Accurate Tracking Collision Detection and Optimal Scheduling of Airport Ground Support Equipment

Usage instructions for the source data, executable application and experiment results corresponding to the paper are given here

1. Source data

(1) Datasets are generated by the source code "**DataGenerate.java**". For different gate size ranging from 1 to 30, 15 instances are generated, respectively. Therefore, we have 450 instances in total.

(2) Generated Datasets are listed in "**Data.zip**". For example, "**AP3_7.txt**" is the 7th instance with gate size equal to 3.

(3) To illustrate the input format of the data, taking "**AP2_1.txt**" for example, the notation corresponding to the "**Table II**" in the paper are given after each below-listed variable.

```
Unit_Service_Time(min): 3          // c
max_dollies_per_tug: 6           //d
max_tugs_per_flight: 4             //m
max_dollies_at_gate: 30          // n
max_duration(min): 60              //g
speed(km/s): 0.3                   //s
tug_cost(S$/tug): 20.0             //r1
travel_cost(S$/km): 0.05           //r2
Gate_Size: 2
1 2
travel time Matrix:   // \tau_{ij}
0.0 0.5 1.0
0.5 0.0 0.5
1.0 0.5 0.0
Total_Flight_Number: 14
Gate_ID: 1 Flight_ID: 1 Time_Window: 109 209 Pickup_Baggage: 0 Delivery_Baggage: 9
Gate_ID: 1 Flight_ID: 2 Time_Window: 252 337 Pickup_Baggage: 0 Delivery_Baggage: 8
Gate_ID: 1 Flight_ID: 3 Time_Window: 438 538 Pickup_Baggage: 8 Delivery_Baggage: 0
Gate_ID: 1 Flight_ID: 4 Time_Window: 601 707 Pickup_Baggage: 9 Delivery_Baggage: 10
Gate_ID: 1 Flight_ID: 5 Time_Window: 775 878 Pickup_Baggage: 10 Delivery_Baggage: 0
Gate_ID: 1 Flight_ID: 6 Time_Window: 898 1008 Pickup_Baggage: 8 Delivery_Baggage: 5
Gate_ID: 1 Flight_ID: 7 Time_Window: 1127 1201 Pickup_Baggage: 5 Delivery_Baggage: 0
Gate_ID: 2 Flight_ID: 8 Time_Window: 57 148 Pickup_Baggage: 0 Delivery_Baggage: 8
Gate_ID: 2 Flight_ID: 9 Time_Window: 226 322 Pickup_Baggage: 8 Delivery_Baggage: 8
Gate_ID: 2 Flight_ID: 10 Time_Window: 411 481 Pickup_Baggage: 7 Delivery_Baggage: 4
Gate_ID: 2 Flight_ID: 11 Time_Window: 552 631 Pickup_Baggage: 5 Delivery_Baggage: 0
Gate_ID: 2 Flight_ID: 12 Time_Window: 720 771 Pickup_Baggage: 4 Delivery_Baggage: 6
Gate_ID: 2 Flight_ID: 13 Time_Window: 853 956 Pickup_Baggage: 9 Delivery_Baggage: 9
Gate_ID: 2 Flight_ID: 14 Time_Window: 1038 1122 Pickup_Baggage: 0 Delivery_Baggage: 6
```

2. Executable application

- "**Data.zip**" should be unzip and be put in the same directory with "**Javacode.jar**" at first. Then, "**Javacode.jar**" can run with JDK 10 or higher version using command "**java -jar Javacode.jar**".
- Each input instance corresponds to one result file. For example, "**AP2_1.txt**" corresponds to "**result_AP2_1.txt**" .
- Upper machine running time limit for each instance is already set to be 10 minutes.

3. How to read the detailed experiment results

Each result file includes two parts.

The first part is displayed from the perspective of tugs:

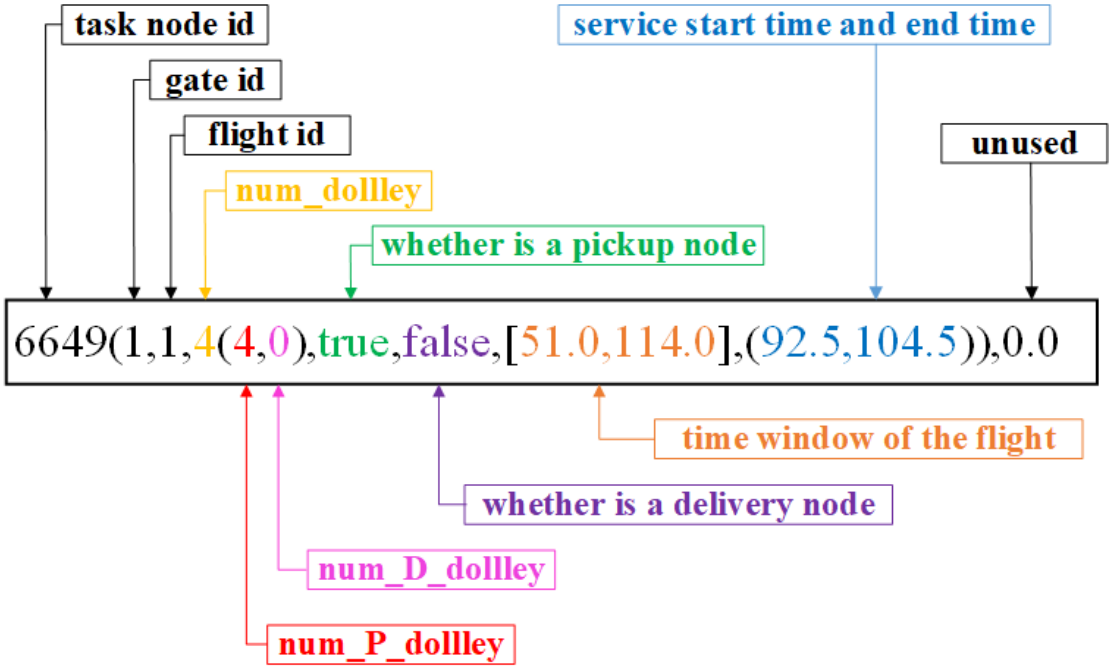
- "Number of Journeys =5" means that 5 tugs in total are used for this data instance.
- For **each tug**, multiple trips are listed in increasing order of start time.
- For **each trip**, information about starting (logical) depot, the sequence of task nodes and destination (logical) depot is showed. Taking a trip showed below for example, this is the **9th trip** of a tug. This trip travels through 2 task nodes starting from 843.5min , back at 884.5min and finish unloading baggage at 899.5min. At the **starting depot** (task node id is 1921, "task node id" is used in the code), the tug takes 12 mins from 843.5min to 855.5min to load 4 dollies. Then, it first travels to **flight 6** (task node id is 6664) at gate 1 to deliver 4 dollies , afterwards travels to **flight 15** (task node id is 6655) at gate 2 to pick up 5 dollies, and finally goes back to the depot. "**num_dolley**" means the number of dollies picked-up/delivered at the task node, "**num_D_after**" means number of dollies attached to the tug which will be delivered to specific flights, "**num_P_after**" means number of dollies attached to the tug which were picked up from some flights and will be taken back to the depot. Please **ignore** "num_pick" and "num_delivery" here (they are defined but unused).

  ```
  --Trip:9  num_nodes=2  starttime=843.5  backtime=884.5  endtime=899.5
    Depot:1921  starttime=843.5  endtime=855.5  num_D_after=4
    Delivery:
    Node:1  Flight:6(6664)  Gate:1  arrivaltime=856.0  starttime=856.0
  endtime868.0  num_dolley=4  num_pick=0  num_delivery=0  num_P_after=0
  num_D_after=0
    Pickup:
    Node:2  Flight:15(6655)  Gate:2  arrivaltime=868.5  starttime=868.5
  endtime883.5  num_dolley=5  num_pick=0  num_delivery=0  num_P_after=5
  num_D_after=0
    Depot:2446  starttime=884.5  endtime=899.5  num_dolley=5
  ```

The second is displayed from the perspective of flights:

- For each flight, it shows the information of task nodes after split procedure.
- Taking a flight showed below for example, this flight has **2 pickup task nodes** and **1 deliver task node**.

  The detailed information of one task node is illustrated in the Fig.



- **Note**: that we can use the “task node id” to connect the tugs' activities and their host flight together.

> Flight:1
>
> --pick_nodes: :2
> 6649(1,1,4(4,0),true,false,[51.0,114.0],(92.5,104.5)),0.0
> 98960(1,1,1(4,2),true,false,[51.0,114.0],(101.5,104.5)),0.0
>
> --delivery_nodes: :1
> 98961(1,1,2(4,0),false,true,[51.0,114.0],(104.5,110.5)),0.0