

# Project V: Transportation and Logistics

---

Karan Kashyap and Erica Song  
(commandLine and DeltaAirlines)

# Solution Strategy

## Break problem into 2 parts:

- Find initial feasible solution
- Optimize

## Initial Solution

- No concern for optimality → so basically, a binpacking problem
- Easy to model using CP

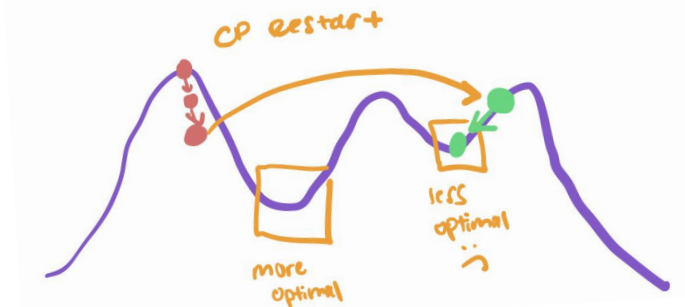
## Optimization

- Local search

# Initial Local Search Strategy (Not Ideal)

- Define multiple moving strategies
- Get **large neighborhood** from each moving strategy and pick the most optimal solution that is feasible
- Exploration
  - **If no new incumbent found within time limit, resolve with CP and continue LS from there**
- Exploitation
  - Large neighborhood due to considering many candidate solutions enables exploitation

**Problem:** Not enough exploitation using work done so far!



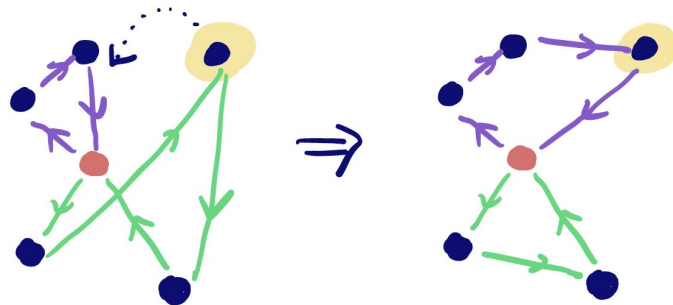
# Improved Local Search Strategy

- Define multiple moving strategies
- Get **a single neighbor** from each moving strategy and pick the most optimal solution that is feasible
- **Tolerance**
  - Allow moves to less optimal solutions within tolerance
  - If no new incumbent found within time limit, **move back to incumbent solution and decrease tolerance**
- Exploration
  - Early on, high tolerance enables exploration of search space
- Exploitation
  - Later on, low tolerance forces movement towards (local) minima

# Neighborhood Definition

## **CrossRouteCustomerMove:**

Move a random customer from current route to a random position in a different (randomly picked) route



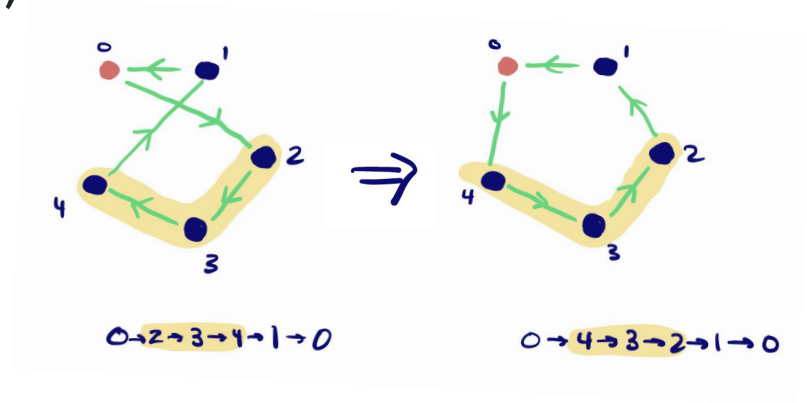
## **RandomCustomerMovement:**

Move random customer to a random location in a randomly picked route (could be within the same route)

## Neighborhood Definition (contd)

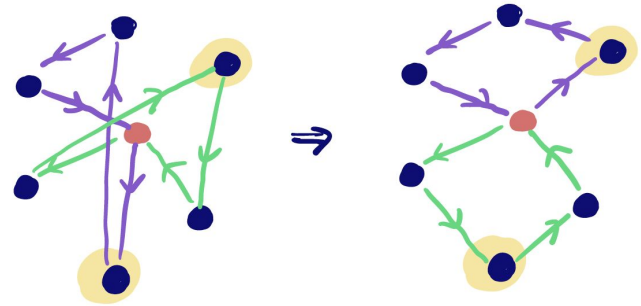
### TwoOpt:

Reverse the order of a random subpath in a route



### CrossRouteCustomerExchange:

Randomly swaps two customers in different routes



# Implementation Techniques

Speed up candidate solution evaluation by using diff (vs previous solution) to compute feasibility, total distance

- Moves in solution space only impact 1-2 routes; therefore, feasibility and distance change computations can be limited to an analysis of those changes

# Other Things We Tried (that didn't work)

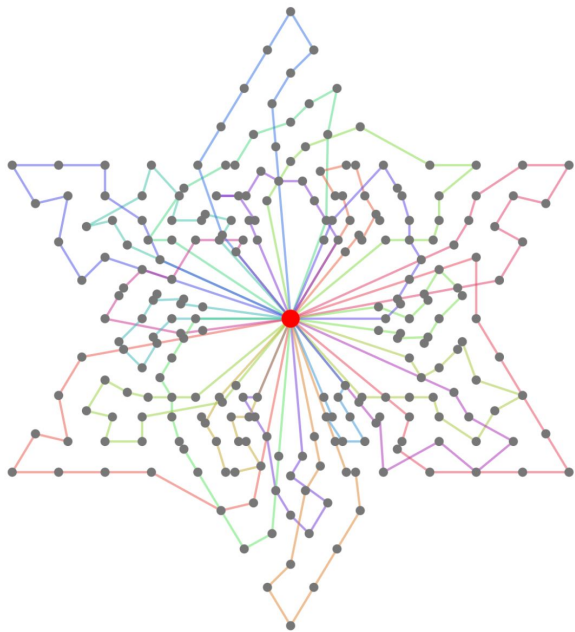
## **Multithreading**

- Helped with large neighborhoods
- Smaller neighborhoods along with optimized feasibility and distance checking for candidate solutions rendered this useless

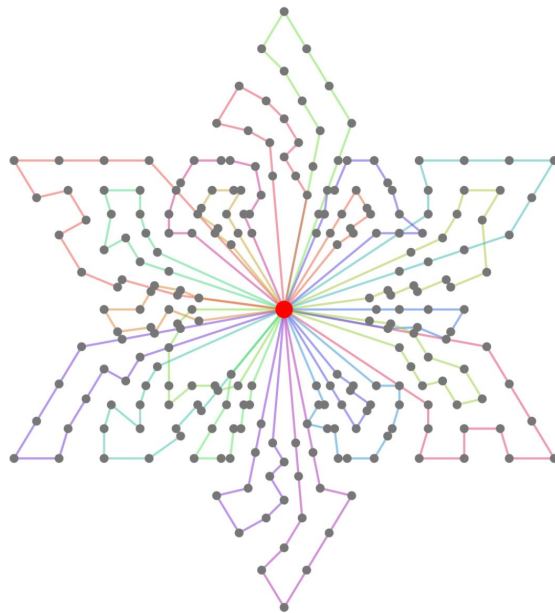


# Experimental Observations (241\_22\_1.vrp)

OR Tools (after 30 seconds):



Our Solution (after 295 seconds):



# Experimental Observations (Verification)

- Comparable to or better than OR tools on every instance
- Verified feasibility and total distance for solutions separately

	<b>OR Tools (after 10 seconds)</b>	<b>Our Solution (after 295 seconds)</b>	<b>Percentage Difference</b>
16_5_1.vrp	337	334.96	-0.61%
51_5_1.vrp	555.5	524.61	-5.56%
151_15_1.vrp	3167.5	3103.27	-2.03%
262_25_1.vrp	6231	5704.01	-8.46%
386_47_1.vrp	28319	25940.84	-8.40%

Time Spent:  
>50 hours

Thank you!  
Questions?