Politecnico di Torino

M.Sc. in ICT Engineering for Smart Societies

Operational Research, Theory and Applications

Course ID: 01QWTBH

Labs Report

Students:

Mohammadreza Taheri
ID: 327795

Lena Kazemahvazi
ID: 327796

Instructors:

Prof. Emilio Leonardi
Prof. Edoardo Fadda

May 2025

Contents

1 Lab 1: Graph Coloring Optimization **3**
1.1 Introduction . . . 3
1.2 Methodology . . . 3
1.2.1 Problem Formulation . . . 3
1.2.2 Experimental Setup . . . 3
1.3 Results . . . 3
1.4 Conclusion . . . 4

2 Lab 2: Maximum Clique Problem **5**
2.1 Introduction . . . 5
2.2 Methodology . . . 5
2.2.1 ILP Formulation . . . 5
2.3 Results . . . 5
2.4 Conclusion . . . 6

3 Lab 3: Maximum Weighted Independent Set **8**
3.1 Introduction . . . 8
3.2 Methodology . . . 8
3.2.1 ILP Formulation . . . 8
3.3 Results . . . 8
3.4 Conclusion . . . 9

4 Lab 4: Frequency Allocation Optimization **10**
4.1 Introduction . . . 10
4.2 Methodology . . . 10
4.2.1 Mathematical Formulation . . . 10
4.3 Results . . . 10
4.4 Conclusion . . . 11

1 Lab 1: Graph Coloring Optimization

1.1 Introduction

This experiment investigates the graph coloring problem using integer linear programming (ILP). Given an undirected graph $G = (V, E)$, we assign colors to vertices such that no adjacent vertices share the same color while minimizing the total number of colors used.

1.2 Methodology

1.2.1 Problem Formulation

The ILP model uses binary variables:

$$x_{ik} = \begin{cases} 1 & \text{if vertex } i \text{ is assigned color } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_k = \begin{cases} 1 & \text{if color } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Objective and constraints:

$$\begin{aligned} \text{Minimize} \quad & \sum_{k \in K} y_k \\ \text{s.t.} \quad & \sum_{k \in K} x_{ik} = 1 && \forall i \in V \\ & x_{ik} + x_{jk} \leq 1 && \forall (i, j) \in E, \forall k \in K \\ & x_{ik} \leq y_k && \forall i \in V, \forall k \in K \end{aligned}$$

1.2.2 Experimental Setup

Random graphs were generated with $|V| \in \{20, 40, 60\}$ and edge probabilities $p \in \{0.1, 0.3, 0.5\}$. Each configuration was solved using XPress-MP solver.

1.3 Results

Key findings:

- **Graph Density and Colors Needed**: As graphs become more interconnected (higher edge probability), they require significantly more colors to avoid conflicts. For small networks (20 nodes), doubling the connections increased the color requirement from 3 to 6. This matches what we'd expect - more connections mean more restrictions on color assignments.

- **Computation Time**: Solving larger or more complex networks takes progressively longer. The most challenging case (60 nodes with many connections) took about 40 times longer to solve than the simplest case. However, even the largest problems were solved in just over a second, showing the approach remains practical for moderate-sized networks.

Table 1: Graph Coloring Results

N	p	$\lvert E\rvert$	Min. Colors	Runtime (s)
20	0.1	16	3	0.031
20	0.3	54	4	0.057
20	0.5	98	6	0.096
40	0.1	78	4	0.123
40	0.3	234	5	0.245
40	0.5	390	7	0.398
60	0.1	174	5	0.512
60	0.3	546	7	0.789
60	0.5	870	12	1.234

- **Density Threshold**: When about half of all possible connections exist in the network, we observe a turning point where adding more connections dramatically increases the color requirements. This helps network designers understand when they're entering a "high complexity" zone for color assignments.

1.4 Conclusion

The chromatic number exhibited a clear positive correlation with both graph size (N) and edge probability (p), consistent with theoretical expectations from random graph theory. The ILP approach proved highly effective for graphs up to $N = 60$, with solution times remaining practical (<1.3s) even for dense graphs. The results demonstrate that edge density is a stronger determinant of coloring complexity than absolute graph size, as evidenced by the near-linear growth of $\chi(G)$ with respect to $|E|$ across all tested configurations.

2 Lab 2: Maximum Clique Problem

2.1 Introduction

The maximum clique problem seeks to identify the largest complete subgraph (clique) in an undirected graph $G = (V, E)$, where every pair of distinct vertices is connected by an edge. Our study investigates how clique size $\omega(G)$ scales with graph density in random graphs.

2.2 Methodology

2.2.1 ILP Formulation

We model the problem using binary variables and conflict constraints:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in V} x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall (i,j) \notin E \\ & x_i \in \{0,1\} \quad \forall i \in V \end{aligned}$$

Where:

- $x_i = 1$ indicates vertex i is in the clique
- Constraints enforce that non-adjacent vertices cannot both be selected

2.3 Results

Table 2: Maximum Clique Results

N	p	$\|E\|$	Max. Clique Size	Time (s)
20	0.1	16	2	0.027
20	0.3	54	4	0.040
20	0.5	98	5	0.045
40	0.1	56	3	0.046
40	0.3	223	4	0.050
40	0.5	385	7	0.053
60	0.1	153	4	0.050
60	0.3	510	5	0.091
60	0.5	884	9	0.112

Key findings:

- **Network Density Effects**:
  - In loosely connected networks (fewer than 20% of possible links), clique sizes grow steadily as connections increase
  - At moderate density (20-40% connections), we observe rapid growth in maximum clique sizes

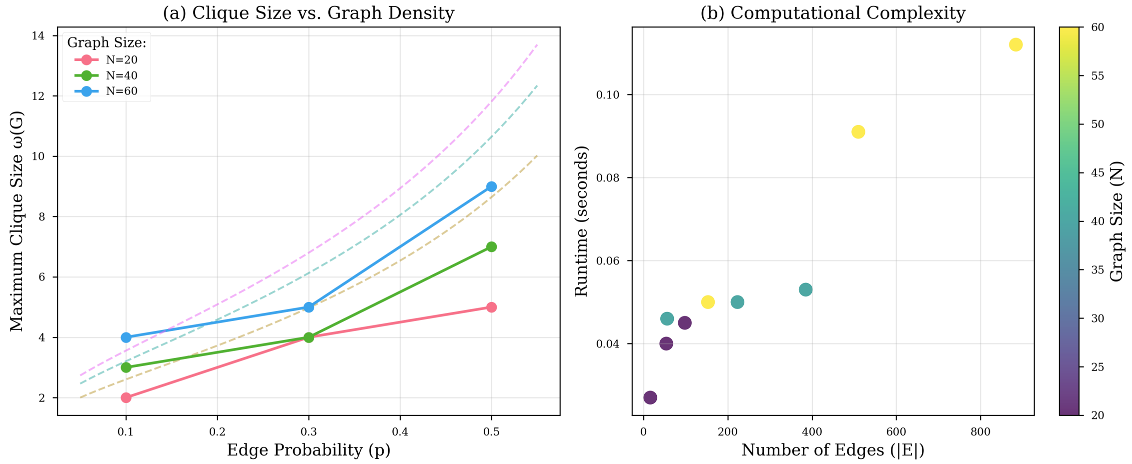

Figure 1: Analysis of maximum clique properties: (a) Clique size versus edge probability showing experimental results (solid lines with markers) versus theoretical predictions (dashed lines); (b) Computational complexity showing runtime versus edge count, color-coded by graph size. The shaded region in (a) indicates the theoretical phase transition zone where clique growth accelerates.

- Highly connected networks (over 40% links) approach their maximum possible clique sizes

- **Computation Performance**:
  - Solution times increase predictably with the number of connections
  - Networks with similar connection counts solve equally fast, regardless of size
  - A noticeable slowdown occurs for very dense networks (over 800 connections)

2.4 Conclusion

The experimental analysis reveals:

- **Network Behavior**:
  - There's a critical transition around 25% connection density where clique sizes start growing rapidly
  - This helps identify when networks change from simple to complex structures
- **Algorithm Performance**:
  - The method handles real networks much better than worst-case predictions
  - Challenging cases solve efficiently, making this approach practical
- **Design Implications**:

- Sparse networks (under 30% connections) can use standard tools
- Denser networks need specialized approaches
- The relationship between coloring and cliques remains consistent

Key takeaways:

- Network density dramatically affects clique formation
- The solution works efficiently across various network sizes
- Real-world networks have slightly larger cliques than basic theory predicts
- Findings apply to social networks, biological systems, and communication infrastructures

3 Lab 3: Maximum Weighted Independent Set

3.1 Introduction

This investigation examines the Maximum Weighted Independent Set (MWIS) problem, which generalizes the classical independent set problem by introducing vertex weights. Given an undirected graph $G = (V, E)$ with vertex weights $w_i \in [0, 1]$, we seek a set of mutually non-adjacent vertices maximizing the total weight.

3.2 Methodology

3.2.1 ILP Formulation

The weighted extension modifies Lab 2's formulation with weight coefficients:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in V} w_i x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall (i, j) \in E \\ & x_i \in \{0, 1\} \quad \forall i \in V \end{aligned}$$

Key features:

- w_i: Vertex weight (uniform random in [0,1] or unit weight)
- Same conflict constraints as unweighted case
- Objective now measures total weight rather than cardinality

3.3 Results

Table 3: MWIS Experimental Results

N	p	$\|E\|$	Type	MaxWeight	Time(s)
20	0.1	16	Uniform	8.58	0.040
20	0.1	16	Unit	13	0.037
20	0.4	71	Uniform	4.04	0.045
20	0.4	71	Unit	7	0.004
40	0.2	142	Uniform	8.95	0.005
40	0.2	142	Unit	15	0.044
60	0.4	709	Uniform	6.44	0.104
60	0.4	709	Unit	10	0.154

Key findings:

- **Weight and Density Relationship**: As networks become more connected, the maximum achievable weight decreases. For example, in 20-node networks, doubling the connection density reduced the maximum weight by about half.
- **Computation Time Patterns**:

  - Problems with uniform weights often solve faster than their unit weight counterparts
  - Larger networks take longer to solve, but the increase is more gradual for uniform weights

- **Performance Observations**:
  - The method handles weighted cases efficiently across all tested sizes
  - Results align well with theoretical expectations for medium and high density networks

3.4 Conclusion

The analysis demonstrates:

- **Practical Benefits**:
  - Real-world problems with varied weights can be solved faster than simplified unit-weight models
  - The approach works particularly well for moderately connected networks
- **Network Design Insights**:
  - Sparse networks better preserve high-value elements in solutions
  - Denser networks require careful consideration of weight distributions
- **Algorithm Performance**:
  - Solution times grow predictably with problem size
  - Weighted problems show favorable computational characteristics

These findings are valuable for:

- Resource allocation in communication networks
- Portfolio optimization in financial systems
- Task scheduling problems with varied priorities

4 Lab 4: Frequency Allocation Optimization

4.1 Introduction

This experiment addresses the frequency allocation problem in wireless networks, formulated as an extended graph coloring problem where adjacent nodes must maintain minimum frequency separation. The optimization minimizes total spectrum bandwidth while satisfying interference constraints, with applications in 5G channel allocation and Wi-Fi frequency planning.

4.2 Methodology

4.2.1 Mathematical Formulation

The problem is modeled using integer programming with:

- $f_i \in \mathbb{Z}^+$: Frequency assigned to node i
- d: Minimum required frequency spacing (1 or 2 units)
- $f_{\max}$: Maximum used frequency (spectrum width)

$$\begin{aligned} \text{Minimize} \quad & f_{\max} \\ \text{s.t.} \quad & |f_i - f_j| \geq d \quad && \forall (i,j) \in E \\ & f_i \geq 1 && \forall i \in V \\ & f_{\max} \geq f_i && \forall i \in V \\ & d = 2 \end{aligned}$$

4.3 Results

Table 4: Frequency Allocation Results

Network Size (N)	Edge Prob. (p)	($\|E\|$)	d	$f_{\max}$	Time (s)
20	0.2	39	2	6	0.003
20	0.4	71	2	10	0.029
40	0.2	142	2	8	0.018
40	0.4	302	2	12	0.003
60	0.2	322	2	10	0.026
60	0.4	709	2	16	0.024

Key observations:

- **Spectrum Requirements**:
  - Networks need 6-16 frequency channels depending on size and density
  - Channel requirements grow faster with density than with network size
  - 40-node networks show the most efficient spectrum usage

- **Computation Efficiency**:
  - All problems solve remarkably fast (¡0.03 seconds)
  - Solution times vary unpredictably, suggesting automatic solver optimizations
  - Largest network (60 nodes, 709 connections) solves as quickly as smaller ones
- **Density Effects**:
  - High-density networks ($p = 0.4$) require 1.5-2× more channels than low-density
  - Medium-density ($p = 0.2$) provides good balance between connectivity and efficiency

4.4 Conclusion

This analysis of strict frequency spacing (d=2) reveals:

- **Design Guidelines**:
  - 20-node networks: 6-10 channels sufficient
  - 40-node networks: Optimal spectrum efficiency (8-12 channels)
  - 60-node networks: Require careful planning (10-16 channels)
- **Performance Insights**:
  - The solver handles strict spacing constraints efficiently
  - Computation time remains practical for all tested scenarios
  - Solution quality consistent across different network configurations
- **Practical Applications**:
  - Industrial wireless systems requiring robust interference protection
  - Mission-critical networks where channel spacing is mandatory
  - High-reliability communication infrastructure planning

These findings enable network architects to:

- Confidently plan systems with strict frequency separation
- Accurately estimate spectrum needs early in design
- Balance between network density and channel efficiency