Hands-On Tutorial on Optimization

M. Buchem, N. Megow, A. Lindermayr, Z. Liu
University of Bremen, SoSe 2024

Graph Coloring and Solver Parameter

Day 4

Outlook

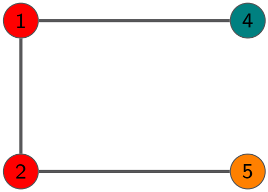
1. The Graph Coloring Problem.
2. Different (I)LP representations for the Graph Coloring Problem.
3. Solving the (I)LPs with different Solver Configurations.
4. (Using different solvers as Backend.)

Graph Coloring

Input: An undirected graph $G = (V, E)$.
Task: Assign a color to each vertex $v \in V$

Graph Coloring

Input: An undirected graph $G = (V, E)$.
Task: Assign a color to each vertex $v \in V$

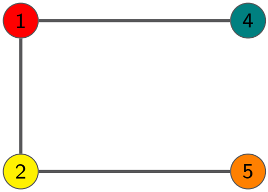


University of Bremen

Graph Coloring

Input: An undirected graph $G = (V, E)$.
Task: Assign a color to each vertex $v \in V$

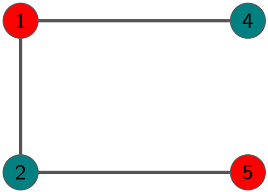
- such that no neighboring vertices have the same color



Graph Coloring

Input: An undirected graph $G = (V, E)$.
Task: Assign a color to each vertex $v \in V$

- such that no neighboring vertices have the same color
- and minimize the number of used colors.

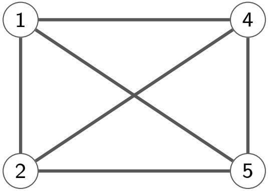


Famous problem (four color theorem), known to be NP-hard.
We consider a special case.

Graph Coloring on Complete Graphs

Input: A complete undirected graph $G = (V, E)$.
Task: Assign a color to each vertex $v \in V$

- such that no neighboring vertices have the same color
- and minimize the number of used colors.

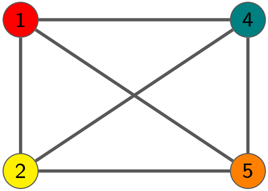


What is the optimal objective value?

Graph Coloring on Complete Graphs

Input: A complete undirected graph $G = (V, E)$.
Task: Assign a color to each vertex $v \in V$

- such that no neighboring vertices have the same color
- and minimize the number of used colors.



What is the optimal objective value? For complete graphs $n = |V|$!

(I)LP Representations

Input: A set of vertices $V = \{1, \ldots, n\}$, a set of colors $C = \{1, \ldots, n\}$.
Task: Assign a unique color to each vertex $v \in V$.
(Minimize the number of used colors.)
How to model this as an (I)LP?

$$x_{v,c} := \begin{cases} 1 & \text{if vertex } v \text{ is colored with } c \\ 0 & \text{otherwise} \end{cases} \qquad y_c := \begin{cases} 1 & \text{some vertex is colored with } c \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{array}{llcll}
\min & \sum_{c \in C} y_c & & & \\
\text{s.t.} & \sum_{c \in C} x_{vc} & = & 1 & \forall v \in V \\
& x_{vc} + x_{uc} & \leq & 1 & \forall c \in C, \forall u, v \in V\colon v \neq u \\
& x_{vc} & \leq & y_c & \forall c \in C, \forall v \in V \\
& x_{vc} & \in & \{0, 1\} & \forall c \in C, \forall v \in V \\
& y_c & \in & \{0, 1\} & \forall c \in C
\end{array}$$

LP Relaxation

The corresponding LP relaxation:

$$x_{v,c} := \begin{cases} 1 & \text{if vertex } v \text{ is colored with } c \\ 0 & \text{otherwise} \end{cases} \qquad y_c := \begin{cases} 1 & \text{some vertex is colored with } c \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{array}{llcll}
\min & \sum_{c \in C} y_c & & & \\
\text{s.t.} & \sum_{c \in C} x_{vc} & = & 1 & \forall v \in V \\
& x_{vc} + x_{uc} & \leq & 1 & \forall c \in C, \forall u, v \in V\colon v \neq u \\
& x_{vc} & \leq & y_c & \forall c \in C, v \in V \\
& 0 \leq x_{vc} & \leq & 1 & \forall c \in C, \forall v \in V \\
& 0 \leq y_c & \leq & 1 & \forall c \in C
\end{array}$$

University of Bremen

Additional Constraints

Suggestions of the course for further constraints to strengthen the model:

$$\sum_{v \in V} x_{vc} \leq 1 \qquad \forall c \in C$$

$$\sum_{v \in V} x_{vc} = y_c \qquad \forall c \in C$$

Alternative Representation

$x_v :=$ Color assigned to vertex v

$$
\begin{array}{llcll}
\min & \max_{v \in V} x_v & & & \\
\text{s.t.} & x_v & \neq & x_u & \forall u, v \in V \colon v \neq u \\
 & x_v & \geq & 1 & \forall v \in V \\
 & x_v & \leq & n & \forall v \in V \\
 & x_v & \in & \mathbb{Z} & \forall v \in V
\end{array}
$$

What is the problem with this formulation?

- A maximum operator in the objective function.
- Pulp does not accept $\neq$-constraints.
- $\neq$-constraints can make a problem non-linear.

There are workaround around these problems (see tricks session on Tuesday)!

Alternative Representation

Getting rid of the max in the objective:

$x_v :=$ Color assigned to vertex v $\quad$ $z :=$ Number of used colors

$$
\begin{array}{llcll}
\min & z & & & \\
\text{s.t.} & x_v & \neq & x_u & \forall u, v \in V \colon v \neq u \\
 & z & \geq & x_v & \forall v \in V \\
 & x_v & \geq & 1 & \forall v \in V \\
 & x_v & \leq & n & \forall v \in V \\
 & x_v & \in & \mathbb{Z} & \forall v \in V
\end{array}
$$

Getting rid of the $\neq$-constraints:

Alternative Representation

$x_v :=$ Color assigned to vertex v $\quad$ $z :=$ Number of used colors

$$y_{vu} := \begin{cases} 1 & \text{if } x_v - x_u < 0 \\ 0 & \text{otherwise} \end{cases}$$

$N :=$ Constant of value $n + 2$

$$\begin{array}{llcll}
\min & z & & & \\
\text{s.t.} & x_v - x_u + y_{vu} \cdot N & \geq & 1 & \forall u, v \in V\colon v \neq u \\
& -(x_v - x_u) + (1 - y_{vu}) \cdot N & \geq & 1 & \forall u, v \in V\colon v \neq u \\
& z & \geq & x_v & \forall v \in V \\
& x_v & \geq & 1 & \forall v \in V \\
& x_v & \leq & n & \forall v \in V \\
& x_v & \in & \mathbb{Z} & \forall v \in V \\
& y_{vu} & \in & \{0, 1\} & \forall v, u \in V
\end{array}$$

Graph Coloring on Complete Graphs

Why do we care about graph coloring on complete graphs?

- Trivial to solve for humans or problem-specific algorithms.
- Not necessarily easy for more general solvers.
- Lots of symmetry and (hidden) structural properties.
- Size of the problem easily scalable.

We use the problem as a benchmark to test the impact of

- different (I)LP formulations,
- solving techniques (cutting planes, symmetry breaking,...)
- integer or no integer variables,
- (different solvers).

Solver Parameter

Solving process of a solver can be configured. Example parameters::

- The number of cutting plane passes.
- How “aggressively” the solver should break symmetries.
- The usage of heuristics.
- The usage of presolve operations.
- …

The concrete parameters are solver-specific.
Pulp provides ways to set them:

- Pass an options list to the solver.
- For Scip: Add a configuration file (`scip.set`) to the directory of your python file.

Conclusion

On StudIP you find:

- Different models for graph coloring on complete graphs and
- a Scip configuration file (`scip.set`).
- A template file if you want to implement the models yourself.

Test how the running time for solving the model changes depending on

- whether you use the ILP representation or LP relaxation,
- the ILP representation you use,
- the solver configuration,
- the solver you use,
- the size of the problem instance,
- ...