ROADEF Challenge 2020 Abstract

Team ID : J38
Category : Junior
Team menbers : Kosei Ono, Yuta Okamoto

January 3, 2021

1 Method

1.1 Overview

Our method consists of a constructive part and an improvement part.
First, we run a constructive method to get a feasible solution.
Second, We run the improvement method to get a better feasible solution in terms of the objective function value.

1.2 Algorithm of the methods

1.2.1 Constructive method

Our constructive method decides a start time of every intervention according to an order.
The order is a descending order of all the interventions in terms of its average $\Delta_{i,t}$ value over all the time steps T.

Inputs Let $order$ as an order of all the interventions which represents the order in which the start time of every intervention should be decided.

Variables Let variable intervention i as a intervention.
Let variable time step t as a time step of T.
Let $feasible$ as a binary value represents whether all constraints are hold for intervention i at time step t. (All constraints are hold$\Rightarrow 1.$ otherwise 0).
Let start time $start_i$ as a start time of a intervention i.

Output A feasible plan.

Algorithm of the constructive method We show the algorithm of our method below.

<u>**Constructive method**</u>

Step 1. Intervention i := the next intervention in $order$.
If this step is executed for the first time, i := the first intervention in $order$.
If there is no next intervention, Stop.

Step 2. variable time step $t := 0$.

(a) Determine variable $feasible$ of intervention i at time step t.
If $t >$ **intervention** i **' s max time step**, Go to 3

(b) **if variable** $feasible = True$
Define variable intervention i ' s start time $start_i := t$
Go to 1.

else
Time step $t := t + 1$
Go to 2a

Step 3. Intervention i ' s start time $start_i := t_{random}$. The t_{random} is chosen randomly.
Determine the interventions which violate any constraints because of intervention i starting at t_{random}.
And determine the start times of those by applying this our method.
Go to 1.

1.2.2 Improvement method

Our improvement method is a type of a local search.
This method is a first admissible move strategy. As soon as we find a better solution in terms of objective function value in a neighbor of now solution, move to the better solution.
The neighbor is a set of feasible solutions generated by changing an intervention i's start time $start_i$ to $start_i' \in \{1, \ldots, i's\ tmax\} \setminus \{start_i\}$ at random. And the intervention i is chosen randomly too.
We continue doing the procedure above within a time limit.

2 Characteristics of computer used

All programs were run of a macbook pro with a 2.3 GHz 2 cores Intel Core i5 processor and 16 GB 2133 MHz LPDDR3 RAM.
The method just described were coded in C++.