CS606 AI Planning and Decision Making

Multi-App Scheduling for Parcel Delivery Riders

Final Report

14 April 2023

Prepared by: CS606 (Group 7)

Biondi Lee	biondi.lee.2020@scis.smu.edu.sg
Lee Shuxian	shuxian.lee.2022@mitb.smu.edu.sg
Yan Licheng	licheng.yan.2021@mitb.smu.edu.sg
Yong Ruo Xi	ruoxi.yong.2022@mitb.smu.edu.sg

Contents

1 Motivation, Project Objectives and Scope

1.1 Motivation

Demand for delivery services in Singapore has grown exponentially in recent years, fuelled by growth in online shopping platforms such as Lazada and Shopee amidst the pandemic. The number of delivery drivers has also increased in tandem. In 2020, there were a total of 11,300 car and light goods vehicle drivers (including delivery drivers) – a 31% increase from 2019 since the figure was first reported (Ministry of Manpower, 2021).

To maximise their income in this line, delivery drivers often have to work long shifts with minimal breaks, multitask, or even "multi-app" where they work on multiple delivery platforms simultaneously to get more and better orders. While multi-apping is not endorsed by delivery companies possibly due to concerns on service delivery standards and liabilities, multi-apping is a common practice among delivery riders in Singapore and other countries such as US (Griffith, 2022) and Australia (Barrat, 2020). Anecdotally, delivery drivers in Singapore reported that they were increase their earnings by as much as two times through multi-apping as compared to working for a single platform.

While multi-apping can make deliveries more efficient with proper route planning, there is no known driver-centric app in Singapore today that consolidates these jobs across different platforms. Presently, drivers manually choose their jobs on a "first-come-first-serve basis" rather instinctively, toggling between different platforms. As a result, a rider can easily overcommit (intentionally or not) – which would lead to inability to meet service delivery standards, financial penalties, and fatigue.

1.2 Project Objectives

In view of the challenges faced by delivery drivers, the objective is to develop a personal scheduling tool for delivery drivers to help them maximize their daily earnings. This driver-centric tool shall (a) support a multi-apping scenario, (b) plan around the driver's preferences, and (c) keep to constraints on service delivery standards and the driver's well-being.

1.3 Scope

The project will focus on parcel delivery scenarios for a single driver across multiple platforms. To simulate a real scenario, drivers will be able to choose their jobs a day before during the pre-match period, and new jobs will also appear dynamically throughout the day.

2 Project Definition and Assumptions

2.1 Definitions and Assumptions

2.1.1 Inputs

For all absolute time inputs, we receive the inputs in local datetime format, convert them into epoch seconds and then offset them by the rider's desired shift start time. This is for easy subsequent calculations.

Rider's inputs:

- $t^s_{start} = 0$: Desired Shift Start Time (in minutes)
- t^s_{end}: Desired Shift End Time (in minutes)
- Desired Shift Start Location ([latitude, longitude] in degree decimal)
- Desired Shift End Location ([latitude, longitude] in degree decimal)
- C: Maximum Vehicle Capacity (in kg)

- R: Minimum Rest Duration (in minutes)
- W: Maximum Working Period before Rest (in hours)

Available Jobs:

- $t^j_{e,pick}$: Expected pickup time for job j (in minutes)
- $t^j_{e,deliv}$: Expected delivery time for job j (in minutes)
- Pickup location for job j ([latitude, longitude] in degree decimal)
- Delivery location for job j ([latitude, longitude] in degree decimal)
- Order received time for job j
- Parcel weight of job j
- p_j: Expected payout of job j
- Platform where job j was received from.

Planning horizon of T minutes. The planning horizon starts at the rider's desired shift start time (0) and ends at the rider's desired shift end time ($T = t^s_{end} - t^s_{start}$).

2.1.2 Outputs

The rider will decide which jobs to accept for his shift in two phases: pre-match and dynamic. The pre-match phase occurs the day before his shift where he can choose from a list of available jobs for the next day. The dynamic phase occurs after the shift has started, where jobs are available for acceptance on an ad-hoc basis.

At **pre-match** phase: Schedule of actual pickup ($t^j_{a,pick}$) and delivery ($t^j_{a,deliv}$) timings for each job accepted (let J denote the list of all accepted jobs) in the entire shift duration subjected to the pre-match constraints in Section 2.1.3.

At **dynamic** phase: Updated schedule of actual pick-up and delivery timings for each job accepted in the entire shift duration subjected to the pre-match and dynamic constraints in Section 2.1.3.

2.1.3 Constraints

2.1.3.1 Pre-match Constraints

Accepted Jobs:

1. Actual pickup and delivery times for all jobs must be within the planning horizon.
2. Actual pickup time must be at or after expected pickup time and before actual deliver time.
3. Actual pickup time must be no later than 15min from the expected pickup time.

Schedule:

4. Travel time[1] between two consecutive locations must be less than the time difference on the schedule. This includes the travel time from the rider's shift start location to the first job's pickup location, and the travel time from the last job's delivery location to the rider's shift end location.

Capacity:

5. Total weight carried at any point in time cannot exceed the driver's vehicle capacity C.

Driver:

[1] The list of all possible locations includes the rider's desired start and end location and all available jobs' pickup and delivery location. We pre-generated the distance-time matrix for each location pair using Project OSRM's Table Service API (OSRM API Documentatio, n.d.) which calculates the distance and estimated duration of a route between two coordinates with OpenStreetMap data.

6. Rider cannot be at two locations in the same 1-minute timeslot.
7. Rider must fulfill his minimum rest period of R minutes after every W hours of working. Rest period refers to the time duration between consecutive locations when the rider is not driving i.e., time difference in schedule – travel duration. We assumed that rest period is non-cumulative i.e., must be taken at one shot between one pair of consecutive locations.

2.1.3.2 Dynamic Constraints

Dynamic constraints are constraints which apply once the shift has started, pre-match constraints still apply.

1. Accepted jobs in the pre-match phase cannot be cancelled.
2. Actual pickup and delivery times of dynamic jobs cannot be before current time.
3. Once rejected, dynamic jobs will no longer be available for consideration.

2.1.4 Objective

Our objective is to maximize the total profit in a single shift. The profit is calculated by the total payout earned from accepted jobs, minus the penalty of late deliveries and the fuel cost incurred.

$$\max \sum_{j\in J}(p_j - \text{penalty}_j) - \text{total fuel cost}$$

$penalty_j$: 50% of the job j's payout p_j if the actual delivery time of job j is later than its expected delivery time, else 0. $t^j_{a,deliv} > t^j_{e,deliv} \rightarrow penalty_j = 0.5 * p_j$

total fuel cost: $\text{fuelprice} * \text{fuelconsumption} * \sum_{i=1}^{len(Locations)} \text{traveldistance}_{i,i+1}$ where Locations is a sequential list of locations that the rider travels to in the schedule. It starts with the shift start location and ends at the shift end location. For this project, we fixed the fuel price and consumption rates at $\text{fuelprice} =$ SGD2.75 per litre, and $\text{fuelconsumption} = 0.075$ litres per km.

2.1.5 Assumptions

2.1.5.1 No multiple events within 1 minute

We assumed that the rider will not be able to attend to multiple events within a 1-minute timespan and ignored the possibility where multiple jobs could be picked up from the same location or delivered to the exact same location at the same time. This assumption should not affect the rider's eventual job schedule significantly if the number of jobs from/to the same location is small as per our dataset (maximum of 2 jobs picked up from/ delivered to the same location). However, if there are many jobs from/to the same location, this may result in an inefficient schedule where the rider attends to a long duration of consecutive 1-minute events at the same location when it could be compressed to a shorter timeframe.

2.1.5.2 Pickup Window Duration

We assumed a pickup window of 15 minutes for customers for each pickup in pre-match Constraint 3. This maximum waiting duration requirement could differ across platforms and jobs if customers could choose the service level required (e.g., express deliveries).

2.1.5.3 Penalty Scheme across Platforms

We assumed a flat penalty rate of 50% for each accepted job delivered late for all platforms. In reality, the penalty scheme for late deliveries will likely differ across platforms.

2.2 Synthetic Data Generation

A total of 2,759 candidate jobs were created across Singapore covering a span of a single day between 0600h and 2300h, simulating peak demand hours in Figure 1. The job attributes are listed in Table 1.

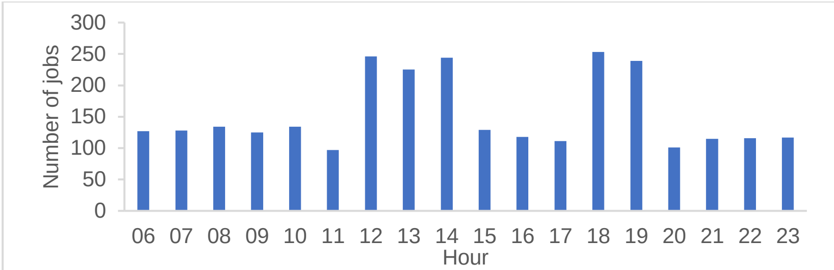


Figure 1: Number of jobs by hour

Table 1: Job attributes

Attribute	Description
Order received time	For pre-match jobs: 2200h on the previous day (or any user input)[2] For dynamic jobs: current day at randomized timings
Pickup and delivery locations	Randomly matched address points in Singapore
Pickup time	Randomly generated time
Delivery time	Estimated, using Google distance matrix API to calculate the driving time with an additional random buffer of 60 to 180 minutes
Platform	Random assignment of one of the following platforms: GrabExpress, J&T Express, Lalamove and Pickupp
Payout	Estimated based on the platform's publicized rates (details in **Annex A**)
Weight	Randomised, between 0.5kg to 10kg
Mode of transport	Car - Driving

A total of 300 dynamic jobs were also created, covering a span of an 8-hour shift between 1000h to 1800h. Further details on the dataset can be found in **Annex A**.

3 Proposed Model and Algorithm

Our team chose the ALNS algorithm to solve this assignment problem due to its powerful optimization capability in solving complex combinatorial problems with non-linear constraints. The repair and destroy functions, along with adaptable weights, enable us to navigate the complex problem space.

3.1 Construction Heuristics

To construct our initial solution, our team used a greedy approach by randomly shuffling the list of jobs and accepts them if they pass the feasibility check with the "can_assign" function.

3.2 Destroy and Repair Heuristics

3.2.1 Destroy Operators

3.2.1.1 Random Destroy

For this operator, we randomly remove one job from those currently assigned to the driver to create space for new assignments that may lead to a higher objective value.

3.2.1.2 Random Destroy Multi

Here we randomly remove multiple jobs at once with a 10% degree of destruction to create space to allow new job assignments that can potentially improve the objective value.

[2] In this case, we assumed the driver would log in to select the jobs during pre-matching at 2200h.

3.2.1.3 *Random Destroy Lowest Pay*

Here we remove a single job randomly from the bottom 20% of all jobs assigned to make room for assignment of potentially higher paying jobs that can improve the objective value.

3.2.2 Repair

3.2.2.1 *Random Repair*

We shuffle unassigned jobs and generate random pickup and delivery time slots for each. Thereafter, using the "can_assign" function, we check if the job can be assigned in a greedy fashion.

3.2.2.2 *Random Repair Best Pay*

We prioritize the most lucrative jobs by arranging them in descending order of the payout value. For each job, we generate random pickup and delivery time slots and assign greedily with "can_assign" function.

3.2.2.3 *Random Repair Long*

We prioritize jobs with similar pickup locations in hopes of improving the objective value. We sort all unassigned jobs by starting longitude in descending order. For each job, we generate random pickup and delivery time slots and assign greedily with "can_assign" function.

3.2.2.4 *Random Repair On-Time*

To minimize penalties, we only repair using jobs whose timeslots fall in the no-penalty zone. The unassigned jobs are shuffled and timeslots checked by "can_assign" function before being allowed to assign to the driver.

3.2.3 Dynamic Destroy & Repair

3.2.3.1 *Dynamic Destroy*

We randomly choose jobs that have not commenced for "pseudo-removal" to allow more flexibility for accommodating new jobs. These previously assigned jobs will be eventually added back into the list because of the business requiement that jobs that haven been previously accepted by the driver cannot be destroyed. If the job was previously assigned in the prematch phase, we will add it to a list called "prev_assigned" with current pickup and delivery time slots.

3.2.3.2 *Dynamic Repair*

In the repair phase, our operator first assigns jobs from the "prev_assigned" list as they were accepted during the pre-match phase. Where possible we will assign new pickup and delivery slots to generate variation. The new dynamic jobs are then assigned greedily similar to Random Repair Operator.

3.3 Weight Update Strategy

Our team chose a grid search over a range of weights and lambda values from 0.90 to 0.98 for 300 iterations. We vary the weight pattern with linear, geometrical, quadratic, and factorial values. Our best performing weight update strategy gets us an objective value of 278.85 using 0.96 lambda and factorial weights.

Table 2: Hyperparameter Search

Grid Search Strategy (300 Iterations)				
Lambda	**Linear [3, 2, 1, 0.5]**	**Geometrical [$2^3, 2^2, 2^1, 2^0$]**	**Quadratic [$4^2, 3^2, 2^2, 1^2$]**	**Factorial [4!, 3!, 2!, 1!]**
0.9	253.81	263.60	267.29	253.18
0.92	217.37	254.01	272.31	264.59
0.94	232.91	251.06	221.49	250.06
0.96	257.54	258.06	211.78	**278.85**
0.98	221.82	269.47	260.28	261.71

4 Experimental Results

4.1 Operator Performance

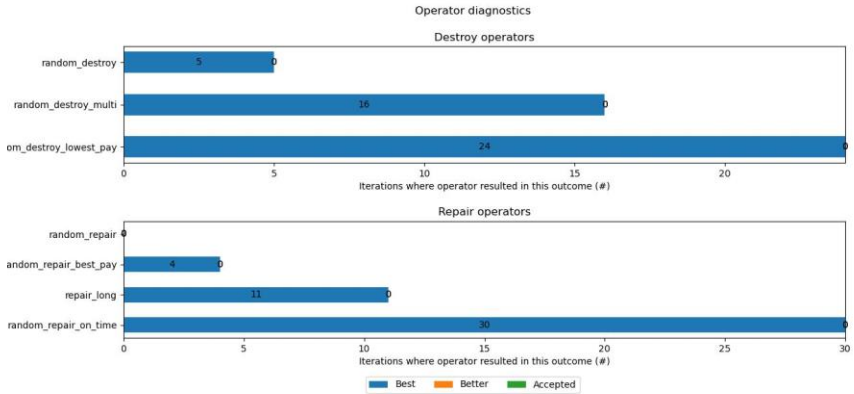


Figure 2: Operator Statistics

The destroy statistics showed that "random_destroy_multi" and "random_destroy_lowest_pay" outperformed "random_destroy". We see that removing jobs with lower pay showed better results than multi-job removal, revealing the sub-optimality in accepting lower-paying jobs in the solution space.

The repair statistics showed that heuristics performed better than random decisions. Of the three, "random_repair_best_pay" performed the worst, suggesting that accepting higher paying jobs does not always lead to optimality due to distance and time constraints. The better performance of “repair_long” showed the benefits of considering jobs in similar locations. Ultimately the best strategys is still to deliver on time to avoid the 50% late penalty.

4.2 Pre-match Performance

To minimize the driver’s waiting time while generating the schedule, our team used 300 iterations for pre-match which completes within 3 mins using our best performing hyperparameters. This gave us the objective value of $278.85. To extend further, our team also experimented with 2000 iterations to obtain 16% improvement to $324.81.

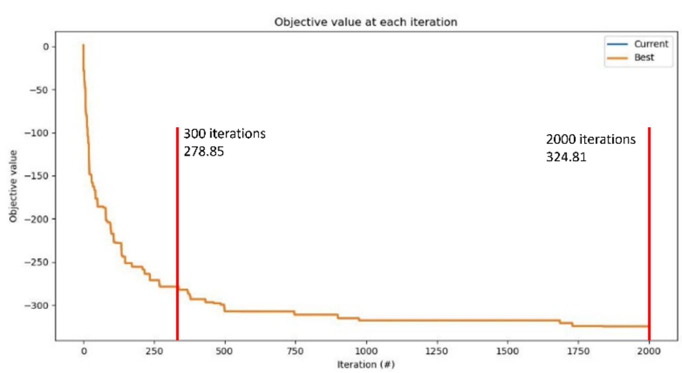


Figure 3: Search Progress

4.3 Dynamic Job Acceptance Performance

Our base scenario is a shift from 10am to 6pm from Toa Payoh Hub to Singapore Management University (SMU). Starting off with an initial value objective value of $278.85, our team simulated a dynamic jobs scenario for 8 hours shift. The ALNS was invoked a total of 30 times between the start and end time when new batches of jobs become available.

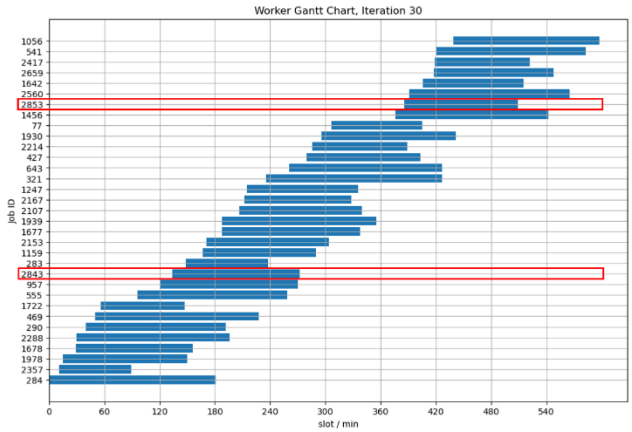


Figure 4: Gantt Chart with new jobs accepted

During the 9th and 10th run of the ALNS, new jobs (2843 & 2853) were found to be suitable and were added to the jobs for the driver. This further improved the objective value to 292.57 which was 5% better than the result from ALNS initial solution of 278.85.

4.4 Sensitivity Analysis

To understand the performance of our model better, our team performed sensitivity analysis to examine how changes to the rider's preferences influence the objective value of our solution.

4.4.1 Scenario 1: Different Shift Durations

In this scenario, we ran ALNS on different shift timings with dynamic jobs. As expected, we see the pre-match earnings drop due to the lesser work hours available to assign jobs. While no new jobs were picked due to strict constraints. We observe improvement to scheduling such that objective value improved due to better scheduling.

Table 3: Objective value with different shift durations

Shift timing	**Shift Start Location**	**Shift End Location**	**Pre-Match**	**Dynamic**
10am – 1pm	Toa Payoh	SMU	\$112.28	\$112.28
1pm – 6pm	Toa Payoh	SMU	\$176.72	\$177.34

4.4.2 Scenario 2: Different Shift Start and End Locations

In this test scenario, we considered different shift start and end locations to see how it will affect the schedule and route that our solution will suggest, and consequently the best objective value achievable. Table 4 outlines the 3 scenarios we created in addition to the baseline scenario. The remaining rider inputs are kept constant per the baseline scenario.

Table 4: Shift Start and End Locations used for sub-scenarios.

	Shift Start Location	**Shift End Location**
Central (Baseline)	Toa Payoh Lat, Lon: 1.3323, 103.8474	Singapore Management University Lat, Lon: 1.2989, 103.8455
Pan-Island (West to East)	Jurong East Lat, Lon: 1.3479, 103.7303	Pasir Ris Lat, Lon: 1.3740, 103.9371
Residential	Woodlands	Woodlands

	Lat, Lon: 1.4354, 103.7748	Lat, Lon: 1.4354, 103.7748
Central Business District (CBD)	Raffles Place Lat, Lon: 1.2843, 103.8510	Raffles Place Lat, Lon: 1.2843, 103.8510

Table 5: Results for different sub-scenarios

	Central (Baseline)	**Pan-Island**	**Residential**	**CBD**
Pre-match phase (after 300 iterations of ALNS)				
No. of jobs accepted	31 jobs	28 jobs	23 jobs	24 jobs
Best objective value	$278.85	$229.59	$224.34	$232.24
Penalty	$6.73	$31.34	$6.25	$24.06
Fuel Cost	$43.70	$49.72	$47.70	$44.36
Dynamic phase (by the end of the 8-hour shift)				
No. of jobs accepted	33 jobs	28 jobs	24 jobs	27 jobs
Best objective value	$302.61	$229.61	$226.90	$266.93
Penalty	$6.73	$31.34	$6.25	$16.29
Fuel Cost	$44.01	$49.70	$49.14	$46.42

We observed that the best objective value is lower for all three additional scenarios as compared to the baseline scenario. We investigated the generated schedules and found that our solution proposed for the rider to travel to and work in the same central region for most of the shift duration for all four scenarios (see **Annex B**). This is because short-distanced point-to-point parcel deliveries are concentrated in the central region in our dataset (see **Annex A**), hence, the solution will route the rider to work mainly in the central region to fulfil more jobs with less travelling. As a result, shift start and end locations in the central region will give rise to higher revenues since the rider need not incur additional travel time from and to these locations.

4.4.3 Scenario 3: Multi-apping vs single-platform

In this test scenario, the rider started at Toa Payoh Hub at 10am and ended at SMU at 6pm. We observe that multi-apping consistently returned a higher objective value compared to single-platform scenarios. This could be attributed to the higher number of better choices available in the multi-apping scenario, hence resulting in a better solution.

The difference in objective value between a multi-apping scenario and individual platforms varied greatly. J&T Express had the lowest objective value where multi-apping would result in 297% more profit at the end of the shift. Pickupp returned the best single-platform objective value where multi-apping would result in 45% more profit at the end of the shift. As the jobs were randomly generated with no observable trends (refer to the details of the jobs in **Annex A**Annex A), the disparity could have likely stemmed from platform-specific payout calculation strategies. While multi-apping consistently outperformed single-platform solutions, the difference in profits is platform-sensitive, and drivers may choose to trade-off with other measures such as penalties incurred.

Table 6: Results for multi-apping and single-platform runs

	Multi-apping (baseline)	**GrabExpress**	**Lalamove**	**J&T Express**	**Pickupp**
Payout calculation	Various	Distance	Distance	weight	Distance + weight
Number of jobs available	2,759	668	692	691	706
Pre-match phase (after 300 iterations of ALNS)					
No. of jobs accepted	31 jobs	20 jobs	20 jobs	23 jobs	23 jobs

Best objective value	$278.85	$80.86	$196.15	$39.86	$208.92
Penalty	$6.73	$3.30	$20.02	$1.16	$5.65
Fuel Cost	$43.70	$44.17	$48.18	$40.90	$43.90
<u>Dynamic phase</u> (by the end of the 8-hour shift)					
No. of jobs accepted	33 jobs	23 jobs	22 jobs	25 jobs	23 jobs
Best objective value	$302.61	$93.92	$199.73	$76.22	$208.92
Penalty	$6.73	$11.56	$22.02	$1.16	$5.65
Fuel Cost	$44.01	$45.96	$48.92	$39.18	$43.90

4.4.4 Scenario 4: Different rest constraints

In this test scenario, we simulated different work-rest strategies to observe the effect of length of breaks and continuity of breaks whilst keeping the rest of the variables constant.

Table 7: Different work-rest strategies

	Minimum Rest Time (min)	**Maximum Work Duration (min)**
Baseline scenario	15	120
3:2 work-rest ratio	120	180
7:2 work-rest ratio	60	210
No rest needed	0	480

Table 8: Results for different work-rest strategies

	Baseline	**3:2 work-rest ratio**	**7:2 work-rest ratio**	**No rest needed**
<u>Pre-match phase</u> (after 300 iterations of ALNS)				
No. of jobs accepted	31 jobs	18 jobs	29 jobs	30 jobs
Best objective value	$278.85	$169.30	$242.75	$265.83
Penalty	$6.73	$16.86	$18.25	$0.00
Fuel Cost	$43.70	$35.46	$43.67	$54.74
<u>Dynamic phase</u> (by the end of the 8-hour shift)				
No. of jobs accepted	33 jobs	19 jobs	33 jobs	32 jobs
Best objective value	$321.61	$190.90	$344.47	$285.21
Penalty	$6.73	$16.86	$18.25	$0.00
Fuel Cost	$44.98	$36.50	$45.31	$58.29

In comparison to the baseline (total of 45mins break), the 3:2 work-rest ratio strategy only allows for the driver to receive 60% of the profit earned in the baseline strategy, despite working 83% the duration. The 7:2 work-rest ratio strategy allows for the driver to receive very similar profits, predictably so due to the similar duration of break taken. Interestingly, for the 'no-rest' strategy, the profits earned is lower during the pre-match phase and the dynamic phase. Despite working for longer hours and having not incurred any penalty, the model panders to jobs with lower pay-outs to be assigned.

5 Findings and Insights

5.1 Comparison with Real-world Conditions

5.1.1 Traffic Conditions

In our model, we used Project OSRM Table Service API instead of Google Maps Distance-matrix API due to the costs associated with the latter. This can introduce some degree of variation from real-world conditions as traffic conditions are deemed to be fair in our implementation.

5.1.2 Varying pay-out structures

Despite our team's best efforts to replicate the pay-out structure currently used by delivery platforms, 'surge-pricing' – a pricing strategy that varies greatly from the usual pay-out structure employed when riders are few, was not implemented in our model.

5.1.3 Frequency and distribution of jobs

Our team generated a copious number of jobs and made efforts to model the distribution based on experience – adding more delivery jobs during lunch breaks and after work hours. However, the actual frequency and distribution of our dataset might still vary from real-life conditions. This could have led to a higher-than-expected performance of our model due to the high number of jobs available to select from.

5.2 Future Works

Overall, our solution proved a case for multi-apping: multi-apping consistently performed better than single-platform scenarios, and our built-in rest constraints resulted in a better objective value (and driver's well-being) compared to a no-rest scenario. Contrary to the belief that accepting more jobs would result in overcommitment and late deliveries, our solution was able to keep penalties at a minimum. With more choices from different platforms, the driver can choose jobs that are more suited for his schedule to minimize late deliveries. However, we also propose further enhancements below to improve our solution.

5.2.1 Additional Destroy and Repair Functions

Our team present new operators as follow:

5.2.1.1 Destroy Chain

Here we destroy a chain of jobs by specifying chain length. A random starting point is chosen in the drivers existing schedule, and we remove a job followed by the next adjacent job until the number of jobs removed matches the chain length.

5.2.1.2 Destroy Furthest Job

This operator uses a distance parameter to target jobs with pickup/drop-off locations far from driver's start and end point to create larger time gaps for insertions of better jobs that are closer in proximity to optimize time and fuel cost.

5.2.1.3 Repair By Regret (Ropke & Pisinger)

Since greedy algorithm is myopic in nature, we should consider the improvement in objective value for all jobs before assignment. This is done via computing the regret value for all jobs before finally applying the greedy assignment to add the job with highest regret value.

5.2.1.4 Repair By Weight

Weight is also a factor to be considered. We can sort jobs by their payload weight and assign the lightest job first, allowing drivers to potentially take on more jobs simultaneously. This can improve the objective value by reducing the need for drivers to complete one job before accepting another.

5.2.2 Improved modelling of real-world conditions

Assuming this project is deemed to be viable with a healthy amount of interest, frequency distributions and assumptions in the model can be replaced with real empirical data in the following ways:

1. Replacement of Project OSRM's Table Service API to another API that can take into consideration traffic conditions to better allow this tool to cater to delivery drivers.
2. Partnering up with delivery platforms to gain access to historical and streaming job orders to ensure no over-generation of jobs and accurate pay-out structure for each order.

References

Barrat, T. G. (2020). 'I'm my own boss...': Active intermediation and ‘entrepreneurial’ worker agency in the Australian gig-economy. *Environment and Planning A: Economy and Space, 52*(8), 1643-1661.

Dollars and Sense. (2021, November 30). *How much can you earn as a Lalamove driver in Singapore?* Retrieved from Dollars and Sense: https://dollarsandsense.sg/much-can-earn-lalamove-driver-singapore/

Griffith, E. (2022, October 11). *Meet the app that helps gig workers know how much they really make.* Retrieved from The New York Times: https://www.nytimes.com/2022/10/11/technology/gig-workers-drivers-para-app.html

J&T Express. (2021, January 11). *J&T Express Singapore.* Retrieved from https://www.facebook.com/jtexpresssg/photos/a.126094425413209/428314811857834/?type=3

Kim, J. (2020, March 30). *GrabFood and GrabExpress for private-hire Grab drivers.* Retrieved from First Lane: GrabFood and GrabExpress for Private-Hire Grab Drivers

Ministry of Manpower. (2021, January 28). *Labor force in Singapore 2020.* Retrieved from https://stats.mom.gov.sg/iMAS_PdfLibrary/mrsd_2020LabourfForce.pdf

OSRM API Documentatio. (n.d.). Retrieved from Project OSRM: http://project-osrm.org/docs/v5.5.1/api/#table-service

Pickupp. (2023). *Delivery services FAQ.* Retrieved from Pickupp: https://sg.pickupp.io/en/faq

Ropke, S., & D, P. (2006). A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research*, 750-775.

Ropke, S., & D, P. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 455-472.

Annex A Estimating payouts for parcel delivery jobs in Singapore

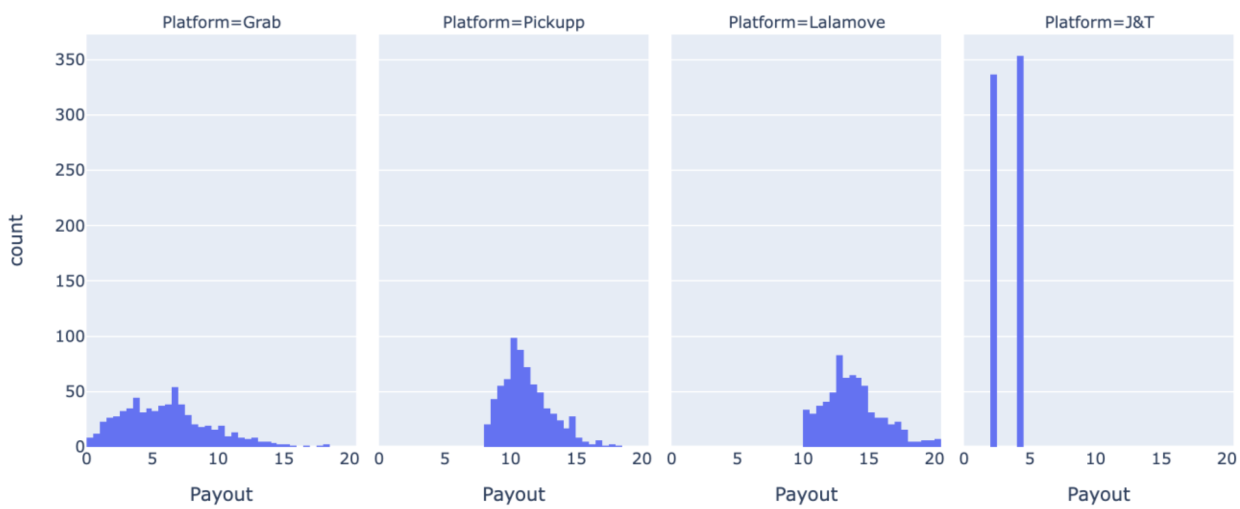
Payout was calculated using

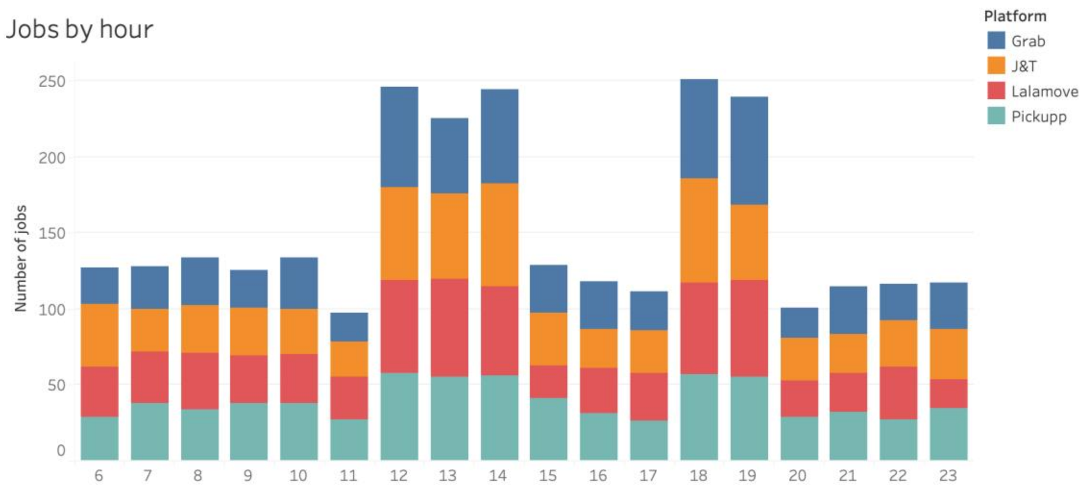
$$\text{payout} = \text{estimated rate} - \text{driver's commission}$$

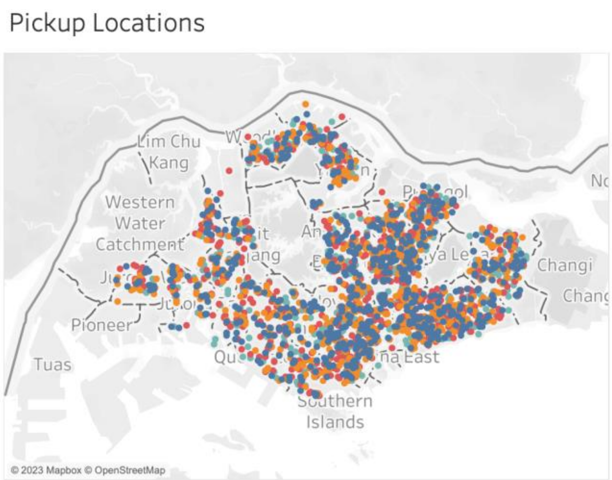
where the estimated rate was calculated referencing the company's publicly available rates compiled below

Platform	**Schedule of rates** (as of 23 March 2022)	**Driver's commission**
GrabExpress (Kim, 2020)	Base fare \$9 + \$1.20/km (first 5km) + \$0.70/km (next 5km)	80% of the job
Lalamove (Dollars and Sense, 2021)	Base fare \$12 (under 1km) + \$1/km (first 3km) + \$0.45/km)	84% of the job
Pickupp (Pickupp, 2023)	4-hour delivery • Below 5kg: \$10 +\$0.40/km • Below 10kg: \$12 +\$0.40/km • Below 15kg: \$15 +\$0.40/km • Below 20kg: \$17 +\$0.40/km	80% of the job
J&T Express (J&T Express, 2021)	• Below 5kg: \$1.49 + \$1.41 • Below 10kg: \$1.49 + \$3.51 • Below 15kg: \$1.49 + \$6.10 • Below 20kg: \$1.49 + \$13.41	80% of job (estimated)

Based on the above, the resulting simulated jobs had the following distributions




Jobs by hour
Platform
Grab
J&T
Lalamove
Pickupp
Number of jobs
0
50
100
150
200
250
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23


Pickup Locations
Lim Chu Kang
Western Water Catchment
Changi
Pioneer
Tuas
Southern Islands
© 2023 Mapbox © OpenStreetMap

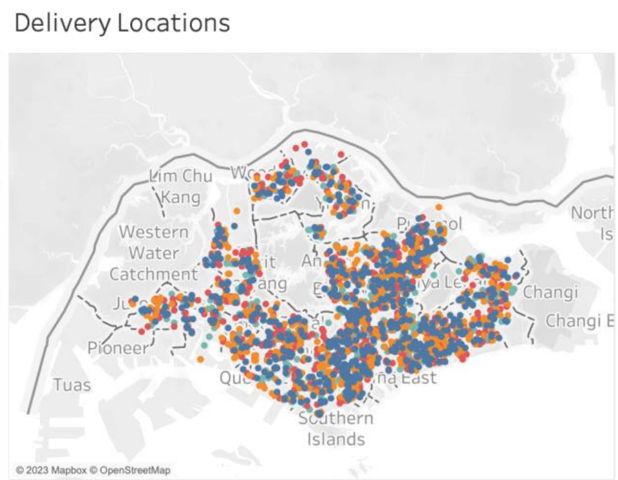
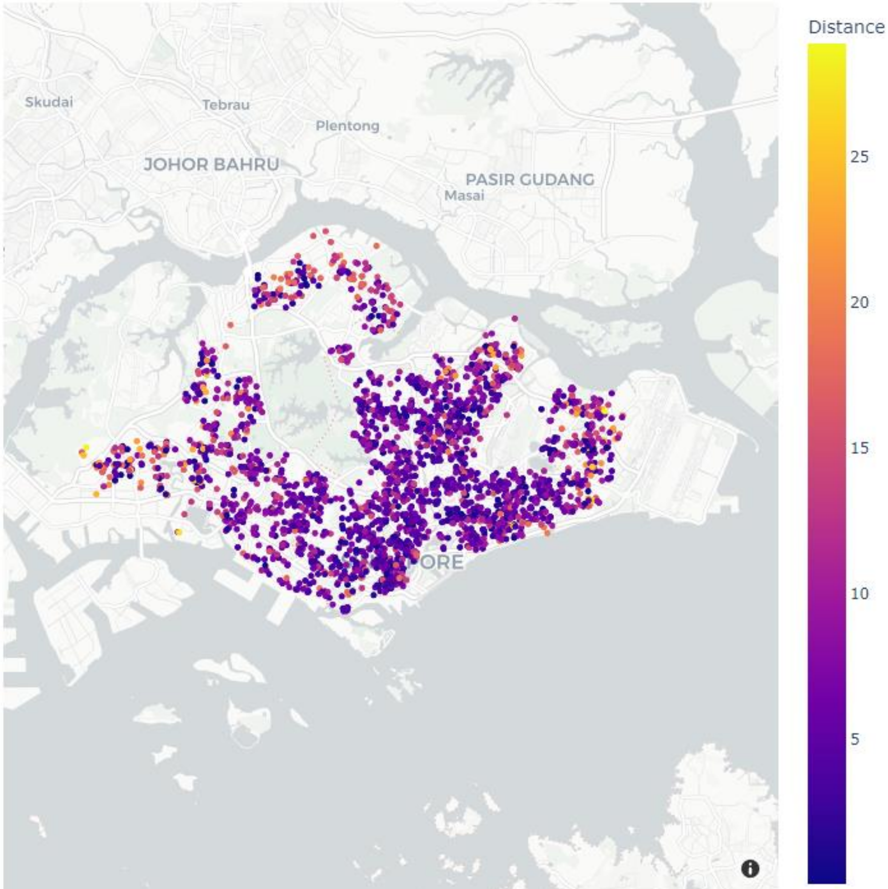

Delivery Locations
Lim Chu Kang
Western Water Catchment
Changi
Pioneer
Tuas
Southern Islands
© 2023 Mapbox © OpenStreetMap

Figure 5: Jobs (Delivery Locations) coloured by the distance between pickup and delivery locations



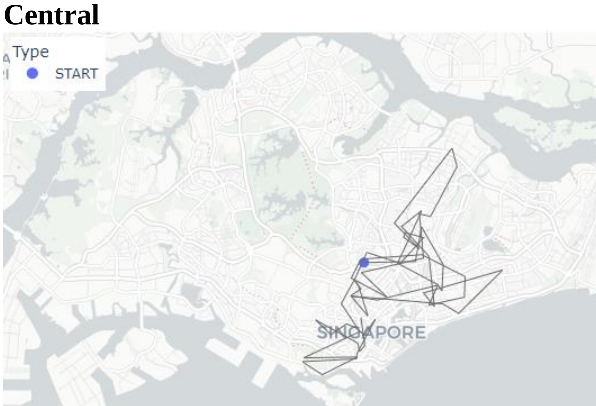
From Figure 5, we can see that most of the short-distanced point-to-point jobs are concentrated in the central region of Singapore.

Annex B Schedule Generated for Scenarios with Varying Start and End Locations

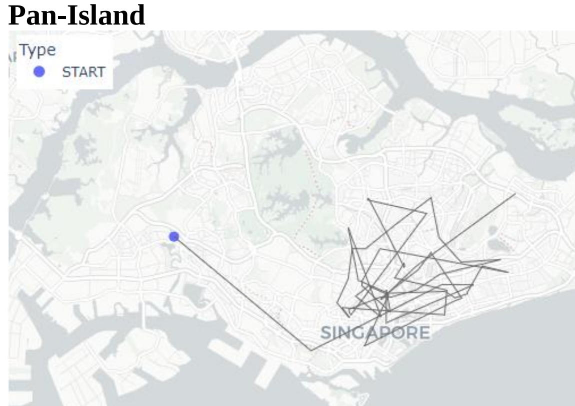
	Central (Baseline)	Pan-Island	Residential	CBD
Pre-match phase (after 300 iterations of ALNS)				
No. of jobs accepted	31 jobs	28 jobs	23 jobs	24 jobs
Best objective value	$278.85	$229.59	$224.34	$232.24
Penalty	$6.73	$31.34	$6.25	$24.06
Fuel Cost	$43.70	$49.72	$47.70	$44.36
Dynamic phase (by the end of the 8-hour shift)				
No. of jobs accepted	33 jobs	28 jobs	24 jobs	27 jobs
Best objective value	$302.61	$229.61	$226.90	$266.93
Penalty	$6.73	$31.34	$6.25	$16.29
Fuel Cost	$44.01	$49.70	$49.14	$46.42

The routes generated for the four scenarios are shown below. The algorithm will direct the rider towards the central region of Singapore for most of the shift duration.

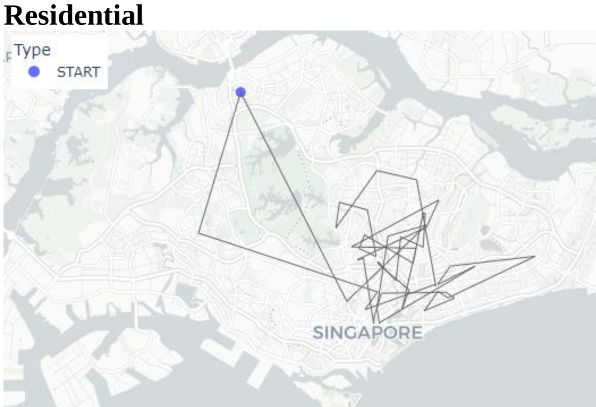
Central



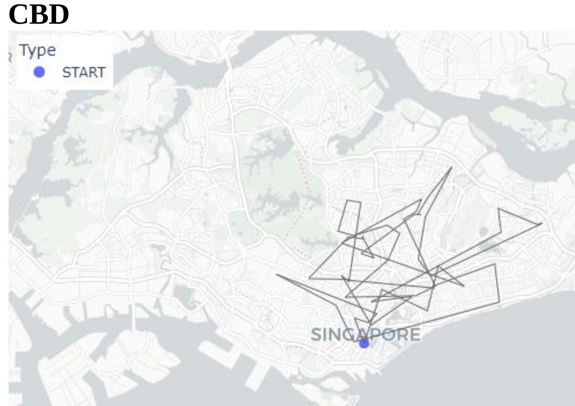
Pan-Island



Residential

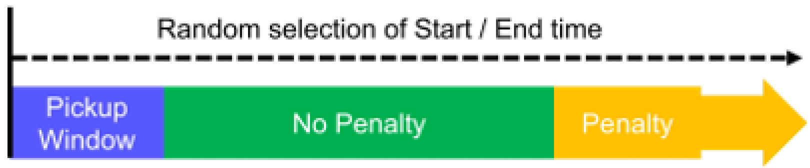


CBD

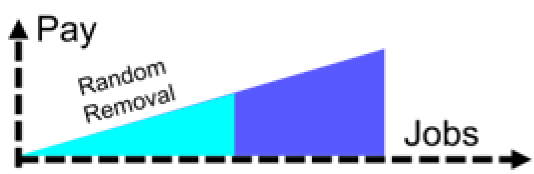


Annex C Diagrams for repair and destroy operators

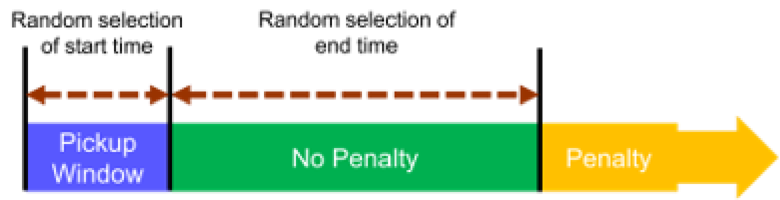
Selection of Pickup & Delivery timeslots



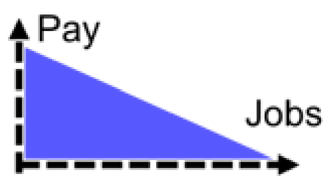
Destroy operator – Remove a job from bottom 20% of jobs payout value



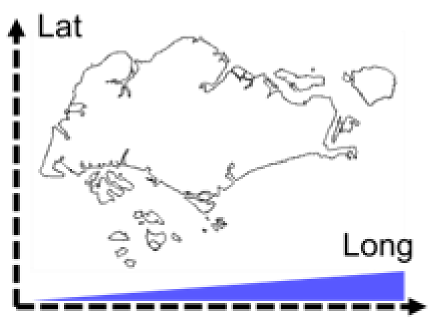
Repair operator – Repair without penalty



Repair Operator – Greedy insertion of jobs with best pay first



Repair Operator – Repair by longitude



Annex D Dynamic Job allocation diagram

The following diagram depicts visually the dynamic job assignment process.