Final Assignment

Heuristic Optimization Methods course (2025-2026)

Optimizing Production Scheduling at Elite Furnishings

In this final assignment, you are tasked with designing and implementing metaheuristic algorithms to optimize production scheduling at Elite Furnishings. Building upon the heuristics you developed/used in your mid-course assignment, the goal of this assignment is to show your knowledge and skills in metaheuristic design, implementation, testing, and reporting.

In this assignment, you are asked to design and implement a metaheuristic algorithm with a set of multiple heuristics/operators (local search/perturbation). You are asked to implement and evaluate 2 different mechanisms described below.

Mechanism 1.

- **M1A: Predetermined order of operators:** a fixed sequence of operators applied during the search process.
- **M1B: Random operator selection:** at each iteration an operator is selected randomly.

Mechanism 2.

- **M2A: Adaptive operator selection:** at each iteration, select an operator considering the feedback from the search on the performance of the operators. Choose a relevant credit assignment method, reward function, and a selection method (as discussed in Lecture 6), and justify your choices.
- **M2B: Q-learning based operator selection:** The idea is to use Q-learning (Lecture 6) to select the best operator at each episode of the search. Each episode is composed of a fixed number of iterations (you may decide the episode length). Choose a relevant state that clearly represents the status of the optimization process, a reward function, and a selection method and justify your choices. This is the more challenging option for mechanism 2 and therefore eligible for a higher subgrade for this component of the assignment.

The final assignment is composed of several tasks given below:

Task 1. Design: There are many different metaheuristics available in the literature for the PFSP. Based on a thorough review of the literature, design an efficient single-solution based metaheuristic for the PFSP with 3 Local search/perturbation operators. Justify your choice for designing this particular metaheuristic based on the literature, characteristics of the problem, and the insights gained from your mid-course assignment. Explain the key steps of the metaheuristic, including each operators used, the list of the parameters, and the operator selection mechanism. Provide references where appropriate.

Task 2. Experimental Design: The main research questions you are supposed to answer are as follows:

- How does the performance of a metaheuristic with multiple operators compare to the performance of a metaheuristic with a single operator in terms of solution quality, computational effort, robustness, and convergence speed?
- How does the performance of Mechanism 2 compare to Mechanism 1 in terms of solution quality, computational effort, robustness, and convergence speed?
- How competitive are your algorithms with multiple operators with respect to the state-of-the-art in terms of solution quality and computational effort?
- How does the problem size affect the performance of your metaheuristic algorithms (you are expected to answer this question by testing your algorithm on different instances as well as determining the worst-case complexity of your algorithm)?

Design relevant experiments to answer the abovementioned research questions. Explain and justify the choices you make for different steps of the experimental design phase (Define Objectives, Select Problem Instances, Select Benchmark Algorithms, Parameter Tuning, Choice of Stopping Criterion, Performance Metrics, etc.). **Note**: when determining your stopping criterion, make sure not to stop your algorithms too early or too late. You can get some ideas on how to set a reasonable stopping criterion by looking at the literature.

Task 3. **Implementation:** Implement your designed algorithm in Python.

Task 4. Execution: Run your algorithm on a set of VRF-hard-large instances contained in *instances.zip*. The *instances.zip* contains an Excel file (*Instances.xlsx)* that list all instances along with their corresponding upper and lower bounds and *.txt* files for each instance. Convert each *.txt* file into a separate worksheet within *Instances.xlsx*. Name each sheet exactly as in the "Instance name" column of *Instances.xlsx*. Read all input data for your runs from the updated *Instances.xlsx* (not directly from the *.txt* files). Run your algorithm 3 times over each instance. For any given instance, do the three runs produce identical results? Explain and justify your answer.

Task 5. Numerical results and reporting: Report and analyse the obtained numerical results and answer each research question separately. Use the requested performance metrics and appropriate tables and graphical tools (visualization) to report the results. You are highly encouraged to use visualization whenever possible. To statistically validate your findings, apply a statistical test like Wilcoxon signed-rank test, if appropriate.

Task 6. Conclusion: Summarize your conclusions based on the analysis and the results obtained and provide insights about the data-driven selection of operators.

Deliverables:

1. **PowerPoint Presentation**: Submit a clear, self-explanatory presentation consisting of no more than 20 slides, including the title slide. Details of the algorithms, supporting code, and results may be included in the appendix, which will not count towards the slide limit.
2. **Codes**: Provide a code script. In providing your code, make sure to make it as general as possible such that a user would be able to run the code without modifications in data path, etc.
3. **Data file**: Submit the *Instances.xlsx* file once you added a sheet for each instance (according to Task 4. Execution).

Make sure to compress these items into a **single** zip file, naming the file with your full name. Submit the zip file via Canvas as a single submission. The deadline for submission is Sunday, October 19th, 2025, by 23:59.

For BDS students:

Please note that for each mechanism (Mechanism 1 and Mechanism 2), you can choose whether you implement A or B. Do not implement both A and B variants.

For EOR students:

EOR students need to answer the same questions as formulated above, but are required to implement and evaluate both the A and B variants of each of the 2 mechanisms. As a result, they also need to extend the analysis of the results in Task 2, by adding 1 or 2 additional questions of their own choosing (which they of course also need to answer in a convincing manner using additional computational experiments or analyses).