Decomposition Problems

COMP4691-8691
School of Computer Science, ANU

1 Complicating Variables and Constraints

- Explain what complicating variables and complicating constraints are. Relate them to Column Generation and Benders Decomposition.
- If we are solving an LP, what is the relationship between complicating variables and complicating constraints?

2 Column Generation

2.1 Vertex Coloring

In this question, we will revisit the vertex coloring problem you saw in COMP3620/6320 and solve it using Column Generation.

Given an undirected graph $G = (V, E)$, where V is the set of vertices and E the set of edges, a valid coloring of G assigns a color to all vertices V such that adjacent vertices have different color. More formally, for all edges $(i, j) \in E$, the color of $i \in V$ must be different from the color of $j \in V$. In this question, we want to find the **minimum number of colors** needed to generate a valid coloring of G.

- Formulate the vertex coloring problem as an ILP. Assume you are given a set C of available colors and use two types of variables: y_c to represent whether the color $c \in C$ is used or not; and x_{ic} to represent whether vertex $i \in V$ has color $c \in C$.
- Reformulate the problem as another ILP using a new set of variables

that might be exponentially large (hint: look at the definition of *independent set*). The linear relaxation of this ILP will be our Master Problem.

- What is the Reduced Master Problem?
- What is the formulation of the pricing problem? Do you know the name of the problem being solved by the pricing problem?
- Now you have the complete column generation approach to solve the Master Problem, i.e., the *linear relaxation* of the reformulated problem. Outline how you would continue by implementing a Branch-and-Price algorithm, i.e., combining Branch-and-Bound with column generation.

2.2 Cutting Stock Implementation

Implement the Cutting Stock example from the Column Generation lecture based on the file `cutting_stock.py`. You do not need to implement branch-and-bound, i.e., you only need to solve the linear relaxation of the original problem. For the pricing problem, solve the MIP problem shown in slide 27.

3 Benders Decomposition – Facility Location Implementation

In this question, you will finish modelling the Facility Location problem from the beginning of the Bender's Decomposition lecture and implement a simplified version of it.

- Using the model in slide 5 of lecture 10, write the Bender's Master Problem (BMP).
- The first simplification is that we will work with the (primal) sub-problem (SP) instead of the dual sub-problem (DSP). Formalize the (primal) sub-problem.
- We still need to the get the extreme points of the dual problem, i.e., the $\arg\max$ of the DSP. How can you get it using the SP?
- The second simplification is that we will add a small modification to the BMP to guarantee that all the SPs are feasible; therefore we will not

need to compute extreme rays to generate feasibility cuts. Write a simple constraint using only the y_k variables for the BMP that guarantees feasibility of the SPs. Explain your answer.

- Now you have all the parts to implement the Bender's decomposition approach for the Facility Location problem. Open the file `facility_location.py` and implement the missing methods. Here are some hints:
  - Place all the constraints in a python variable because you will need to access and manipulate them, for instance, `constr = Var('x') >= 10` then `model += constr`.
  - Since we will work on the SP as opposed to the DSP, you will need to change the constraints of the SP. Use the method `changeRHS`, e.g., `constr.changeRHS(-1)`.
  - To get the value of a dual variable associated to a constraint, use the attribute `pi`, e.g., `constr.pi`