Homework 3.1

10 points. End term: 12-th week (december 18, 2025)

1. (5.5 points) Devise an implemetation (C, C++, C#, Java) for the Branch-and-Bound algorithm using the following pseudo-code:

$\mathscr{S} \leftarrow \varnothing$; // $\mathscr{S}$ *is a stack containing LP problems of the form* $\{objective, constraints\}$.
$\mathcal{P} \leftarrow \{\max \mathbf{c}^T\mathbf{x}, \mathbf{A}\mathbf{x} \leqslant b, \mathbf{x} \geqslant \mathbf{0}\}$;
push($\mathscr{S}, \mathcal{P}$);
$(\overline{\mathrm{x}}, \overline{z}) \leftarrow (\varnothing, -\infty)$; // $\overline{\mathrm{x}}$ *is the best current solution and* $\overline{z}$ *is its corresponding objective value.*
while ($\mathscr{S} \neq \varnothing$) {
  $\mathcal{P} \leftarrow$ **pop**($\mathscr{S}$);
  if ($\mathcal{P}$ is unbounded)
    return "unbounded problem";
  if ($\mathcal{P}$ **is** feasible) { // *otherwise the current problem is fathomed by infeasibility.*
    (x^0, z_0) = **TwoPhaseMethod**($\mathcal{P}$);// x^0 *is the solution of the LP relaxation and* z^0 *is its objective values.*
    if($z_0 > \overline{z}$) // *otherwise the current problem is fathomed by bound.*
      if ($x_i^0 \in \mathbb{Z}$, $\forall\, i \in \mathfrak{I}$) { // *the current problem is fathomed by integrality.*
        $\overline{\mathrm{x}} \leftarrow \mathrm{x}^0$;
        $\overline{z} \leftarrow z_0$;
      }
      else {
        fie $j \in \mathfrak{I}$ **cu** $x_j^0 - [x_j^0] = \max\left\{x_i^0 - [x_i^0]\ :\ i \in \mathfrak{I}\right\}$;
        push($\mathscr{S}, \mathcal{P} \cup \left\{x_j \leqslant [x_j^0]\right\}$);
        push($\mathscr{S}, \mathcal{P} \cup \left\{x_j \geqslant [x_j^0] + 1\right\}$);
      }
  }
}
if ($(\overline{\mathrm{x}}, \overline{z}) = (\varnothing, -\infty)$)
  return "infeasible problem";
return ($\overline{\mathrm{x}}, \overline{z}$);

2. (3 points) Devise an implementation (C, C++, C#, Java) for a version of the Cutting-Plane algorithm using the following pseudo-code:

for ($i = 0$ **to** i_{max}) {
  if ($\mathcal{P}$ **is** feasible) {
    (x^0, f^0) = **TwoPhaseMethod**($\mathcal{P}$);
    if ($\mathbf{x}^0 \in \mathbb{Z}^n$)
      return $\mathbf{x}^0$;
    let $x_i^0 \notin \mathbb{Z}$;
    let $x_i + \sum_{j \in N} \lfloor \tilde{a}_{ij} \rfloor x_j \leqslant \lfloor \tilde{b}_i \rfloor$ **be a Gomory fractional cut;**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{x_i + \sum_{j \in N} \lfloor \tilde{a}_{ij} \rfloor x_j \leqslant \lfloor \tilde{b}_i \rfloor\}$;
  }
  else
    return "infeasible problem";
  $i++$;
}

3. (1.5 points) Run the Branch-and-Bound and the Cutting-Plane implemented algorithms on the following problem:

$$\begin{array}{rcll} \min\ z = \sum_{j=1}^{k} w_j & & & \\ \sum_{j=1}^{n} x_{uj} & = & 1, & \forall u \in V \\ x_{uj} + x_{vj} & \leqslant & w_j, & \forall uv \in E, 1 \leqslant j \leqslant k \\ x_{uj}, w_j \in \{0,1\} & & & \end{array},$$

where k is an upper bound for $\chi(G)$ (it could be n); G is one of the following graphs

- `https://cedric.cnam.fr/~porumbed/graphs/dsjc125.1.col`
- `https://cedric.cnam.fr/~porumbed/graphs/dsjc125.5.col`
- `https://cedric.cnam.fr/~porumbed/graphs/dsjc125.9.col`
- `https://cedric.cnam.fr/~porumbed/graphs/dsjc250.1.col`