The Flying Sidekick Traveling Salesman Problem with Recourse

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Mehmet Kurt

The Flying Sidekick Traveling Salesman Problem with Recourse

- Our study is an extension of the Flying Sidekick Traveling Salesman Problem (FSTSP) of (Dell'Amico et al., 2022).
- In this study, as with conventional FSTSP, the drone and truck must operate synchronously.
- However, additionally, our model incorporates a recourse function that addresses drone disruption in the event of bad weather or other unforeseen issues.

The Flying Sidekick Traveling Salesman Problem with Recourse

- Truck and drone delivery models have been extensively studied
- Although weather conditions significantly impact drone delivery operations, in literature, they are mostly assumed to be unchanging
- From a practical point of view, this is not realistic and leads to incorrect planning.
- Some studies take into account weather effects on drone delivery operations but these studies are limited, and they primarily benefit from historical data to develop good predictions of future weather conditions.

The Flying Sidekick Traveling Salesman Problem with Recourse

- Our modeling approach is; at the start of the delivery plan, drone and truck routes are determined considering that weather conditions will not change and will not adversely impact the drone delivery.
- While delivery is in progress, at certain times, disruptions are assumed to happen.

The Flying Sidekick Traveling Salesman Problem with Recourse

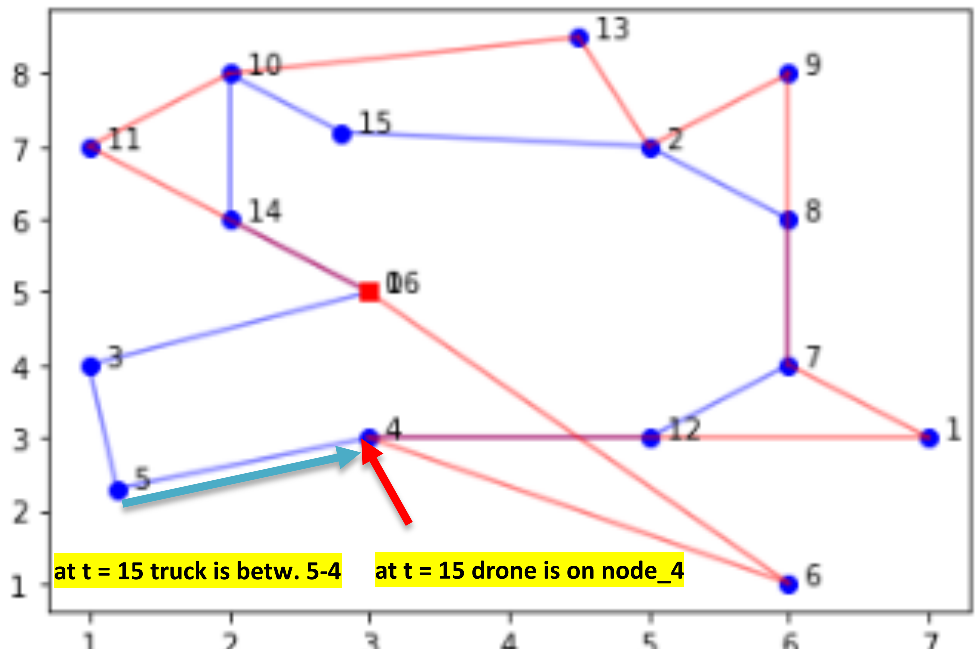
- Then, to handle the disruptions, we propose a recourse.
- The model is structured as a two-stage decision process:
  - the first stage works out the routes for the drone and the truck
  - the second stage determines the new routes for the remaining nodes (customers) in the absence of the drone for the disruption duration.
- The objective function of this study is to minimize the completion time of all deliveries.

The Flying Sidekick Traveling Salesman Problem with Recourse

- The contribution of this study:
  - proposes a new recourse approach in case of adverse weather conditions and unforeseen events,
  - explores two potential outcomes of drone disruption,
    - disruption starts at a particular time and impacts the whole planning area for a specific period (common disruption),
    - disruption starts at a specific time and moves over the region in a rolling fashion (rolling disruption)

Optimum Solution 15 - customer instance (v15)

Optimum solution (INITIAL)

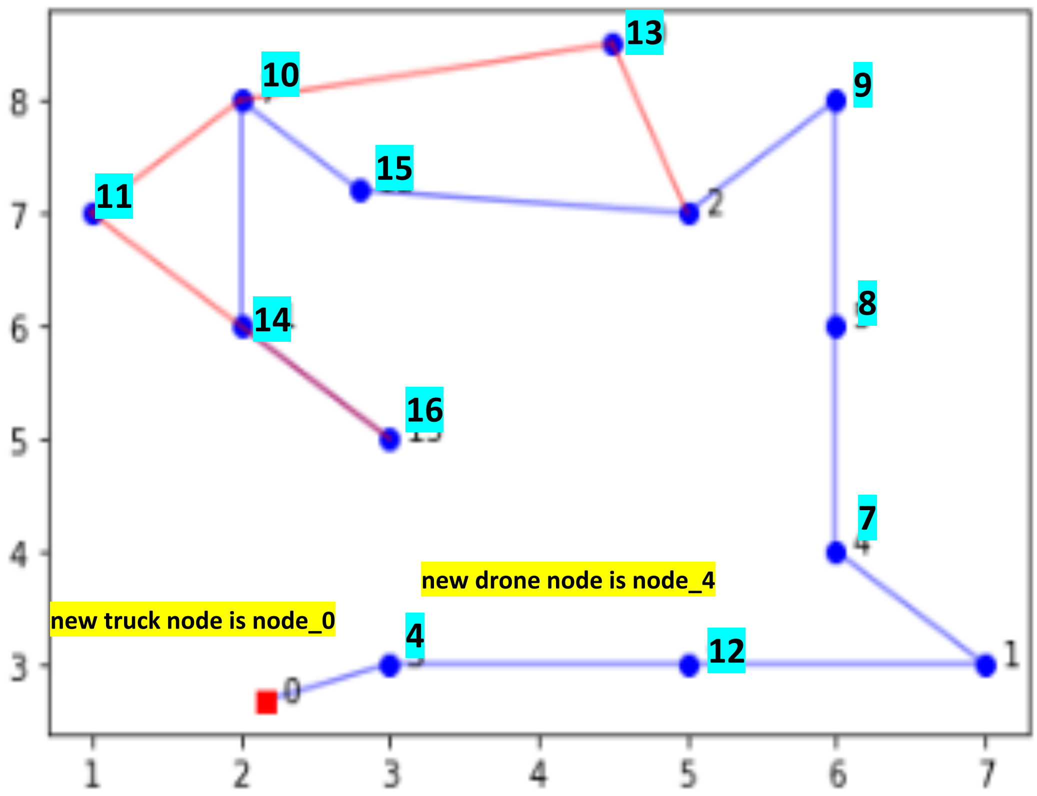



```
#TRUCK ROUTES valid updated for drone(scen_15_cust_20En2IFnewv15-2dfv)
active_arcs3 =[(0, 3),(3, 5),(5, 4), (4, 12), (12, 7), (7, 8), (8, 2), (2, 15), (15, 10), (10, 14), (14,16)]
```

```
#DRONE ROUTES valid updated for drone(scen_15_cust_20En2IFnewv15-2dfv)
active_arcs4 = [(0, 6), (6 , 4 ),(4, 1), (1, 7), (7, 9), (9, 2), (2, 13), (13, 10), (10, 11), (11, 16)]
```

Common disruption 15 - customer instance (v15)

Optimum solution (DISRUPTION)



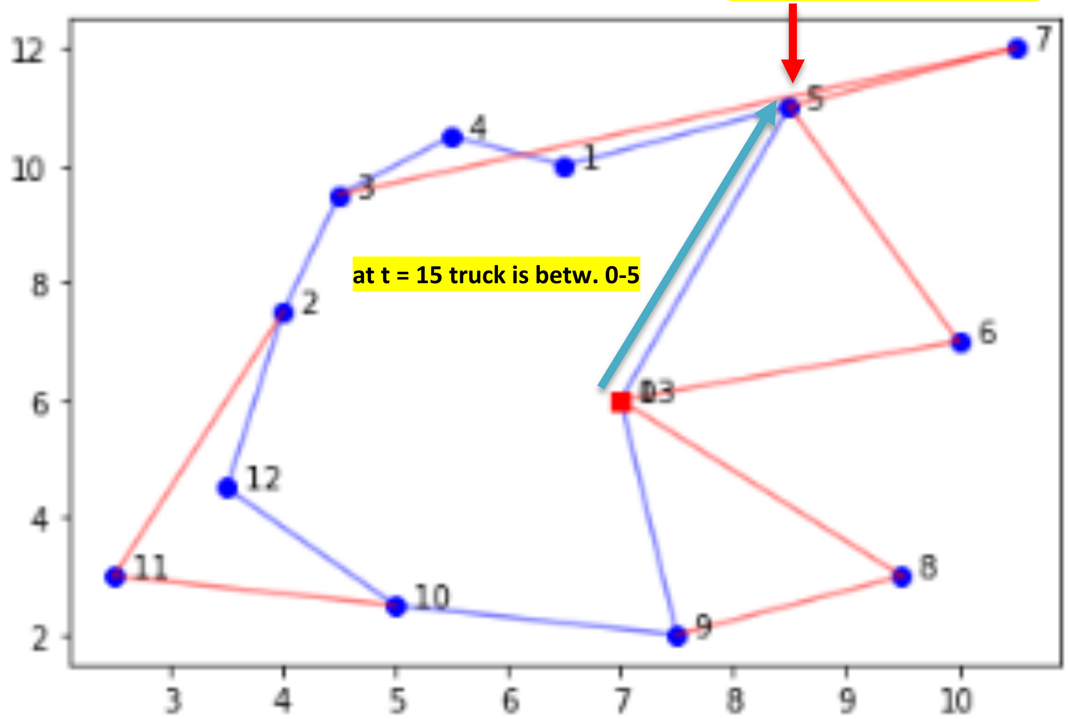

```
#TRUCK ROUTES
active_arcs3 =[(0, 3),(3, 9),(9, 1), (1, 4), (4, 5), (5, 6), (6,2),
               (2, 12), (12, 7), (7,11), (11, 13)]
```

```
#DRONE ROUTES
active_arcs4 = [(2, 10), (10 , 7), (7, 8), (8, 13)]
```

Optimum Solution 12 - customer instance (v5)

Optimum solution (INITIAL)

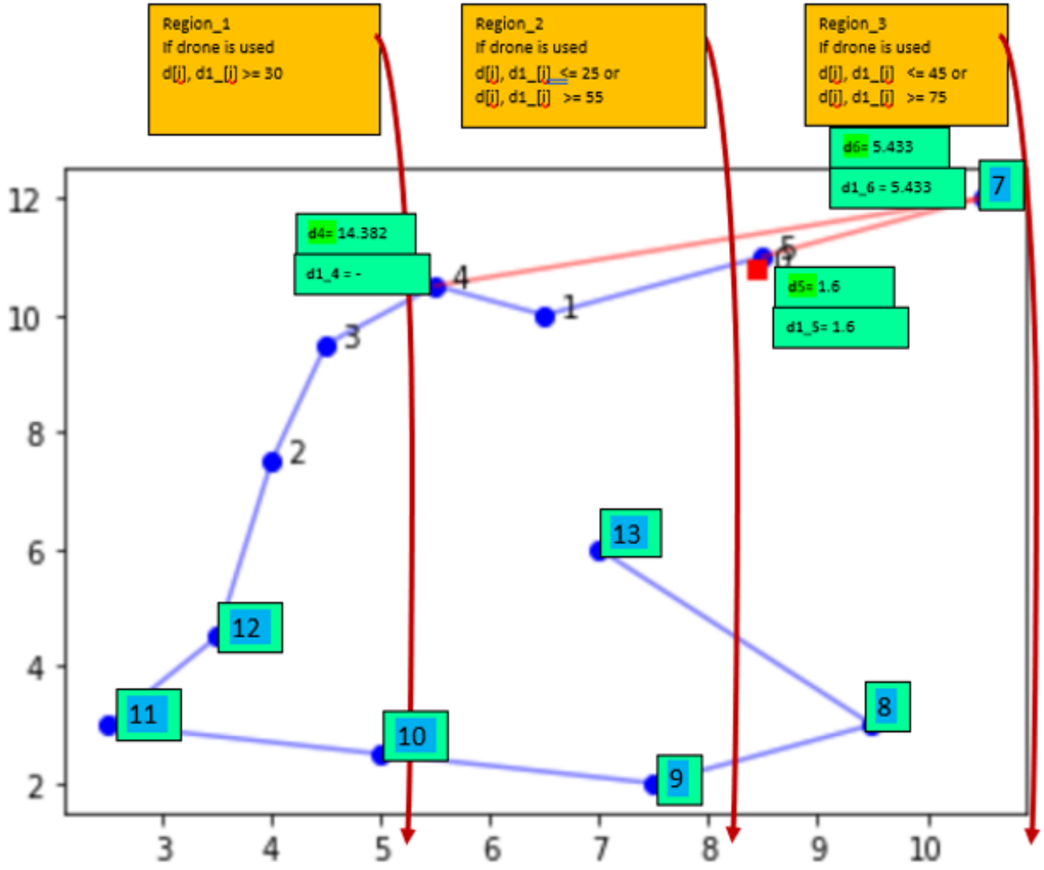
at t = 15 drone is on node_5



Rolling disruption 12 - customer instance (v5)

Optimum solution (ROLLING DISRUPTION)

```
#TRUCK ROUTES
active_arcs3
[(0, 5),
 (1, 4),
 (2, 11),
 (3, 2),
 (4, 3),
 (5, 1),
 (7, 12),
 (8, 7),
 (9, 8),
 (10, 9),
 (11, 10)]
```




```
#DRONE ROUTES
active_arcs4
[(5, 6), (6, 4)]
```