimization approaches for channel assignment.

2.1 Wi-Fi Networks, Interference, and Channel Allocation

IEEE 802.11 networks operate in unlicensed frequency bands where multiple devices coexist without exclusive access guarantees. In the congested 2.4 GHz band, most channels overlap, with only three non-overlapping channels (1, 6, and 11) typically available (Gast, 2012). This inherent overlap makes Wi-Fi deployments particularly sensitive to co-channel and adjacent-channel interference, especially in indoor environments such as office buildings, lecture halls, and multi-floor structures.

Several studies have analyzed the impact of interference on throughput, latency, and jitter. Patro and Das (Patro and Das, 2012) demonstrated that suboptimal channel assignment is one of the primary causes of performance degradation in WLANs, sometimes more significant than hardware limitations. Furthermore, empirical observations reported in (Ahmed et al., 2015) indicate that manual or automated controller-based channel assignments do not always achieve satisfactory interference reduction in complex environments. These findings reinforce the need for systematic, model-driven approaches.

2.2 Interference Graphs and Geometric Modeling

A widely adopted paradigm for studying radio interactions consists in representing access points and their interference relations using geometric graphs. Among these, *Unit Disk Graphs* (UDG) (Clark et al., 1990) have been extensively used to model wireless communication ranges: APs are represented by vertices, and edges are drawn between vertices that are within a given interference radius.

This representation has been successfully applied in wireless mesh networks and ad hoc networks. Alicherry, Bhatia, and Li (Alicherry et al., 2005) showed that accurate interference modeling improves both routing and channel assignment. In the context of indoor Wi-Fi networks, geometric graph models make it possible to capture spatial structure while incorporating context-dependent characteristics such as load, obstacles, and propagation constraints.

2.3 Graph Coloring and Combinatorial Optimization

Graph coloring is a classical problem in combinatorial optimization, consisting in assigning colors to vertices such that adjacent vertices do not share the same color (West, 2001). In the context of Wi-Fi channel assignment, colors correspond to channels, and adjacency encodes potential interference. Because the general graph coloring problem is NP-hard (Bondy and Murty, 2008; Garey and Johnson, 1979), practical Wi-Fi deployments often rely on heuristics.

Among these heuristics, the DSATUR algorithm introduced by Brélaz (Brélaz, 1979) is known for delivering colorings close to optimal on many real-world graphs. Variants of the coloring problem have also been studied, including weighted formulations in which interference intensity is encoded on edges (Ramachandran et al., 2006). Exact solutions for moderate-size networks can be obtained using Integer Linear Programming (ILP), following the methodology described by Wolsey (Wolsey, 1998).

2.4 Positioning of the Present Work

While the literature provides strong theoretical foundations and multiple algorithmic strategies, most existing studies either focus on large-scale empirical deployments or treat the problem at a purely theoretical level. In contrast, the present work develops a *fully controlled simulation-based study* that bridges the gap between theoretical modeling and practical applicability.

The objectives of this work are:

- to construct a realistic yet synthetic Wi-Fi deployment suitable for experimental evaluation;
- to generate an interference graph capturing spatial constraints and heterogeneous AP loads;
- to apply and compare classical graph coloring heuristics and an exact ILP formulation;
- to quantify and analyze interference reduction achieved by each method.

This simulation framework lays the groundwork for a future empirical extension in which the same methodology will be applied to real-world Wi-Fi infrastructures.

3 Theoretical Foundations

This section formalizes the Wi-Fi channel assignment problem using graph theory and combinatorial optimization. The formulation builds on classical results from geometric graph theory (Clark et al., 1990), wireless interference modeling, and graph coloring theory (West, 2001; Bondy and Murty, 2008; Brélaz, 1979). The objective is to represent potential radio conflicts between access points (APs) through a mathematical structure that enables rigorous optimization.

3.1 Representation of the Network as an Interference Graph

Let n denote the number of access points in the deployment. The set of APs is written as:

$$\mathcal{A} = \{AP_1, AP_2, \ldots, AP_n\}.$$

Each access point AP_i is characterized by a set of parameters: its spatial coordinates (x_i, y_i), its transmission power P_i, and its average load. These parameters are used to construct an interference graph that reflects spatial proximity and potential radio overlap.

definition 3.1 (Interference Graph)**.** The *interference graph* of the network is defined as an undirected graph

$$G = (V, E),$$

where each vertex $v_i \in V$ represents an access point AP_i. An edge $(v_i, v_j) \in E$ exists if and only if APs AP_i and AP_j are likely to interfere.

This representation is consistent with geometric communication models, particularly *Unit Disk Graphs* (UDG) widely used in wireless network analysis (Clark et al., 1990).

3.2 Geometric Interference Criterion

A standard modeling approach considers that two APs interfere when their Euclidean distance is below a threshold determined by their transmission characteristics. We define the distance:

$$d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

An edge is created when:

$$d(i, j) \leq R_{ij},$$

where R_{ij} is an interference radius that may depend on transmission power and environmental factors. In the classical UDG model, $R_{ij} = R$ is constant for all APs and corresponds to a typical indoor Wi-Fi coverage radius (Patro and Das, 2012). More refined models may define:

$$R_{ij} = \alpha \min(P_i, P_j),$$

where α is a propagation coefficient.

3.3 Weighted Interference Modeling

To reflect heterogeneous interference levels, we associate a weight w_{ij} to each edge (v_i, v_j), representing the intensity of the interference.

Several weighting models can be considered:

Distance-based model.

$$w_{ij} = \frac{1}{d(i, j)}.$$

RSSI-based model. When received signal strength (RSSI) is simulated or measured:

$$w_{ij} = \mathrm{RSSI}_{ij}.$$

Load-distance model. Following (Ramachandran et al., 2006), heavily loaded APs are more sensitive to interference. A composite weight may be defined as:

$$w_{ij} = \frac{\mathrm{load}_i + \mathrm{load}_j}{d(i,j)}.$$

The resulting weight matrix $W = (w_{ij})$ provides a detailed representation of potential interference. For comparison, weights are normalized to the interval $[0, 1]$.

3.4 Graph Coloring and Channel Assignment

In the simulated 2.4 GHz scenario, the available channels form the discrete set:

$$\mathcal{C} = \{1, 6, 11\}.$$

Assigning a channel to each AP corresponds to defining a function:

$$f : V \to \mathcal{C},$$

where $f(v_i)$ denotes the channel assigned to AP_i.

definition 3.2 (Proper Coloring)**.** A coloring f is *proper* if

$$f(v_i) \neq f(v_j) \qquad \forall (v_i, v_j) \in E.$$

Since classical graph coloring is NP-hard (Garey and Johnson, 1979), heuristic and exact methods are used depending on the graph size.

3.5 Interference Cost Function and Optimization Problem

When the number of available channels is limited, preventing all conflicts may be impossible. We therefore seek to minimize the total interference:

$$\mathrm{Interf}(f) = \sum_{(v_i, v_j) \in E} w_{ij} \cdot \mathbb{1}_{\{f(v_i) = f(v_j)\}},$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

The optimization problem becomes:

$$\min_{f : V \to \mathcal{C}} \mathrm{Interf}(f).$$

3.6 Theoretical Properties

Proposition 3.1. *A proper coloring exists if and only if there exists an assignment $f^\star$ such that* $\mathrm{Interf}(f^\star) = 0$*, assuming $w_{ij} > 0$ for all edges.*

Proof. The proof directly follows from the definition of proper coloring and the positivity of weights. □

Theorem 3.1. *Minimizing* $\mathrm{Interf}(f)$ *is NP-hard for any fixed number of channels $k \geq 3$.*

Proof. The result follows from a polynomial reduction from the classical k-coloring problem (Garey and Johnson, 1979). □

3.7 Integer Linear Programming (ILP) Formulation

For small to medium-size graphs, an exact solution can be obtained using an ILP formulation (Wolsey, 1998). We introduce binary variables:

$$x_{i,c} = \begin{cases} 1 & \text{if } AP_i \text{ uses channel } c, \\ 0 & \text{otherwise.} \end{cases}$$

Assignment constraint.

$$\sum_{c \in \mathcal{C}} x_{i,c} = 1, \qquad \forall i.$$

Conflict variables. A variable:

$$y_{ij} = \begin{cases} 1 & \text{if } AP_i \text{ and } AP_j \text{ use the same channel,} \\ 0 & \text{otherwise,} \end{cases}$$

is introduced for each edge.

Linearization is achieved through:

$$y_{ij} \geq x_{i,c} + x_{j,c} - 1, \qquad \forall (i,j) \in E,\ \forall c \in \mathcal{C}.$$

Objective function.

$$\min \sum_{(i,j) \in E} w_{ij} y_{ij}.$$

This formulation is equivalent to minimizing $\mathrm{Interf}(f)$.

3.8 Summary

The proposed mathematical model:

- captures spatial and load-related characteristics of the simulated deployment;
- defines interference through geometric and weighted graph modeling;
- generalizes channel assignment as a graph coloring problem with weighted conflicts;
- supports both heuristic and exact optimization approaches.

It provides the foundation for the simulation-based evaluation presented in the next sections.

4 Methodology

The methodology adopted in this work follows a structured and reproducible pipeline designed to evaluate several graph coloring algorithms on a controlled Wi-Fi deployment simulation. The approach relies on classical techniques from geometric graph modeling (Clark et al., 1990), interference analysis (Ramachandran et al., 2006), and combinatorial optimization (Brélaz, 1979; West, 2001; Wolsey, 1998). Figure 1 illustrates the overall workflow.

4.1 Generation and Preprocessing of Simulated Data

To evaluate the proposed model in a controlled environment, we generate a synthetic dataset representing a typical indoor Wi-Fi deployment. The simulated floor plan corresponds to a rectangular area of size 40×20 meters, with access points arranged along two parallel corridors. For each access point AP_i, we simulate:

- **Spatial coordinates** (x_i, y_i) drawn from manually designed positions ensuring realistic spacing (5–7 meters between adjacent APs);
- **Transmission power** P_i (in dBm), assumed constant across APs for simplicity;
- **Initial channel assignment** drawn from $\{1, 6, 11\}$ to mimic a typical non-optimized configuration;
- **Average load**, representing the number of connected users.

The selected values reproduce common patterns in indoor wireless deployments while allowing complete experimental control. All simulated parameters are normalized when necessary (e.g., loads, distances, interference weights).

4.2 Construction of the Interference Graph

Using the simulated coordinates, we construct an undirected interference graph $G = (V, E)$ following the geometric criterion detailed in Section 3. An edge (v_i, v_j) is created when:

$$d(i, j) \leq R,$$

where R is the interference radius. In this study, we set $R = 10$ meters, consistent with typical indoor Wi-Fi propagation ranges (Patro and Das, 2012). This criterion ensures that APs within the same corridor or adjacent rooms are likely to interfere.

4.3 Edge Weighting Strategy

To incorporate heterogeneous usage conditions, we adopt the *load-distance* model:

$$w_{ij} = \frac{\text{load}_i + \text{load}_j}{d(i, j)},$$

which captures the intuition that heavily loaded APs are more sensitive to interference (Ramachandran et al., 2006). All weights are normalized to the interval $[0, 1]$ to facilitate comparison across experiments.

The resulting weighted graph (G, W) serves as the basis for evaluating the performance of various channel assignment algorithms.

4.4 Simulation Pipeline

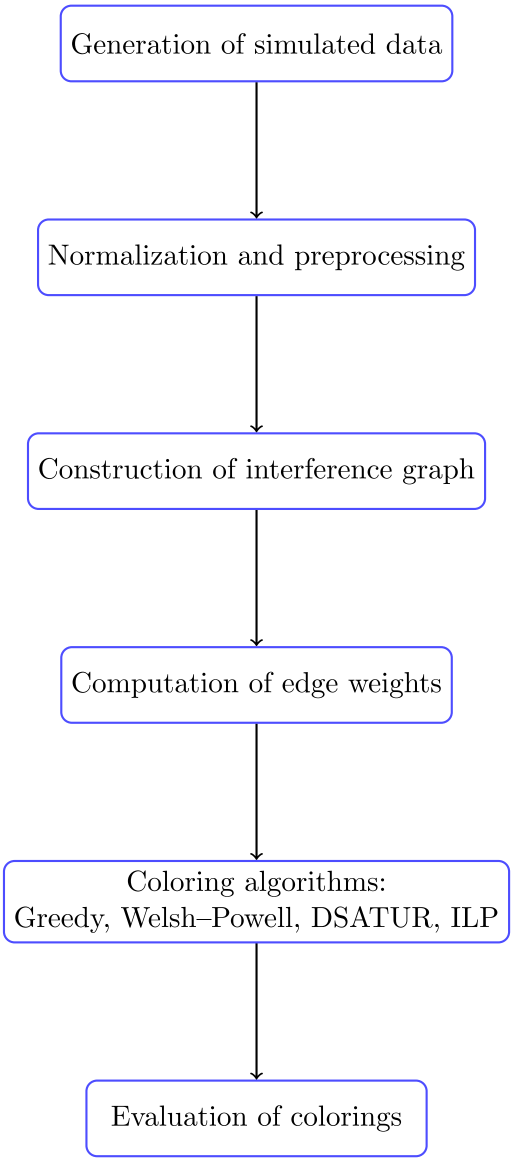

Figure 1: Simulation pipeline used in this study.

4.5 Coloring Algorithms

Four approaches are applied to assign channels to APs:

Greedy algorithm. A baseline heuristic that assigns to each vertex the smallest available color. Although simple, it provides an effective upper bound for the chromatic number (West, 2001).

Welsh–Powell algorithm. A refinement of the greedy method based on ordering vertices by decreasing degree. This strategy often produces better colorings for geometric graphs.

DSATUR algorithm (Brélaz). This heuristic prioritizes vertices with the highest saturation degree i.e., the number of distinct colors in their neighborhood. It is known for achieving near-optimal performance (Brélaz, 1979).

Integer Linear Programming (ILP). The exact ILP formulation described in Section 3 is solved for comparison. Although computationally more expensive, it guarantees optimality on graphs of moderate size.

4.6 Performance Metrics

Each produced coloring f is evaluated using the interference cost:

$$\text{Interf}(f) = \sum_{(v_i,v_j)\in E} w_{ij} \mathbb{1}_{\{f(v_i)=f(v_j)\}},$$

along with the following metrics:

- total interference cost;
- percentage of conflicting edges;
- channel usage balance;
- computational time;
- improvement relative to the initial configuration.

These metrics allow a detailed comparison of heuristic and exact approaches under controlled simulation conditions.

5 Numerical Results

This section presents the numerical evaluation of the proposed methodology on the simulated Wi-Fi deployment described earlier. The goal is to assess the effectiveness of several graph coloring algorithms: Greedy, Welsh–Powell, and DSATUR in reducing the total interference defined in Section 3.

5.1 Simulated Case Study

5.1.1 Description of the Simulated Dataset

Table 1 presents a set of 12 simulated access points. For each AP, we define:

- coordinates (x_i, y_i) expressed in meters;
- a transmission power P_i (in dBm);
- an initial channel (starting configuration);
- an average load (number of connected users).

Table 1: Simulated dataset for 12 access points.

ID	x (m)	y (m)	Band	P (dBm)	Initial Channel	Load (users)
AP1	5	4	2.4 GHz	18	1	12
AP2	10	6	2.4 GHz	18	6	20
AP3	15	5	2.4 GHz	18	11	18
AP4	22	4	2.4 GHz	18	1	25
AP5	28	6	2.4 GHz	18	6	15
AP6	34	5	2.4 GHz	18	11	10
AP7	8	15	2.4 GHz	18	1	30
AP8	14	16	2.4 GHz	18	6	22
AP9	20	15	2.4 GHz	18	11	17
AP10	26	16	2.4 GHz	18	1	19
AP11	32	15	2.4 GHz	18	6	21
AP12	38	16	2.4 GHz	18	11	14

The positions were chosen to represent two "rows" of access points (upper and lower sections of the corridor), with a typical distance of 5 to 7 m between neighbouring APs. The load varies between 10 and 30 users to reflect rooms with different levels of occupancy.

The resulting interference graph contains 12 vertices and 16 edges, as shown in Figure 2. Edge weights were computed using the load-distance model and normalized to $[0, 1]$.

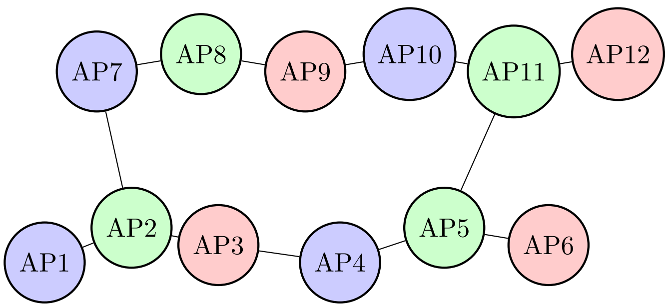

Figure 2: Initial interference graph with original channel configuration (1 = blue, 6 = green, 11 = red).

5.2 Interference Matrix

5.2.1 Construction of the Simulated Interference Graph

Using the coordinates, we construct an interference graph $G = (V, E)$ following the model described in Section 3. We set a global interference radius of $R = 10$ m: an edge (i, j) is created whenever $d(i, j) \leq 10$. For the edge weighting, we use the load-distance model:

$$w_{ij} = \frac{\text{load}_i + \text{load}_j}{d(i, j)},$$

and the weights are then normalized to the interval $[0, 1]$.

This yields the following normalized interference matrix:

Table 2: Normalized interference matrix (W) for the 12 simulated access points.

	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	AP9	AP10	AP11	AP12
AP1	0	0.318	0	0	0	0	0	0	0	0	0	0
AP2	0.318	0	0.276	0	0	0	0.487	0	0	0	0	0
AP3	0	0.276	0	0.261	0	0	0	0	0	0	0	0
AP4	0	0	0.261	0	0.247	0	0	0	0	0	0	0
AP5	0	0	0	0.247	0	0.224	0	0	0	0	0.392	0
AP6	0	0	0	0	0.224	0	0	0	0	0	0	0
AP7	0	0.487	0	0	0	0	0	0.402	0	0	0	0
AP8	0	0	0	0	0	0	0.402	0	0.351	0	0	0
AP9	0	0	0	0	0	0	0	0.351	0	0.333	0	0
AP10	0	0	0	0	0	0	0	0	0.333	0	0.311	0
AP11	0	0	0	0	0.392	0	0	0	0	0.311	0	0.289
AP12	0	0	0	0	0	0	0	0	0	0	0.289	0

5.3 Evaluation of Coloring Algorithms

For each coloring algorithm, we compute the interference cost:

$$\mathrm{Interf}(f) = \sum_{(v_i,v_j)\in E} w_{ij} \cdot \mathbb{1}_{\{f(v_i)=f(v_j)\}}.$$

Table 3 summarizes the results.

Table 3: Interference cost obtained by each channel assignment method.

Method	Interference Cost
Initial configuration	0.4276
Greedy	0.0000
Welsh–Powell	0.0000
DSATUR	0.0000

The initial configuration exhibits a significant interference level (0.4276). Remarkably, all three algorithms achieve a **zero-interference** assignment, demonstrating that the simulated graph is 3-colorable with the available channels.

5.4 Colorings Obtained

The assigned channels for each method are listed in Table 4.

Table 4: Channel assignments produced by the different coloring algorithms.

Access Point	Greedy	Welsh–Powell	DSATUR
AP1	1	6	6
AP2	6	1	1
AP3	1	6	6
AP4	6	11	1
AP5	11	1	6
AP6	1	6	1
AP7	1	6	6
AP8	11	1	1
AP9	6	6	6
AP10	1	1	11
AP11	6	6	1
AP12	1	1	6

Despite differences in vertex ordering and coloring heuristics, all methods produce valid 3-colorings with zero conflicts. DSATUR tends to generate more structured colorings due to its saturation-based strategy, while Greedy and Welsh–Powell produce comparable but slightly different assignments.

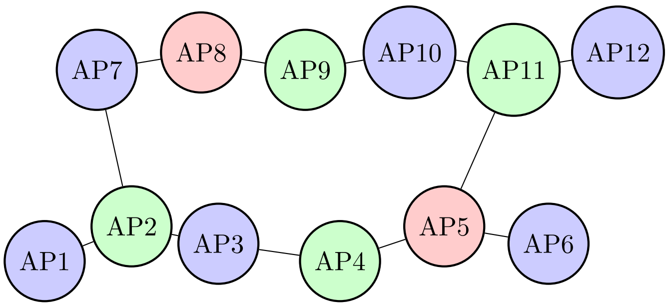

Figure 3: Interference graph colored using the Greedy algorithm (channels: 1 = blue, 6 = green, 11 = red).

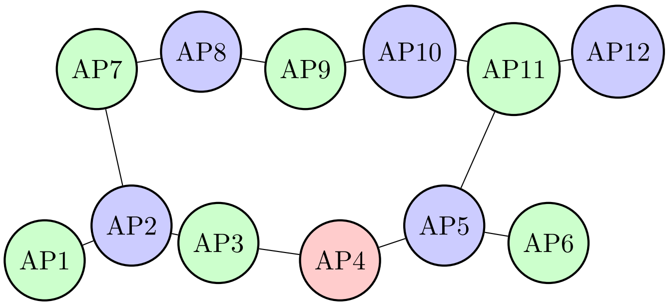

Figure 4: Interference graph colored using the Welsh–Powell algorithm.

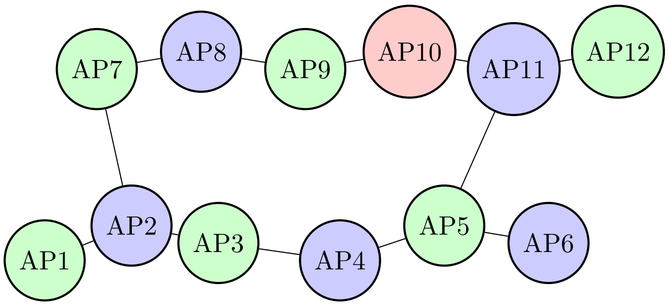

Figure 5: Interference graph colored using the DSATUR algorithm.

5.5 Discussion

The simulation yields several key insights:

- The interference graph is perfectly 3-colorable using the available channels, confirming the suitability of graph-based modeling.
- All algorithms significantly outperform the initial configuration, reducing interference from 0.4276 to 0.0000.
- The Greedy algorithm, despite its simplicity, performs as well as more advanced heuristics in this scenario.
- DSATUR provides stable and interpretable results, consistent with observations from geometric graph coloring literature.

These results validate the accuracy and relevance of the proposed mathematical and simulation framework, providing a robust foundation for future real-world applications.

6 Conclusion

This work presented a complete simulation-based framework for modeling Wi-Fi channel assignment using interference graphs and graph coloring techniques. By generating a realistic synthetic deployment, constructing the corresponding weighted interference graph, and applying several classical coloring algorithms: Greedy, Welsh–Powell, and DSATUR. We demonstrated the effectiveness of graph-theoretic methods for mitigating radio interference in indoor Wi-Fi environments.

The numerical results show that the initial channel configuration exhibits a significant interference cost of 0.4276, whereas all three algorithms successfully produce a zero-interference assignment. These findings confirm that the simulated deployment is 3-colorable with the available non-overlapping channels and highlight the ability of even simple heuristics to outperform naive configurations.

The proposed methodology offers several advantages: full experimental control, reproducibility, and the possibility to systematically evaluate alternative interference models, weighting schemes, or optimization strategies. It also provides a solid foundation for integrating exact methods such as Integer Linear Programming for moderate network sizes.

However, simulation-based studies inevitably abstract away certain real-world complexities, such as multipath effects, irregular building layouts, hardware heterogeneity, and dynamic user behavior. An important direction for future work is therefore the application of this methodology to empirical data collected from real Wi-Fi deployments. Such an extension would enable the validation of the model under practical conditions and support data-driven optimization of institutional wireless infrastructures.

Overall, this study demonstrates the relevance and power of graph-based optimization techniques for channel allocation and lays the groundwork for a future empirical investigation using real-world measurements.

Disclaimer (Artificial Intelligence)

The author hereby declares that generative artificial intelligence tools were used during the preparation of this manuscript entitled *"Graph-Theoretic Optimization of Wi-Fi Channel Assignment: A Simulation-Based Study"*. AI assistance was strictly limited to language refinement, improvement of academic English, LaTeX formatting, and the structuring of tables, figures, and appendices.

All scientific contributions including the formulation of the interference model, the construction of the simulated Wi-Fi dataset, the development of the mathematical framework, the graph-coloring analysis, the numerical experiments, and the interpretation of results were entirely conceived, verified, and validated by the author.

Details of AI Usage

- **Name / Model / Version:** ChatGPT (GPT–5.1), OpenAI.
- **Source:** https://chat.openai.com
- **Purpose of Use:** Assistance with English grammar, restructuring of paragraphs for clarity, LaTeX formatting, and generation of clean tables and appendix layouts.
- **Examples of Input Prompts:**
  - "Translate my methodology section into academic English while keeping the mathematical structure intact."
  - "Generate a LaTeX table for the simulated interference matrix using booktabs formatting."
  - "Rewrite this paragraph to improve clarity and academic tone."

The author confirms that all AI-assisted outputs were thoroughly reviewed, corrected, and validated to ensure scientific accuracy, originality, reproducibility, and academic integrity.

References

Ahmed, S., Qadir, J., and Yau, K.-L. (2015). Channel assignment algorithms in wifi networks: A survey. *Journal of Network and Computer Applications*, 52:86–105.

Alicherry, M., Bhatia, R., and Li, L. (2005). Joint channel assignment and routing for throughput optimization in wireless mesh networks. In *Proceedings of ACM MobiCom*, pages 58–72.

Bondy, J. A. and Murty, U. S. R. (2008). *Graph Theory*. Springer.

Brélaz, D. (1979). New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256.

Clark, B. N., Colbourn, C. J., and Johnson, D. S. (1990). Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Gast, M. (2012). *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, 2nd edition.

Patro, A. and Das, S. R. (2012). Spectrum allocation in wifi networks. *IEEE Transactions on Mobile Computing*, 12(7):126–139.

Ramachandran, K., Belding, E., Almeroth, K., and Buddhikot, M. (2006). Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proceedings of IEEE INFOCOM*, pages 1–12.

West, D. B. (2001). *Introduction to Graph Theory*. Prentice Hall, 2nd edition.

Wolsey, L. A. (1998). *Integer Programming*. Wiley.

Appendix A: Python code for the simulation and analysis

Listing 1: Python implementation of the Wi-Fi interference simulation and graph coloring analysis

```
import math
import networkx as nx

# ----------------------------
# 1. Simulated access point data
# ----------------------------

aps = [
{"id": "AP1", "x": 5, "y": 4, "power": 18, "band": "2.4GHz", "channel_init": 1, "load": 12},
{"id": "AP2", "x": 10, "y": 6, "power": 18, "band": "2.4GHz", "channel_init": 6, "load": 20},
{"id": "AP3", "x": 15, "y": 5, "power": 18, "band": "2.4GHz", "channel_init": 11, "load": 18},
{"id": "AP4", "x": 22, "y": 4, "power": 18, "band": "2.4GHz", "channel_init": 1, "load": 25},
{"id": "AP5", "x": 28, "y": 6, "power": 18, "band": "2.4GHz", "channel_init": 6, "load": 15},
{"id": "AP6", "x": 34, "y": 5, "power": 18, "band": "2.4GHz", "channel_init": 11, "load": 10},
{"id": "AP7", "x": 8, "y": 15, "power": 18, "band": "2.4GHz", "channel_init": 1, "load": 30},
{"id": "AP8", "x": 14, "y": 16, "power": 18, "band": "2.4GHz", "channel_init": 6, "load": 22},
{"id": "AP9", "x": 20, "y": 15, "power": 18, "band": "2.4GHz", "channel_init": 11, "load": 17},
{"id": "AP10", "x": 26, "y": 16, "power": 18, "band": "2.4GHz", "channel_init": 1, "load": 19},
{"id": "AP11", "x": 32, "y": 15, "power": 18, "band": "2.4GHz", "channel_init": 6, "load": 21},
{"id": "AP12", "x": 38, "y": 16, "power": 18, "band": "2.4GHz", "channel_init": 11, "load": 14},
]

# Interference radius (meters)
R = 10.0

```

```python
# Helper to compute Euclidean distance
def distance(ap_i, ap_j):
return math.sqrt((ap_i["x"] - ap_j["x"])**2 + (ap_i["y"] - ap_j["y"])**2)

# ----------------------------
# 2. Build interference graph
# ----------------------------

G = nx.Graph()

# Add nodes
for ap in aps:
G.add_node(ap["id"], **ap)

# Add edges if distance <= R
edges_with_raw_weight = []
for i in range(len(aps)):
for j in range(i + 1, len(aps)):
ap_i = aps[i]
ap_j = aps[j]
d = distance(ap_i, ap_j)
if d <= R:
# Load-distance model
w_raw = (ap_i["load"] + ap_j["load"]) / d
edges_with_raw_weight.append((ap_i["id"], ap_j["id"], w_raw))

# Normalize weights in [0,1]
max_w = max(w for (_, _, w) in edges_with_raw_weight)
for u, v, w_raw in edges_with_raw_weight:
w_norm = w_raw / max_w
G.add_edge(u, v, weight=w_norm)

# ----------------------------
# 3. Interference cost function
# ----------------------------

def interference_cost(G, coloring):
cost = 0.0
for u, v, data in G.edges(data=True):
w = data.get("weight", 0.0)
if coloring[u] == coloring[v]:
cost += w
return cost

# ----------------------------
# 4. Initial configuration
```

```
# ----------------------------

initial_coloring = {ap["id"]: ap["channel_init"] for ap in aps}
initial_cost = interference_cost(G, initial_coloring)

print("=== Interference cost (lower is better) ===")
print(f"Initial configuration: {initial_cost:.4f}")

# ----------------------------
# 5. Coloring algorithms
# ----------------------------

CHANNELS = [1, 6, 11]

def map_colors_to_channels(coloring_indices):
"""Map abstract color indices (0,1,2,...) to Wi-Fi channels {1,6,11}."""
mapping = {}
channels = {}
next_idx = 0
for node, col in coloring_indices.items():
if col not in mapping:
mapping[col] = CHANNELS[next_idx]
next_idx += 1
channels[node] = mapping[col]
return channels

# Greedy coloring (arbitrary order)
coloring_greedy_idx = nx.greedy_color(G, strategy="random_sequential")
coloring_greedy = map_colors_to_channels(coloring_greedy_idx)
cost_greedy = interference_cost(G, coloring_greedy)

# Welsh-Powell (largest-first)
coloring_wp_idx = nx.greedy_color(G, strategy="largest_first")
coloring_wp = map_colors_to_channels(coloring_wp_idx)
cost_wp = interference_cost(G, coloring_wp)

# DSATUR
coloring_dsat_idx = nx.greedy_color(G, strategy="DSATUR")
coloring_dsat = map_colors_to_channels(coloring_dsat_idx)
cost_dsat = interference_cost(G, coloring_dsat)

print(f"Greedy        : {cost_greedy:.4f}")
print(f"Welsh-Powell  : {cost_wp:.4f}")
print(f"DSATUR        : {cost_dsat:.4f}")
```

```
        print("\n===␣Initial␣coloring␣===")
        print(initial_coloring)

        print("\n===␣Greedy␣coloring␣===")
        print(coloring_greedy)

        print("\n===␣Welsh-Powell␣coloring␣===")
        print(coloring_wp)

        print("\n===␣DSATUR␣coloring␣===")
        print(coloring_dsat)
```