Franziska Eberle, Ruben Hoeksma, Nicole Megow **Summer 2019**

Hands-on Tutorial on Optimization

Exercise Sheet: Graph Coloring

Exercise 1 (Model)

Consider the graph coloring problem: Let $G = (V, E)$ be a graph with node set V and edge set E. Let $n = |V|$ be the number of nodes in the graph. Find a coloring of the nodes that minimizes the total number of colors used while no two adjacent nodes, i.e., $v, w \in V$ with $\{v, w\} \in E$, are given the same color.

More formally, find an assignment $c : V \to \{1, \dots, n\}$ such that $c(v) \neq c(w)$ for all $\{v, w\} \in E$ and $|\{k \in \{1, \dots, n\} : c^{-1}(k) \neq \emptyset\}|$ is minimal.

Model this problem as an (I)LP.

Exercise 2 (Implementation)

Assume that $G = K_n$ for $n \in \mathbb{N}$, i.e., G is the complete graph on n nodes.

Implement the LP relaxation of the model for $G = K_n$ in CPLEX. Solve the LP relaxation giving the solver the following options for several values of n.

```
execute PARAMS {
    cplex.preind = 0;
    cplex.prepass = 0;
    cplex.heurfreq = -1;
    cplex.cutpass = -1;
}
```

These comments deactivate some intelligence of the solver such as preprocessing, using heuristics, ...

What is the largest value of n that your model can still solve in reasonable time (say less than 5 minutes)?

Exercise 3 (Integrality)

Now incorporate the integrality conditions for `y(c)` and `x(v,c)` by declaring the variables to be binary with the following code snippet:

```
dvar boolean x[nodes, colors];
dvar boolean y[colors] in 0..1;
```

What is the largest value of n that your model can still solve in reasonable time (say less than 5 minutes)? Can you explain this behavior? Do you have an idea how to strengthen your model?