IE 504- Heuristic Methods
Fall 2023
Homework #2

Due date: *December 14, Thursday* @ 23:59

Instructions:

- In this homework, you can work <u>alone or in groups of two</u>. One submission per team is sufficient.
- Please submit your assignment (the report, the codes, input files, important references, etc.) as a single .zip file through LMS. The report file must be named with the last names of the students.
- Present all codes in the Appendix of your report. Alternatively, provide your project files in an easily accessible format.
- It is critical to list the steps of your heuristics clearly in your report. Students often do hard work, but do not present the methods in an organized and understandable way, which leads to a reduction in grades. Please pay attention to this.
- Present solutions obtained by each method clearly (e.g., schedules, job assignments, completion times, etc.). Please do not provide screenshots of your codes (e.g., from the console) to present solutions.

****

Answer **Questions 1-3)** by addressing the following scheduling problem.

Problem. Consider a parallel machine scheduling problem that aims to schedule *n* jobs on *m* parallel machines. Each job *i* is characterized by its processing time on machine *j*, p_{ij}, its due date for completion d_i, and the unit penalty for tardiness u_i. The unit penalty reflects the importance of the job (e.g., the criticality of the customer, the urgency of the order); a higher penalty indicates higher importance. The process times of jobs across machines are different due to operators' skills and technological differences.

Moreover, jobs are grouped into families, *f*, and setup times are required between jobs belonging to different families. Assume that the setup time required on machine *j* for family *f* is represented by s_{jf}. Assume that the setup time occurs at the beginning of the planning horizon, before the first job, and when a job from a different family is scheduled on the same machine.

Each job can be processed on at most one machine at a time. Moreover, preemption is not allowed; that is, the processing of a job cannot be interrupted and resumed at a later time.

For a given schedule, C_i denotes the completion of the job *i*. The problem is to find a processing order of the jobs that minimizes the total weighted tardiness: $\sum_i u_i \times T_i$, where $T_i = \max(0, C_i - d_i)$ represents the tardiness for the job *i.*

Data for <u>two example problem instances</u> are provided in **HW2_instances.xls.**

Question 1) (30 points)

a. Develop an initial solution for this problem based on a constructive heuristic (CH). Provide the steps of the CH. Present your code.

b. Develop a deterministic local search (LS) algorithm that accepts the CH's solution and sequentially applies two neighborhood moves in each iteration of the LS. Provide the steps of the LS. Present your code.

c. Solve the problem instances by using CH and the LS algorithms. Apply the LS algorithm by 250 iterations. Report the results in a table (initial solution, final solution) and present your observations on the results related to solution quality and time.

Question 2) (40 points)

a. Develop a **Simulated Annealing (SA)** algorithm for the problem. That is, the LS algorithm you designed in Question 1 must be put in an SA framework.

   Set the SA algorithm parameters as below.
   - Initial temperature = 5000;
   - A geometric cooling schedule with a cooling rate of 0.95;
   - Epoch length: 2 iterations;
   - Algorithm termination criteria: The algorithm should stop when the temperature achieves a small value (i.e., 0.01).

   Present your code.

b. Apply the SA algorithm to solve the given problem instances. Note that the SA algorithm is a stochastic algorithm; therefore, in order to evaluate its performance, you need to make multiple runs. Specifically, run the algorithm 3 times for each instance, and evaluate the performance of the algorithm (compared to the solutions found in Question 1) based on its average, worst case, and best-case performances.

c. Repeat part b, with different algorithm parameter settings; specifically, set the initial temperature to 50,000 and cooling rate to 0.99 (keep everything else the same). Run the algorithm 3 times for each instance, present the results, and discuss the quality of the solutions compared to the previous solutions.

d. Discuss your observations related to solution accuracy and time of the SA algorithm. How does the initial temperature affect solution performance? Plot the current and the incumbent values or use other visualizations to support your evaluations.

Question 3) (30 points) Consider the above problem and present a **Genetic Algorithm (GA)** for this problem by answering the following questions. You do not need to code the algorithm, but all steps must be clear.

a. Propose a solution representation for the problem, and illustrate it on an example solution in instance #1.

b. Assume that the population size is 8. Initiate it with the solutions you found in Questions 1 and 2. Specify the fitness values of the solutions and the incumbent solution value.

c. Apply the roulette wheel method for parent selection and select two chromosomes. Specify the probabilities assigned to each chromosome.

d. Describe and illustrate a crossover operator for your solution representation, and generate two offspring. Make sure the offspring are feasible solutions; that is, propose a repair operator if necessary for your proposed crossover operator. Specify the fitness value of both offspring.

e. Describe and illustrate a mutation operator. Propose a repair operator if necessary for this mutation operator. Apply the mutation on the generated offspring in part d. Specify the fitness value of the mutated offspring.

f. Propose and apply a strategy for the generation of the new population. Compare the chromosome values and the incumbent solution value with the previous iteration.