Goal of the assignment is to reproduce an existing Mixed Integer Linear Programming (MILP) formulation taken from the literature and perform verification, sensitivity analysis, and validation. As such, the assignment entails the following steps:

- identification of a model of interest from the literature, consistently with the suggestions given in class. For example, a *runway scheduling model*, or a *capacitated vehicle routing problem (CVRP)*. Note that "unconventional" models can be chosen, but they need to be discussed with the lecturers. Students should familiarize with the notation, the objective function, the decision variables, and the constraints
- implementation of the model using any combination of programming language and MILP solver (e.g., Python + Gurobi)
- verification of the implemented model. With verification, it is meant that students should devise test instances to verify that constraints work as they should. For example, if in a runway scheduling model two aircraft occupy simultaneously the same runway, this means that the associated constraint is not properly enforced
- sensitivity analysis addressing the main parameters. This step is important to assess how (in)sensitive the model is with respect to certain parameters. For example, how does the solution change if transportation costs in a CVRP are doubled?
- (OPTIONAL) if applicable, validation of the implemented model. In case no modifications to the model have been added that make the model"unique", some instances taken from the literature can be used as benchmark, and the solution of the same instance using the implemented model should, in principle, be the same in terms of best objective. Note that this might not be the case if
    1. the model taken from the literature is very complex, and hence it was solved using another approach rather than a classic branch & bound (e.g., column generation, branch and price, a meta-heuristic). Implementing such solution techniques is beyond the scope of the assignment
    2. the benchmark model was solved with branch and bound, but allowing a very large computational time. If you use a time-limit, it could be the case that your solution has not converged yet to the optimal value. For example, in a minimization problem your best incumbent can be higher than the optimal solution of the specific instance. What should not happen if that you achieve a best incumbent (again, for a minimization problem), that is lower than the published optimal solution for the instance you are focusing on. In that case, it means you did something wrong, and hence it is suggested to double-check the input data and parameters, the definition of the decision variables and constraints, and to go back to the verification process if needed
- reporting of all the previous steps in a reasonably concise, yet thorough paper. The different criteria that will be graded are: - complexity of the used model (the more challenging the model, the higher the score for this specific criterion), - model description, - verification scenarios, - sensitivity analysis, - readability of the report, and - conclusions