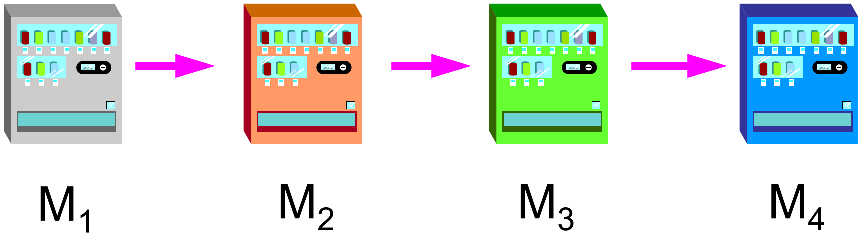
Homework 1

Solving Permutation Flowshop Scheduling Problem using Trajectory-based Metaheuristics

Instructor: Tsung-Che Chiang
tcchiang@ieee.org
Department of Computer Science and Information Engineering
National Taiwan Normal University

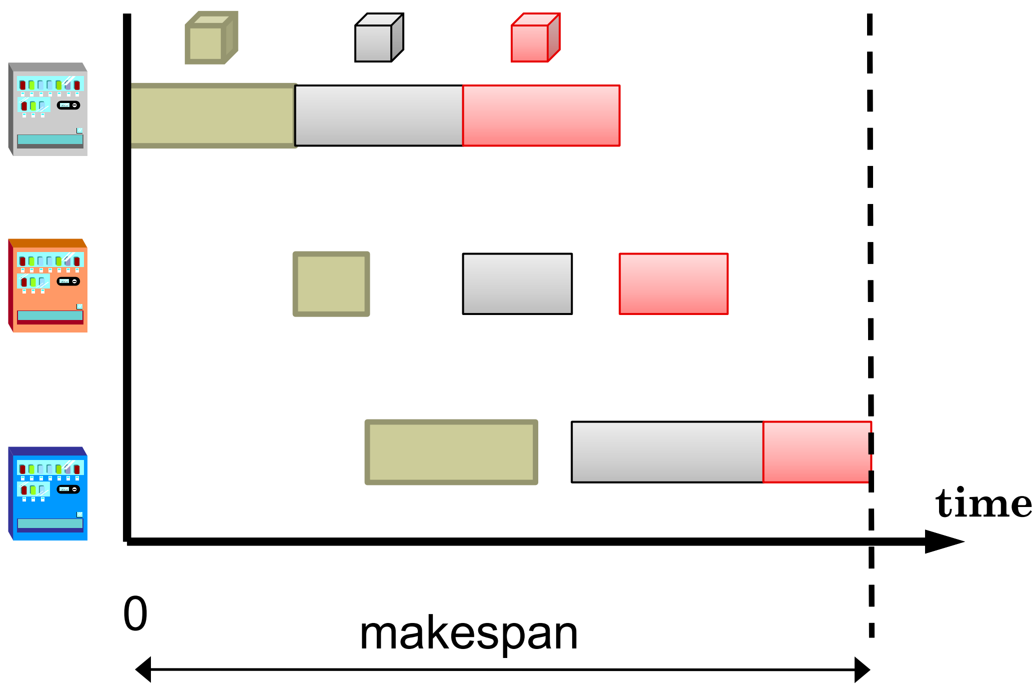
Permutation Flowshop Scheduling

- Problem definition
  - There are m machines and n jobs.
  - Each machine can process only one job at a time.
  - Each job is processed by machine $1, 2, 3, \ldots, m$ in order.
  - Each machine except the first one processes jobs in a FIFO order.



Permutation Flowshop Scheduling

- Objective
  - Find the schedule with the shortest makespan.

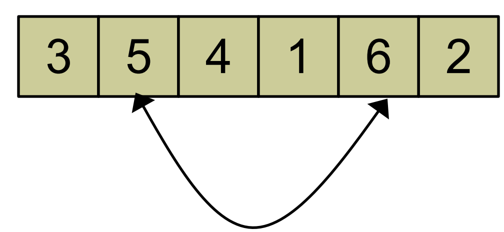


Algorithms

- Tested algorithms
  - iterative improvement
  - simulated annealing
  - tabu search

- Common parts of the tested algorithms
  - permutation-based encoding
    - a string of integers 1, 2, …, n
  - swap neighborhood
    - swap two arbitrary jobs

e.g.



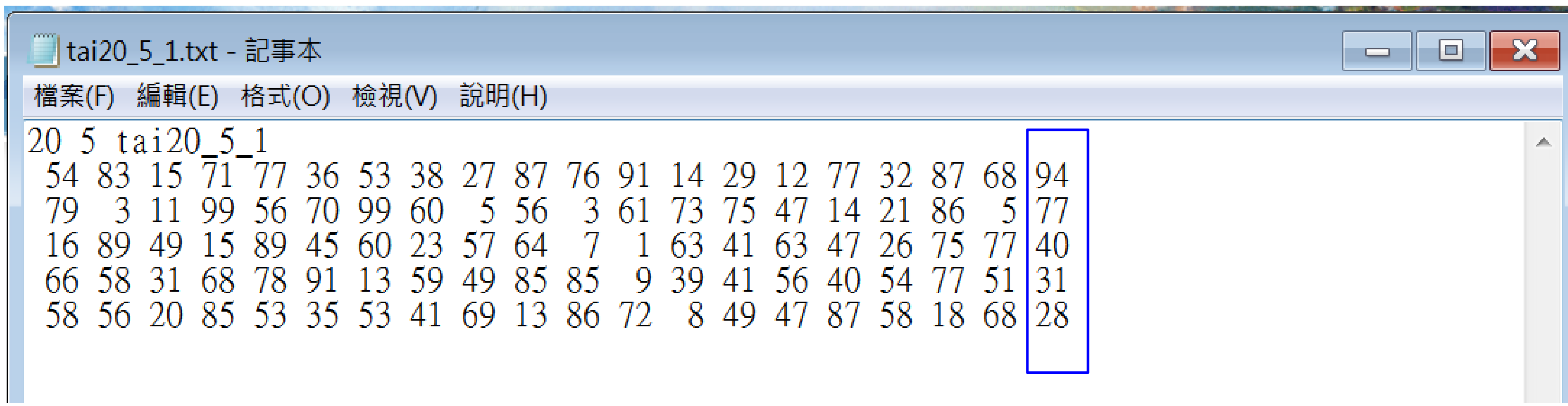
Requirements

- Nine instances from Taillard (1993) will be given on moodle.
  - n = {20, 50, 100} $\times$ m = {5, 10, 20}
- Solve the nine problem instances using three kinds of metaheuristics.
- Each instance should be solved by each algorithm for at least 20 times.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal ofOperational Research*, 64, 278–285.

Requirements

20×5: 20 jobs and 5 machines




```
20 5 tai20_5_1
 54 83 15 71 77 36 53 38 27 87 76 91 14 29 12 77 32 87 68 94
 79  3 11 99 56 70 99 60  5 56  3 61 73 75 47 14 21 86  5 77
 16 89 49 15 89 45 60 23 57 64  7  1 63 41 63 47 26 75 77 40
 66 58 31 68 78 91 13 59 49 85 85  9 39 41 56 40 54 77 51 31
 58 56 20 85 53 35 53 41 69 13 86 72  8 49 47 87 58 18 68 28
```

Processing times of the 20th job on the five machines.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal ofOperational Research*, 64, 278–285.

Documentation

- Algorithm description
  - For each algorithm, you need to give detailed descriptions.
  - For example, the cooling schedule of SA and the tabu structure of TS should certainly be presented clearly.
- Experimental setting
  - The values of parameters (e.g. initial temperature, epoch length, tabu tenure, etc.) in the experiments should be provided.

Documentation

- Experimental results
  - The best-, average-, and worst-case performance over 20 (or more) runs should be given.
  - Computer environment and average computation time should be provided, too.
- Comparison between II, SA, and TS
  - Compare them in terms of solution quality, computational efficiency, simplicity, robustness, etc.

Documentation

- You are encouraged to do more experiments.
  - Difficulty of problem instances
    - Run a random search with equal computational effort and see how much difference between random search and your algorithms is.
    - Run iterative improvement for a large number of times (say 1,000,000) and record the number of local optima and size of basin of attraction.
    - Define a distance between solutions and check if there exists the "big valley" structure.
    - Discuss how the three algorithms perform differently on instances with different sizes.
    - Intensification or diversification? Which is more important?

Documentation

- You are encouraged to do more experiments.
  - Performance analysis
    - Try different setting of cooling schedule and tabu list and see how the performance varies.
    - Examine which parameters are more critical to the algorithm performance.
    - Examine which algorithm is more sensitive to the parameters.
    - Draw the "makespan vs. iteration" plot and see how the algorithms converge and which algorithm converges faster.

Grading

- Correctness
  - Define the problem and objective correctly.
  - Describe the algorithm correctly.
  - Verify your results.
  - Interpret the results and draw the conclusions correctly.
    - Do not make any claim without evidence.
    - Use the experimental results to support your claim.
- Clarity
  - Describe your ideas, algorithms, and experiments in detail.
- Carefulness
  - fonts, notations, figures, tables, references, etc.
- Completeness
  - how much effort you spent on this project

Submission

- Deadline: 2018/3/31 0:00
- The package "MHPS2018-HW1-TeamX.rar" should include
  - a directory called "code," containing the source code.
  - a document called "MHPS2018-HW1-TeamX.pdf."
- Submit the package on moodle by the team member with the smallest ID.