Advanced Models and Methods in Operations Research
Project: Unrelated parallel machines scheduling

Florian Fontan

2021–2022

For each problem considered, instances and a code skeleton containing an instance parser and a solution checker are provided in the `data/` and `python/` folders of the project.

The algorithms must be implemented in the provided files between the tags `TODO START` and `TODO END`.

They must be tested on all the provided instances with the command: `python3 problem.py -i instance.json -c certificate.json`

And each solution file must be validated by the provided checker: `python3 problem.py -a checker -i instance.json -c certificate.json`

The results must be reproducible.

The delivrable must contain:

- A *short* report describing and justifying the proposed algorithms
- The code implementing the algorithms
- The solution files obtained on the provided instances

1 Dynamic Programming

We consider the Single machine order acceptance and scheduling problem with objective Total tardiness:

- Input:
  - n jobs with $(j = 1, \dots, n)$
    - processing time $p_j \in \mathbf{N}^+$
    - due date $d_j \in \mathbf{N}^+$
    - profit $v_j \in \mathbf{N}^+$
- Problem: find a sub-sequence of jobs
- Objective: maximize the total profit minus the total tardiness of the scheduled job

For a job j starting at s, its tardiness is equal to:

$$T_j = \max\{0, (s + p_j) - d_j\}$$

Propose and implement an algorithm based on Dynamic Programming for this problem.

2 Heuristic Tree Search

We consider the Single machine order acceptance and scheduling problem with objective Total weighted tardiness:

- Input:
  - n jobs with $(j = 1, \dots, n)$
    - processing time $p_j \in \mathbf{N}^+$
    - due date $d_j \in \mathbf{N}^+$
    - profit $v_j \in \mathbf{N}^+$
    - weight $w_j \in \mathbf{N}^+$
- Problem: find a sub-sequence of jobs
- Objective: maximize the total profit minus the total weighted tardiness of the scheduled job

For a job j starting at s, its weighted tardiness is equal to $w_j T_j$.

Propose and implement an algorithm based on Heuristic Tree Search with Dynamic Programming for this problem.

3 Column Generation + Dynamic Programming

We consider the Unrelated parallel machine scheduling problem with objective Total tardiness:

- Input:
  - m machines
  - n jobs with $(j = 1, \dots, n)$
    - processing time $p_j^i \in \mathbf{N}^+$ for each machine $i = 1, \dots, m$
    - due date $d_j \in \mathbf{N}^+$
- Problem: find a schedule for each machine such that
  - each job is scheduled exactly once
- Objective: minimize the total tardiness of the schedule

Propose an exponential formulation and implement an algorithm based on a Column Generation heuristic for this problem.

4 Column Generation + Heuristic Tree Search

We consider the Unrelated parallel machine scheduling problem with objective Total weighted tardiness:

- Input:
  - m machines
  - n jobs with $(j = 1, \dots, n)$
    - processing time $p_j^i \in \mathbf{N}^+$ for each machine $i = 1, \dots, m$
    - due date $d_j \in \mathbf{N}^+$
    - weight $w_j \in \mathbf{N}^+$
- Problem: find a schedule for each machine such that
  - each job is scheduled exactly once
- Objective: minimize the total weighted tardiness of the schedule

Propose an exponential formulation and implement an algorithm based on a Column Generation heuristic for this problem.