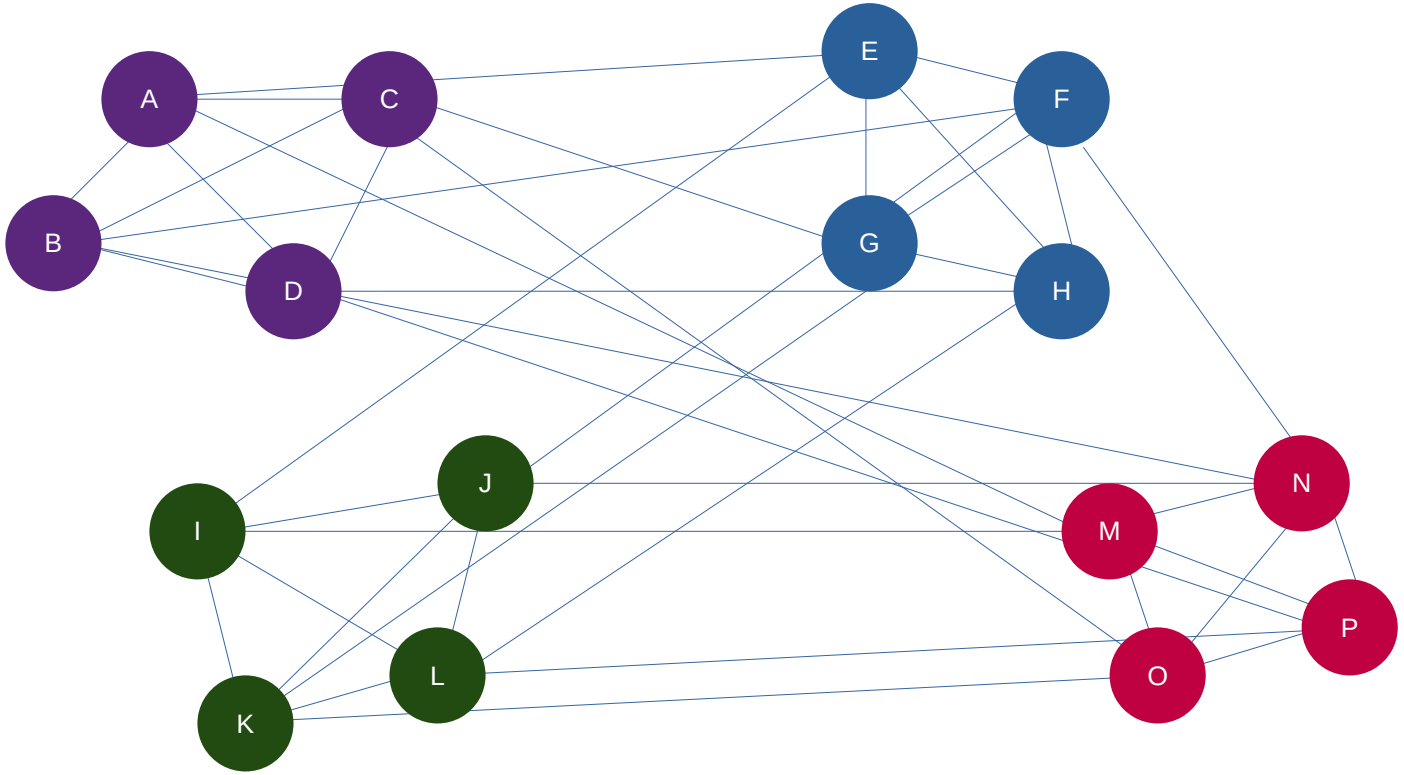
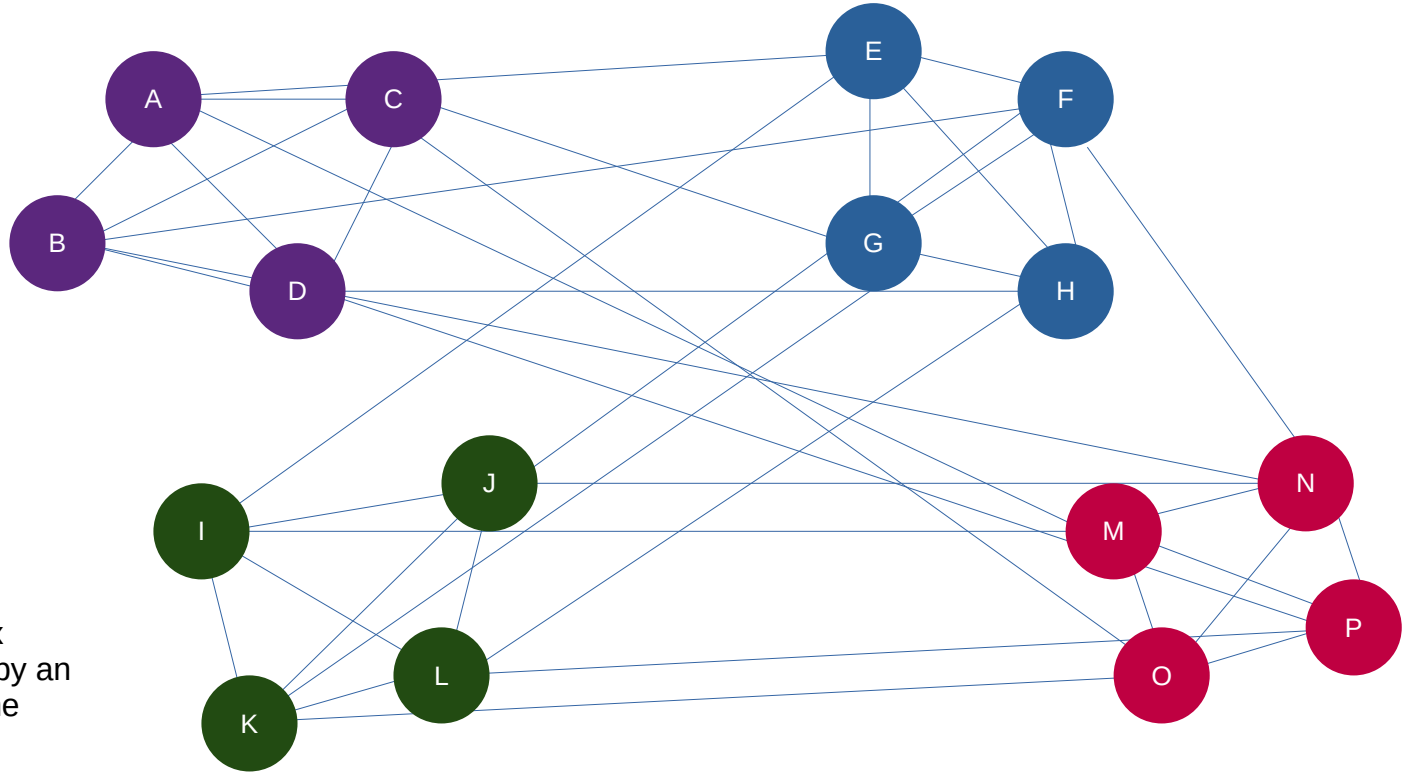
A Multi-Agent Solution for the Graph Coloring Problem

Graph Coloring Problem – Starting Situation



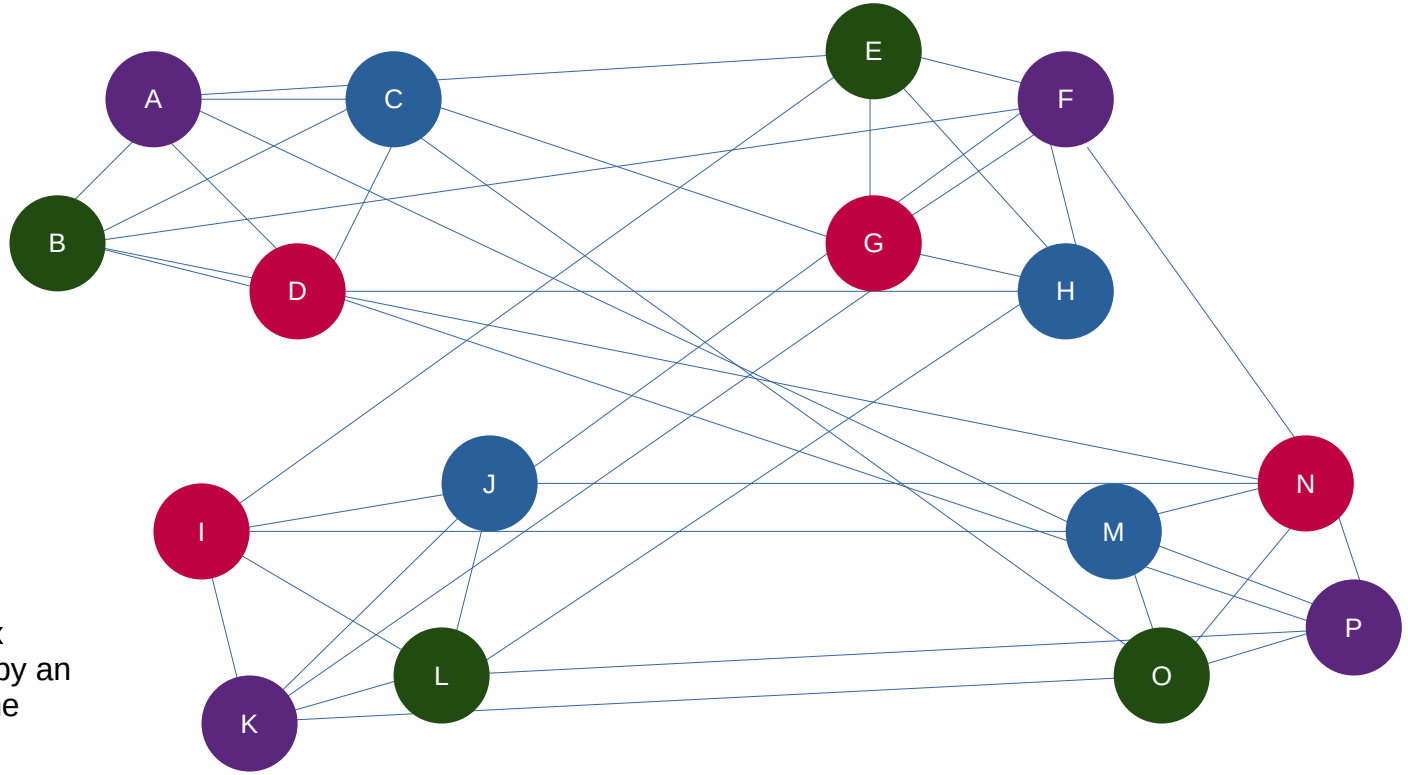
Graph Coloring Problem – Starting Situation



Rules

- One color per vertex
- Vertices connected by an edge may not have the same color

Graph Coloring Problem – Found Solution



Rules

- One color per vertex
- Vertices connected by an edge may not have the same color

Graph Coloring Problem – Multi Agent Setting

Agents

- Node Agents
- Monitoring Agent

Environment

- Vertices
- Edges

Interactions

- Observations
- Messages

(Main)

- Experiment Controller
- Initiates Environment and Agents

Graph Coloring Problem – Agents

Monitoring Agent

- Does NOT coordinate the coloring
- Is NOT involved in the coloring process

- Monitors the system for complete and correct coloring
- Conducts logging to sysout during simulation
- Signals to stop the simulation when complete

Graph Coloring Problem – Agents

Node Agents

- One per vertex
- Autonomous entity responsible for coloring its vertex

- Compatible Goals
- Insufficient Resources
- Insufficient Abilities
-> Coordinated Collaboration

- Beliefs = My and neighbours current colors
- Desires = Conflict-free coloring
- Intentions = Choose color or resolve conflict
-> Simplified BDI

- Stigmergic Coordination

```
18      private final Set<Integer> neighborColors;
19      private int myColor;
```

```
@Override
public void run() {
    while (running) {
        try {
            gatherNeighborColors();

            if (!environment.getNode(agentId).isColored()) {
                chooseColor();
            } else if (environment.hasConflict(agentId)) {
                resolveConflict();
            }

            broadcastMyColor();

            processMessages();
```

Graph Coloring Problem – Interactions

Observations

- Passive observation of neighbor colors

Messages

- Active notification of color changes to neighbors

- Both systems are redundant
- Informative Pattern

Graph Coloring Problem – Environment

Organization

- No hierarchy
- No central authority for coloring decisions
- Flat (all agents equal)

- Roles: Node Agents, Coordinator Agents

- Boundaries: Clear membership, no entry control

Institution

Constitutive Rules:
- If vertex, then Agent
- If edge, then constraint

Regulative Rules:
- No same color for adjacent vertices
- Agents must resolve conflicts

Norms

- Respect neighbors color choices
- Change color if my colors cause conflict
- Broadcast color changes

Live Demonstration