# Technical Evaluation- Optimization

## Valourec - Operations Research Team

Candidate: Matheus Freire Wu

Campinas, March 2024

## Problem 1: Job scheduling optimization

In this problem, a series of *n* jobs must be scheduled through a single machine. The starting and finishing time must be constrained by an initial release date and a deadline. This case is multi-objective and must reduce order delays and the maximum make-spam. One approach is associating different costs for delay and make-span, but here the problem is solved twice:

1. Reduce and determinate the minimum total delay.
2. Fixing the maximum delay as a new constraint and minimizing the make-spam.

### Problem instance

To test the developed model, a 10 jobs instance was randomly created:

Table 1: Random 10 jobs instance (jobs.csv)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **job** | **process\_time** | **setup\_time** | **release\_time** | **deadline** |
| job\_1 | 10 | 2 | 0 | 15 |
| job\_2 | 5 | 3 | 0 | 20 |
| job\_3 | 15 | 1 | 0 | 30 |
| job\_4 | 18 | 4 | 10 | 40 |
| job\_5 | 4 | 2 | 60 | 90 |
| job\_6 | 6 | 3 | 60 | 90 |
| job\_7 | 9 | 2 | 60 | 80 |
| job\_8 | 20 | 2 | 30 | 60 |
| job\_9 | 21 | 4 | 30 | 70 |
| job\_10 | 8 | 2 | 200 | 70 |

### Sets:

I: previous jobs from *n* include starting condition.

J: current job from *n*

### Variables and constants

Decision variable (binary):

Time variable (real):

Dalay variable (real):

Make-span variable (real):

Process, setup, release and deadline time: , , and .

### Objectives

Reduce the total delay:

Reduce the make-span:

### Constraints

All jobs must be performed once:

Only one job can follow the previous one:

Time sequence:

*M*: arbitrary large value.

Release time:

Deadline time:

Make-span:

Fixed delay (second optimization):

*Td*: total delay from first optimization.

A graph of a job schedule

Description automatically generated with medium confidence

Figure 1. job scheduling result

## Problem 2: Container shipping

The following model is proposed to solve the shipping of pipes from different containers:

### Sets

C: containers

O: orders from each container

P: pipes from each container order

### Variables

Containers:

Pipes: , with *w* weight and *v* volume.

### Objective

There is no objective sense in this problem, it is a case of feasibility.

### Constraints

Total weight and volume sum:

Maximum number of containers:

Only one pipe from each order can be used from each container order:

Container constraints:

1. The chosen containers are listed in *problem2\_output\_a.csv.*

Table 2. Sample from solution a.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Container** | **Sales Order** | **Steel Pipe** | **val** | **weight** | **Volume** |
| B | SO\_1 | 1 | 1 | 977 | 101,14 |
| B | SO\_2 | 1 | 1 | 194 | 24,01 |

1. To check if there is more than one solution, an approach is to add a new constraint to avoid the exact sum of the original solution. This approach is similar to dynamic programming. The results are listed in *problem2\_output\_b.csv.*

Table 3. Sample from solution b.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Container** | **Sales Order** | **Steel Pipe** | **val** | **weight** | **volume** |
| B | SO\_1 | 6 | 1 | 605 | 80,13 |
| B | SO\_2 | 2 | 1 | 106 | 20,78 |

## Problem 3: Parking problem with metaheuristics

GRASP (Greedy randomized adaptive search procedure) methodology is proposed to tackle the parking problem. The core idea is to initialize a series of greed initial solutions with a certain degree of randomness to explore a bigger set of solutions. Then, iterate over a series of steps for local search, by changing cars between lanes until the lanes are equal or the stop condition is reached.

When the limitation of 15 m for a lane is added, the metaheuristics must be adapted in two senses:

1. In constructive step, the algorithm must stop adding cars when the limit is reached.
2. In local search, solutions above the limit are not accepted and the stop condition is reached when a lane is equal to 15 m.