# ShipIt

## Milestone 4

Khaled Seyam, Omar Elfatairy, Omar Khairy, Mohamed Alaa, Ahmed Osama

10th July, 2021

1. Business Problem
2. Input & Output
3. Research Problem
4. Mathematical formulation
5. Brief Description of Each Approach
6. Results

1. **Business Problem**
2. Input & Output
3. Research Problem
4. Mathematical formulation
5. Brief Description of Each Approach
6. Results

## Business Problem

- Small businesses find it costly to deliver small shipments overseas through carriers.

# Business Problem

- Small businesses find it costly to deliver small shipments overseas through carriers.
- Carriers find it a hassle to specially handle small shipments and may even reject the shipments.

## Solution

- Provide service that facilitate shipments for small businesses to different locations around the world.

## Solution

- Provide service that facilitate shipments for small businesses to different locations around the world.
- Offers discounted prices which are done through a similar process of consolidating and routing the shipments.

# Solution

- Provide service that facilitate shipments for small businesses to different locations around the world.
- Offers discounted prices which are done through a similar process of consolidating and routing the shipments.
- By working as a medium between the carriers and the small businesses, the carriers offer us discounted rates which benefits the small businesses and the carriers.

1. Business Problem
2. Input & Output
3. Research Problem
4. Mathematical formulation
5. Brief Description of Each Approach
6. Results

# Input Format

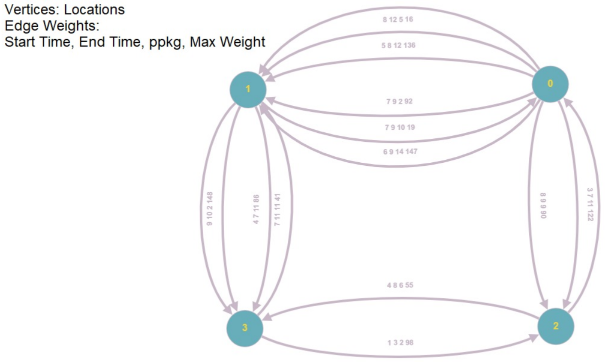
```
6 10 4 6
0 1 10 5 40
1 2 10 3 40
0 3 7 5 40
0 5 5 6 50
0 1 1 4 1 20
0 1 1 4 2 15
0 1 1 3 3 20
0 1 1 3 4 50
0 3 1 4 5 30
3 5 5 6 3 30

nl,mt,ns,nt
ns lines of: orgs,dests,wght,time,pps
nt lines of: startTripLoc,endTripLoc,startTime,endTime,ppkg,mw
```

# Output Format

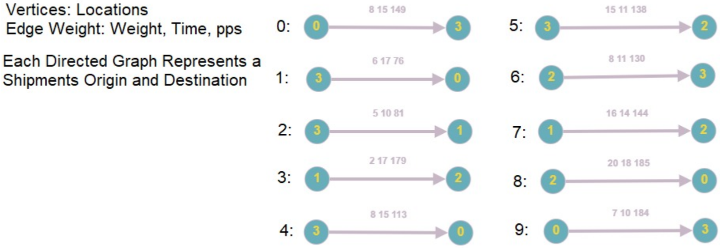
2D array with the index of the shipment and the inner array are the trips that the shipments taken.
For example: [ []*nt ]*ns

# Example Visual Input



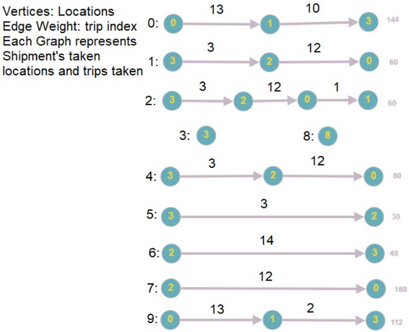

Representation of available trips.

# Example Visual Input




Representation of shipments and their origin and final destination.

# Example Visual Output




Representation of output path and trips taken by the shipments.

1. Business Problem
2. Input & Output
3. Research Problem
4. Mathematical formulation
5. Brief Description of Each Approach
6. Results

## Research Problem

- The problem is a variant to the famous Vehicle Routing Problem (VRP)

# Research Problem

- The problem is a variant to the famous Vehicle Routing Problem (VRP)
- Our Carriers does not return to depot similar to Open VRP(OVRP)

# Research Problem

- The problem is a variant to the famous Vehicle Routing Problem (VRP)
- Our Carriers does not return to depot similar to Open VRP(OVRP)
- Shipments have time deadline similar to VRP Time Window(VRPTW)

# Research Problem

- The problem is a variant to the famous Vehicle Routing Problem (VRP)
- Our Carriers does not return to depot similar to Open VRP(OVRP)
- Shipments have time deadline similar to VRP Time Window(VRPTW)
- Shipments have different origins similar to Multiple Depots VRP (MDVRP).

# Research Problem

- The problem is a variant to the famous Vehicle Routing Problem (VRP)
- Our Carriers does not return to depot similar to Open VRP(OVRP)
- Shipments have time deadline similar to VRP Time Window(VRPTW)
- Shipments have different origins similar to Multiple Depots VRP (MDVRP).
- Trips may pick shipments along the way if they have the same destination similar to Pickup-and-delivery VRP (PDVRP).

1. Business Problem
2. Input & Output
3. Research Problem
4. Mathematical formulation
5. Brief Description of Each Approach
6. Results

# Ranges

- $i, j \in 0..nl$ will be used as index for locations

## Ranges

- $i, j \in 0..nl$ will be used as index for locations
- $s \in 0..ns$ will be used as index for shipment

# Ranges

- $i, j \in 0..nl$ will be used as index for locations
- $s \in 0..ns$ will be used as index for shipment
- $t \in 0..nt$ will be used as index for trip

# Ranges

- $i, j \in 0..nl$ will be used as index for locations
- $s \in 0..ns$ will be used as index for shipment
- $t \in 0..nt$ will be used as index for trip
- $b, e, l \in 0..mt$ will be used as index for time

# Decision Variables

- $x_{st}$ is a boolean that represents a shipment $s$ will be sent in trip $t$

# Decision Variables

- $x_{st}$ is a boolean that represents a shipment $s$ will be sent in trip $t$
- $y_s$ is a boolean that represents that shipment $s$ will be sent

## Decision Variables

- $x_{st}$ is a boolean that represents a shipment $s$ will be sent in trip $t$
- $y_s$ is a boolean that represents that shipment $s$ will be sent
- $f_{sil}$ is a boolean that represents if shipment $s$ is in location $i$ at time $l$

# Model Parameters

## Model Parameters

1. $ppkg_t$ represents the price per kg given by the carrier for trip $t$
2. $mw_t$ represents the max weight given by the carrier for trip $t$
3. $route_{tij}$ represents that trip $t$ goes from $i$ to $j$
4. $tim_{tbe}$ represents that trip $t$ starts at time $b$ and ends at time $e$
5. $orgs_s$ represents the origin of shipment $s$
6. $dests_s$ represents the location that shipment $s$ needs to be delivered to
7. $wght_s$ represents the weight of shipment s
8. $time_s$ represents the max time to deliver shipment $s$ stated by the customer
9. $pps_s$ represents the price the customer is willing to pay for the shipment delivery

# Objective Function

## Objective Function

We need to maximize $\sum_{s} y_s$

# Constraints

$$(1) x_{st}, y_s, f_{sil} \in 0, 1 \qquad Boolean$$

# Constraints

$$(1) x_{st}, y_s, f_{sil} \in 0, 1 \qquad \textit{Boolean}$$

$$(2) \sum_t x_{st} * ppkg_t * wght_s \leq pps_s * y_s \qquad \forall s \qquad \textit{Respect stated price by customer}$$

# Constraints

$$(1) x_{st}, y_s, f_{sil} \in 0, 1 \qquad \textit{Boolean}$$

$$(2) \sum_t x_{st} * ppkg_t * wght_s \leq pps_s * y_s \qquad \forall s \qquad \textit{Respect stated price by customer}$$

$$(3) \sum_s x_{st} * wght_s \leq mw_t \qquad \forall t \qquad \textit{Respect weight per trip}$$

# Constraints

$$(1) x_{st}, y_s, f_{sil} \in 0, 1 \qquad \textit{Boolean}$$

$$(2) \sum_t x_{st} * ppkg_t * wght_s \leq pps_s * y_s \qquad \forall s \qquad \textit{Respect stated price by customer}$$

$$(3) \sum_s x_{st} * wght_s \leq mw_t \qquad \forall t \qquad \textit{Respect weight per trip}$$

$$(4) f_{s(orgs_s)0} = 1 \qquad \forall s, i \qquad \textit{Packages are at their origin initially}$$

## Constraints

$$(5) f_{sil} = f_{si(l-1)} - \sum_{jet} x_{st} * route_{tij} * tim_{tle} + \sum_{jbt} x_{st} * route_{tji} * tim_{tbl} \qquad \forall s, i, l \qquad l > 0$$

*The shipment will only be delivered to another location from the current facility only if it arrived at the current facility, and the the shipment will be persistent as long as the shipment was not sent away. The shipment can be dropped at any facility at a certain time*

## Constraints

$$(5) f_{sil} = f_{si(l-1)} - \sum_{jet} x_{st} * route_{tij} * tim_{tle} + \sum_{jbt} x_{st} * route_{tji} * tim_{tbl} \qquad \forall s, i, l \qquad l > 0$$

*The shipment will only be delivered to another location from the current facility only if it arrived at the current facility, and the the shipment will be persistent as long as the shipment was not sent away. The shipment can be dropped at any facility at a certain time*

$$(6) \sum_{tj} x_{st} * route_{tij} \leq 1 \qquad \forall s, i \qquad \textit{Shipments can leave a location only once}$$

# Constraints

$$(7) \sum_{i} f_{sil} \leq 1 \qquad \forall s, l \qquad \textit{Shipment should be in max one place at a given time}$$

# Constraints

$$(7) \sum_{i} f_{sil} \leq 1 \qquad \forall s, l \qquad \textit{Shipment should be in max one place at a given time}$$

$$(8) \sum_{j,t} x_{st} * route_{t(dests_s)j} = 0 \qquad \forall s \qquad \textit{Shipments can't leave their final destination}$$

## Constraints

$$(7) \sum_{i} f_{sil} \leq 1 \qquad \forall s, l$$ *Shipment should be in max one place at a given time*

$$(8) \sum_{j,t} x_{st} * route_{t(dests_s)j} = 0 \qquad \forall s$$ *Shipments can't leave their final destination*

$$(9) f_{s(dests_s)(times_s)} = y_s \quad \forall s$$ *Shipment will be taken only if present at its destination*

# Constraints

$$(7)\sum_{i} f_{sil} \leq 1 \qquad \forall s, l \qquad \textit{Shipment should be in max one place at a given time}$$

$$(8)\sum_{j,t} x_{st} * route_{t(dests_s)j} = 0 \qquad \forall s \qquad \textit{Shipments can't leave their final destination}$$

$$(9) f_{s(dests_s)(times_s)} = y_s \quad \forall s \quad \textit{Shipment will be taken only if present at its destination}$$

$$(10)\sum_{t} x_{st} \leq nt * y_s \quad \forall s \quad \textit{Shipments can't travel if they will not arrive at destination}$$

1 Business Problem

2 Input & Output

3 Research Problem

4 Mathematical formulation

5 Brief Description of Each Approach

6 Results

## Brief Description of Each Approach

### Integer programming

In the integer programming approach we transform our mathematical model into code by passing the constraints and creating arrays to hold each model parameter, then the whole thing is passed to the MIP solver of ORTOOLS, along with the maximization function to provide us with the optimal solution.

# Brief Description of Each Approach

## Dynamic programming

The approach in dynamic programming was to recursively try to send each shipment across all trips that are available to be taken without violating any constraints and then recursively compute which is the best path for this shipment. Then if the origin of the current shipment is the destination we move on to the next shipment, the state in our algorithm is the number of shipments left, the weight available in each trip, the origin of the current shipment, and the current time of this shipment, thus we just retrieve the value of the this state using memoization to avoid any unnecessary complexity.

# Brief Description of Each Approach

## Greedy algorithm

The simple approach here is that we iterate over the shipments and we check the destination of the shipment trying to find a trip that goes directly without violating any constraints we send the shipment but if not we send the package with the cheapest trip trying to blindly an indirect trip, of course if there are more than one trip to be taken directly the first one encountered would be taken.

## Brief Description of Each Approach

### Genetic Algorithm

Create an initial population using a randomized greedy approach to ensure diversity of solutions. We run the algorithm to create a population of maximum size 20 unique solutions and we create 100 generations. Calculate the fitness score of each solution by counting the number of successful shipments. The process of creating a generation is divided into 3 parts. Crossover, pair each parent together to produce 2 off springs that take from the parents the shipments based on 50% probability with uniform distribution. Mutation, mutate by randomly dropping a shipment with a probability of 1.5%. Selection, replace the least fittest solution in the population by the offspring if the fitness score of the offspring is greater.

1. Business Problem
2. Input & Output
3. Research Problem
4. Mathematical formulation
5. Brief Description of Each Approach
6. Results

# Results Summary

## Results summary

The results shown in the table below where each algorithm-z represents the number of shipments successfully sent, and algorithm-t shows the time to compute in seconds.

| test_case | ns | nt | Genetics_z | Genetics_t | Greedy_z | Greedy_t | DP_z | DP_t | IP_z | IP_t |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 6 | 3 | 0 | 2 | 0 | 3 | 0 | 3 | 1 |
| 1 | 7 | 15 | 7 | 1 | 5 | 0 | 7 | 7 | 7 | 2 |
| 2 | 7 | 15 | 2 | 1 | 0 | 0 | 3 | 1 | 3 | 2 |
| 3 | 10 | 15 | 5 | 1 | 5 | 0 | 5 | 0 | 5 | 11 |
| 4 | 7 | 10 | 3 | 0 | 3 | 0 | 3 | 0 | 3 | 5 |
| 5 | 7 | 10 | 4 | 0 | 4 | 0 | 4 | 0 | 4 | 5 |
| 6 | 10 | 15 | 8 | 1 | 7 | 0 | 0 | 0 | 8 | 20 |
| 7 | 12 | 20 | 6 | 1 | 3 | 0 | 0 | 0 | 6 | 21 |
| 8 | 12 | 20 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 |
| 9 | 15 | 10 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| 10 | 15 | 30 | 7 | 1 | 5 | 0 | 0 | 0 | 0 | 0 |
| 11 | 15 | 30 | 7 | 5 | 4 | 0 | 0 | 0 | 0 | 0 |
| 12 | 20 | 40 | 10 | 2 | 8 | 0 | 0 | 0 | 0 | 0 |
| 13 | 20 | 40 | 5 | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| 14 | 25 | 50 | 6 | 2 | 5 | 0 | 0 | 0 | 0 | 0 |
| 15 | 25 | 50 | 7 | 2 | 5 | 0 | 0 | 0 | 0 | 0 |
| 16 | 30 | 60 | 14 | 3 | 8 | 0 | 0 | 0 | 0 | 0 |
| 17 | 30 | 60 | 18 | 3 | 4 | 0 | 0 | 0 | 0 | 0 |
| 18 | 35 | 70 | 24 | 4 | 14 | 0 | 0 | 0 | 0 | 0 |
| 19 | 40 | 75 | 24 | 4 | 18 | 0 | 0 | 0 | 0 | 0 |
| 20 | 45 | 90 | 26 | 5 | 10 | 0 | 0 | 0 | 0 | 0 |
| 21 | 50 | 100 | 22 | 4 | 21 | 0 | 0 | 0 | 0 | 0 |
| 22 | 10 | 150 | 10 | 3 | 10 | 0 | 0 | 0 | 0 | 0 |

# Detailed Results

| features | Genetics | Greedy | DP | IP |
|---|---|---|---|---|
| Total no. c | 4 | 4 | 4 | 4 |
| Total Price | 90 | 92 | 92 | 90 |
| Total Wei | 13 | 13 | 13 | 13 |
| Average D | 3.5 | 3.5 | 3.5 | 3.5 |
| Average P | 22.5 | 23 | 23 | 22.5 |

## Detailed result

The detailed result is a table representing the solutions for a testcase using a matrix. The matrix rows consists of the Total no. of Shipments, Total price Paid, Total weight, average delivery time and average price per shipment, respectively. Each column is a solution of the said features using the mentioned algorithms.

# Thank you!