Homework assignment

Combinatorial Optimization (LINMA2450)

Part 1 - Boards cutting and delivery

Responsible of the project
Erwan Meunier

Responsibles of the teaching unit
Prof. Geovani Nunes Grapiglia
Prof. Julien Hendrickx

Version as of October 21, 2025

Problem 1 - Optimal boards cutting

A woodworking factory produces large wooden boards of standard length L. The factory receives n customer orders for smaller boards of various lengths l_j, each requiring a certain number of pieces d_j for $j = 1, \dots, n$. The goal is to minimize the number of large boards used while satisfying all customer demands (i.e. wood wastes will be minimized).

Let $P = \{x^t\}_{t=1}^{T}$ be the set of all cutting patterns, where x_j^t is the number of pieces of length l_j in pattern t. Denote X the set of all feasible patterns.

Item j	l_j	d_j
1	2	4
2	2	3
3	3	1

Table 1: Toy example with $L = 5$

1 Model

1. Show that X can be defined by a knapsack type constraint.
2. Suppose that you are given X, write the MILP model which solves the problem of optimally cutting boards. Denote this model P_1.
3. What are the number of variables possible in this model? Is this model tractable?

2 Primal solution

1. Use a greedy heuristic for the knapsack problem to generate a set $\tilde{X}$ of m valid cutting patterns.
2. Show that any solution to P_1 with $\tilde{X} \subset X$ is a primal solution to P_1 computed with the full X.
3. Consider the following algorithm:

   (a) $\tilde{X} \longleftarrow \emptyset$

   (b) While $\left|\tilde{X}\right| < m$ do

      i. Solve P_1 with $\tilde{X}$ and store information in s

      ii. $\tilde{X} \longleftarrow \tilde{X} \cup \{\textsc{NewPattern}(s)\}$

   (c) Return $\tilde{X}$

   Based on information (to be explicited) retrieved at each round define a NewPattern strategy which selects a new "interesting" pattern susceptible of improving the primal bound.

3 Implementation

- Implement the MILP model with Julia / JuMP and find primal solutions to the set of provided instances (in `/BoardsCutting`). Use HiGHs or Gurobi (usually faster) to solve JuMP models. Compare your primal bounds to the optimal ones [1]. Store and present the CPU time to solve each instance and define a time limit *a priori* after which the solving process stops in the case instances are too large.
- Give the optimal solution of the toy example provided in Table 1.

[1] You can retrieve lower and upper bounds on the optimal value by installing the following package: `https://github.com/rafaelmartinelli/BPPLib.jl` and looking at `CSPData.lb` and `CSPData.ub` fields for each instance in `/BoardsCutting`.

Problem 2 - Optimal Delivery Assignment

The wooden boards factory has a set warehouses and a set of clients who must be delivered regularly. Denote by $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of nodes and $E \subseteq V \times V$ a set of edges where $d(e) > 0$ is the length of edge $e = (i, j) \in E$. The set of drivers' / warehouses' positions is denoted by $D \subset V$ while clients' positions are denoted by $C \subset V$ where $|D| \geq |C|$. Non-stated constraints such as capacities or range of vehicles are ignored and supposed satisfied in every case.

Model

1. Model the problem of assigning every client to a warehouse while minimizing the total distance for the delivery.
2. Do some research on the so-called *Hungarian Method* and briefly describe it and in which cases it applies.
3. Justify that the Hungarian Method provides an optimal solution to this minimization problem. Implement the Hungarian Method.

Implementation

1. Implement the MILP model with Julia / JuMP and run it on instances provided in (`/DeliveryAssignment`).
2. Do the same with the Hungarian method.
3. Compare the solving time of the solver to the one of the Hungarian Method for each instance. Discuss the results.

Expected work:

- Your code. Everything must be path-relative and well documented.
- A report answering questions and highlighting important ideas you used to solve the two problems (format PDF).
- Your work will be compressed into a *.zip* file of the following form: `FIRSTNAME1_LASTNAME1_FIRSTNAME2_LASTNAME2.zip` and uploaded on Moodle by the **14th of November, 23:59:59 CET.**