Test AMAOR5-B Programming Assignment 2024-2025

Introduction

A paint shop provides services for car parts that have to be painted. Each order corresponds to such a car part and for the upcoming period there is a set of orders available, each with their individual deadline. While it is not impossible to switch between varying colours, it is not desirable as it requires additional work for the cleaning of the machine where the parts are painted. Hence planning orders sequentially in such a way that the colours go from light to dark, or even better remain the same colour, typically results in less setup time. On the other hand the deadlines are what is most important for the company and as such maintaining a low setup time is only desirable if it doesn't result in extra late orders.

The company has several machines available where they do the painting, and hence the setup between different colours only holds for orders processed within the same machine. These machines are fully automated, and it is known that the processing speed of each machine differs. The processing time of each order depends not only on the setup times but also on the size of the order, which is typically measured in the surface that has to be painted. For each machine we know the speed at which they paint a certain surface area, with which the duration in processing for each machine and order can be deduced. Note that if an order is scheduled on a machine it cannot be rescheduled partially.

The data for the paint shop is provided in three separate tables, that can be found in three different sheets of an Excel workbook. The information on the orders in the first table indicate the surface area, colour, deadline, and penalty. The penalty is incurred for each time unit the order is late. In the second table the machine speed, which is the amount of surface area the machine can cover per time unit is indicated. Finally in the last table the setup times between different colours when switching from one colour to another is indicated. You may assume that initially when starting your schedule the machines are clean.

The paint shop regularly has these scheduling issues. They are not looking for a one-off solution, but for a solution method that allows them to properly solve this issue with new data each time. They want a robust solution method that will deliver high quality solutions in any case in a reasonable amount of time. There are several solution methods which can create a high quality schedule for datasets fast: constructive heuristics, improving search heuristics, and meta-heuristics. Based on experimental results, you will need to advice on a proper solution method.

You are given a dataset for September 2024 at the start of this assignment. On **October 7th, 2024**, you will receive a November 2024 dataset. You will design, implement, test your algorithms, and execute experiments on both datasets. Based on your tests and experiments, you select a high quality schedule for November 2024. Apart from the preferred schedule, you will hand in your Python scripts and the rationale of your design decisions, tests and experimental results.

Working methodology

The basis of a mathematical model of this problem will be given on Canvas. You can use this starting point for any necessary model extensions. Make sure your model is generic enough such that the paint shop can also solve future scheduling instances with the model.

Next, you will design algorithms for this optimization model. At least three different solution methods from Rardin (2014) [1]need to be made appropriate for the paint shop problem and implemented, namely

1. a **constructive heuristic** that constructs a feasible schedule from scratch;
2. a simple *first improvement* or *best improvement* heuristic based on **Discrete Improving Search**, that will improve a given feasible schedule to a local optimum;
3. a more complex meta-heuristic, such as **Tabu Search** or **Simulated Annealing**, that will provide a robust solution method for generating a high-quality schedule.

Each algorithm requires design decisions (e.g., which free component to assign a fixed value in the constructive heuristic, which neighbourhood structure for improving search). The more complex algorithms also require decisions on parameter settings and experiments to find out proper settings. You will need to present your algorithm's logic as pseudocode and justify your design decisions in a separate document; information will follow later in this document.

Based on your designs, you will implement your solution methods in **Python**, using a code editor (IDE) like Microsoft VS Code . All the necessary data objects need to be declared. The problem instance, that is, the input data in Excel, needs to be imported. Furthermore, each algorithm needs to be executed for the given parameter settings. Finally, relevant outputs for the schedule need to be computed to compare results for experiments with different solution methods or parameter settings.

Deliverables

The final delivery of the assessment contains the following products:

- All code files of type .py containing the Python script(s) used to generate schedules for the paint shop problem. The script contains the implementations of all solution methods (constructive, simple first/best improvement, meta-heuristic) . The code contains inline comments.

- A well organised and readable pdf document (exported from PowerPoint, LaTeX, Word, etc.) containing the rationale of the design choices made for the heuristics, the pseudocode of the solution methods implemented, the planned tests and experiments, and the experimental results. A template for this rationale document is available from Canvas.

- An Excel file with a high quality schedule for the November 2024 paint shop problem, generated by the Python script. The Excel file should conform to the prescribed format; see the technical requirements below.

- If applicable: any relevant additional Excel files on the basis of which the Python code was tested. The Excel input data files for the September '24 and November '24 paint shop problems are already available and therefore do not need to be delivered.

All digital files should be submitted in one zip file via Canvas.

[1] Ronald R. Rardin (2014), *Optimization in Operations Research*, Pearson; Chapter 12.

Technical Requirements

- Each Python script should correctly import the given Excel file containing the September 2024 and November 2024's problem input data. The format of this Excel file is as follows:
    - One worksheet named 'Orders' with five columns: Order, Surface, Colour, Deadline and Penalty, one worksheet named 'Machines' with two columns: Machine and Speed and a third worksheet named 'Setups' with three columns: 'From colour', 'To colour' and 'Setup time'.
    - First row on each tab contains column names. Each next row contains all the detailed information per tab.
- The Python script should export an Excel file containing a schedule for November 2024. The format of this Excel file should be as follows:
    - One worksheet named 'Schedule' with 11 columns named 'Order','Machine','SeqNo','Setup','Start','Process','End','Deadline','Lateness','Penalty','Cost'.
    - First row contains column names. Followed by a row for each order in the schedule
    - On each order row, all relevant scheduling information of this order is presented. Namely: (A) the order name, (B) the machine on which the order will be executed, (C) the sequence number of the order on the machine, (D) the setup time required before the painting starts, (E) the starting time of the painting, (F) the painting time, (G) the end time of the painting, (H) the order's deadline, (I) the number of time units that the order is completed too late, (J) the cost per time unit too late for this order, and (K) the penalty costs for completing this order too late,

In Python, try to hard-code as little data as possible for the model. Make sure you can import a dataset with any number of orders and machines. Moreover, save any possible solution to the problem (i.e., any schedule of products) generated by one of your solution methods. It may be wise to also note the corresponding objective value.

Possible elaboration

Elaborating on OR techniques can be achieved in different ways. You are free to choose, below are possible research directions:

- More than one neighbourhood structure correctly implemented and compared; quality of solutions found determined
- Or: Efficient and well-readable written code (little to no unnecessary code or calculations), using functions etc., delivered
- Or: Robustness script increased (script directly applicable to input data other than September 2023 and December 2023 without adjustments)
- Or: ...

Choose one research direction and pursue it wholeheartedly to ensure thorough and meaningful results, rather than exploring multiple paths halfway, which may lead to shallow research outcomes.

Assessment

You may work on this programming assignment with a fellow student, and hand in the work as a duo. Your work is assessed via the pdf document containing the justification of your work.

In this justification, your rationale should be logically structured and linguistically sound. The document should justify design decisions regarding the heuristic solution methods implemented. It

should at least include the logic behind the implemented solution method in the form of pseudocode, the test plan with the tests performed, the experimental setup and the experimental results.

You and your duo partner must submit original, i.e. your own, work. Both the Python script and your document must be written by you. You may consult and spar with fellow students or use Python code from chatbots for inspiration; you must ultimately develop everything independently into a working and tested script and justify your choices.

Assessment takes place within three academic weeks from the time of submission. The Assessment Form AMAOR5-B 24-25 is published on Canvas. Once the grades have been determined, they will be shared with you via Progress. It is assumed that the workload of the assignment is evenly distributed between both duo partners, and that each student can be held accountable for the choices made. In principle, therefore, a group assessment takes place. Although typically, a single grade will be assigned to both duo partners for their collaborative assignment, individual grading may be considered in cases where one student's effort or knowledge significantly differs from their partner's contribution.

Communication

All questions about the assignment and the paint shop problem can be addressed via the Discussion section on Canvas. New information will be shared via an Announcement on Canvas.

Timing

- By **Monday 7th of Oct. at 9 am**, the paint shop data for November 2024 will be on Canvas.
- Deadline for submitting the working version of your rationale document in Canvas: **Monday 7th of Oct. 2024, 11:59 pm**.
- Deadline for submitting the desired deliverables in Canvas: **Mon 21th of Oct. 2024, 11:59 pm**.