OPTIMIZATION & DECISION 2021/2022
PROJECT

Important notes:

- A **group** is formed by a minimum of **two and a maximum of three students and need to be registered in fenix**.
- The projects (report and necessary data/software) have to be submitted in the course website no later than the **22nd April 2022, 23:59**.
- The submission should be done through a compressed file named: **OD22_<#student1>_<#student2>_<#student3>.**
- All projects must be **presented orally**, and should be based on a slideshow. Each group has **5 min** to present their work. The presentations will take place on the **8th week**, during OD class schedule.
- The **report**, with a maximum of **20 pages**, should formulate the problem to be solved, explain the methods used, and compare the results obtained using proper metrics. At least two methods should be used for each problem, and for groups with three students should be used three methods for each problem. The students must be critical on the used methods and obtained results. Relevant references must be included.
- Each group must select **one** project, which can be from the list below, and **submit its choice in the excel file available in Fenix** (first come first served). Two groups are allowed to be working in the same project if the algorithms used are different and the benchmarks are also different.

List of available projects:

List of available projects:

1. Dual-Resource Constraints in Classical and Flexible Job Shop Problems
2. Dynamic Scheduling
3. Vehicle Routing Problem with Soft Time Windows - Stochastic Travel Times (VRPsTW-STT)
4. Vehicle Routing Problem with Soft Time Windows in GoWizi
5. Traveling Salesman Problem (TSP)
6. Asymmetric traveling salesman problem (ATSP)
7. Sequential ordering problem (SOP)
8. Capacitated vehicle routing problem (CVRP)
9. Knapsack Problem (KP)
10. Multiple Knapsack Problem (MKP)
11. Max Cut Problem (MCP)
12. Quadratic Assignment Problem (QAP)
13. Bin Packing and Cutting Stock Problem
14. Collective Search and Rescue

9. Knapsack Problem (KP)

The **knapsack problem** is a problem in combinatorial. The knapsack problem has been studied for more than a century, with early works dating as far back as 1897. The decision problem form of the knapsack problem (*Can a value of at least* V *be achieved without exceeding the weight* W*?*) is NP-complete, thus there is no known algorithm both correct and fast (polynomial-time) in all cases. There are many variations of the knapsack problem that have arisen from the vast number of applications of the basic problem. The main variations occur by changing the number of some problem parameter such as the number of items, number of objectives, or even the number of knapsacks. Knapsack problems appear in real-world decision-making processes in a wide variety of fields, such as finding the least wasteful way to cut raw materials, selection of investments and portfolios, selection of assets for asset-backed securitization, etc.

Common to all versions are a set of n items, with each item $1 \leq j \leq n$ having an associated profit p_j, and weight w_j. The binary decision variable x_j is used to select the item. The objective is to pick some of the items, with maximal total profit, while obeying that the maximum total weight of the chosen items must not exceed W (the total capacity). Generally, these coefficients are scaled to become integers, and they are almost always assumed to be positive. The knapsack problem in its most basic form:

$$\begin{aligned}
&\text{maximize} && \sum_{j=1}^{n} p_j x_j \\
&\text{subject to} && \sum_{j=1}^{n} w_j x_j \leq W, \\
& && x_j \in \{0,1\} \qquad \forall j \in \{1,\ldots,n\}
\end{aligned}$$

The following site contains single Knapsack and other problems:
https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/knapsack_01.html.

Choose two problems from the database, one low-dimensional and one large-scale problem.

Solve the problems using two or three optimization algorithms (according to the number of students in the group) and compare the results. At least one of the algorithms must be a meta-heuristic. Please note that there are Matlab toolboxes for all metaheuristics.

Support on the project will be given by TAs: Miguel Martins and Bernardo Firme.

10. Multiple Knapsack Problem (MKP)

Consider the description of the knapsack problem n the previous proposal. Alternatively you can choose a multiple Knapsack problem (MKP), which differs from the original KP in the number of available knapsacks with different capacities:
https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_multiple/knapsack_multiple.html.

Choose two problems from the database, one low-dimensional and one large-scale problem.

Solve the problems using two or three optimization algorithms (according to the number of students in the group) and compare the results. At least one of the algorithms must be a meta-heuristic. Please note that there are Matlab toolboxes for all metaheuristics.

Support on the project will be given by TAs: Miguel Martins and Bernardo Firme.