Optimization of an Illegal Food Delivery Service in a University Campus

Final Report

CIE 5015 Operations Research

Submitted by:

Sukrityranjan Samanta (r09521127)
廖浚評 (r10521516)
Luis Fernando Cordazzo Vargas (r10521534)
Joshua Philip Suarez (r10521535)
Budiman Halim (b07501083)
Christian Kurniawan Bambang (b07501106)

Submitted to:

Prof. Albert Chen

Contents

About the Problem

Due to a so-called “unfair competition”, the association of restaurants inside NTU pushed to prohibit delivery inside the campus by the evil and capitalist brands Food Koopa and Bowser Eats. Understanding that this restriction created an emergent market for smuggled food, Mario and Luigi plan to sell hamburgers to a list of customers inside the campus previously contacted online. To be successful, the brothers need to choose the best route to deliver the food to customers in different buildings before the police inside the campus catches them. Each has his own vehicle and may traverse different routes before exiting the campus. Bikes and cars are considered for delivery, each with different speeds, capacities and unit cost (Table 1).

Table 1. Proposed delivery configurations in the study

	Mario	**Luigi**
Case 1	Bike	Bike
Case 2	Car	Car

This mission is initially deemed to be insurmountable given the very tight security inside the university campus. Understanding this challenge, Mario and Luigi hired the services of Yoshi to help disable all security cameras and remove all batteries from the security’s two-way radios. As a result, the average response time of the police increased from 5 minutes to 20 minutes. This means that Mario and Luigi each have to finish their entire trip and leave the campus within this threshold; otherwise, they will be sent to jail where the evil Wario is waiting.

In addition, Mario and Luigi want to avoid the possibility of being trapped inside the university campus when the police decide to close the gates. As such, they asked Princess Peach to serve as a watchdog and possibly distract the police with her beauty at their chosen entry point, the university main gate. The brothers must begin and end their trip at the exact same portal to ensure their escape.

The potential delivery points are shown in **Figure 1** below. The estimated demand is visualized in **Figure 2**.

Figure 1. Potential delivery points within the NTU campus

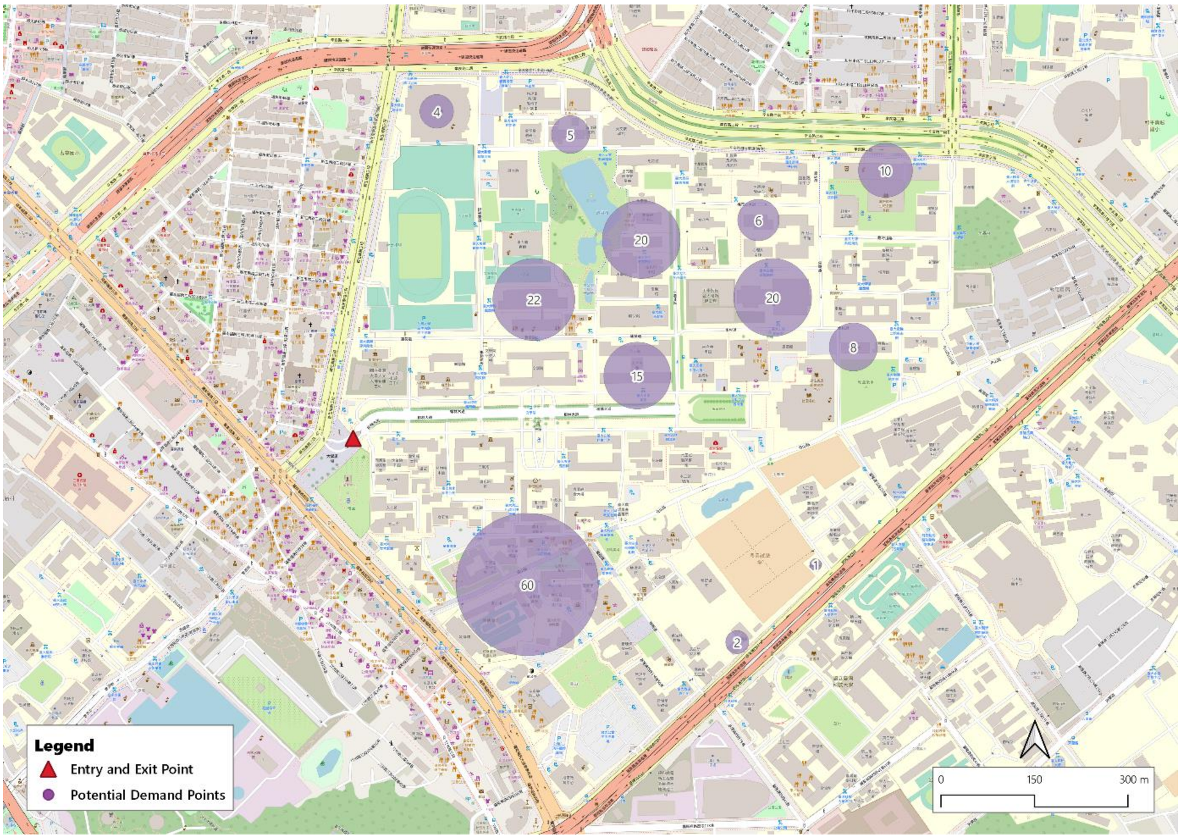


Figure 2. Estimated demand for hamburgers within the NTU campus

The problem was considered interesting for the researchers because of three main reasons. First, the problem was seen as a real-life and relatable application in the transportation context, a field where most of the members of the group are currently aiming to specialize at. In addition to its potential societal impact, undertaking the project would provide the students a better understanding of the logistics field and equip them with decision-making skills when they someday take on bigger real-world problems. Second, the concepts of the multiple travelling salesman problem (mTSP) and the multiple vehicle routing problem were not deliberately discussed in class. As such, the researchers found it challenging to solve and go beyond what was learned during the homework and exams. Finally, the problem's main characters (Mario and Luigi) and its context (i.e., being involved in a vehicle chase with the police) brought back memories to those who have spent a significant time of their childhood playing video games.

Literature Review

The concept of the travelling thief problem (TTP) has initially been examined in the project. The problem is a combination of two well-known optimization problems (TSP and KP) *[Faulkner et al. 2015]*. The study of *Faulkner et al. 2015* consisted of a set of cities, each containing a set of available items with weights and profits with the objective to maximize the resulting profit. *Laporte 1980* summarized different approximate algorithms developed for the Vehicle Routing Problem (VRP). TSP is one kind of VRP. The advanced form of TSP is multiple traveling salesman problem (mTSP). *Bektas 2006* has provided an overview of formulations and solution procedures for mTSP. The characteristics of mTSP seem more appropriate for real-life applications, and it is also possible to extend the problem to a wide variety of VRPs by incorporating some additional side constraints *[Bektas 2006]*.

Bonyadi et al. 2013 mentioned that two main characteristics of combination and interdependence are important for defining the sub-problems of TTP, where two sets of parameters are introduced that result in generating two instances of TTP. For this case, two models of TTP are introduced and their characteristics are investigated. These two models are different from each other based on the way in which the sub-problems are interdependent. It is shown that solving each sub-problem in isolation is not effective and the two sub-problems have to be considered together.

Hansknecht et al. 2021 depicted the formulation for time-dependent traveling salesman problem (TDTSP). Time dependency to this study has been controlled by velocity of selected vehicles to traverse between nodes of specific distance. Additional time constrained is maintained by restricting the total travel time from the node of initiation. It was shown that even if the TSP and the knapsack are solved to optimality, the final solution is not necessarily the optimal solution for TTPs. There are many other parameters and constraints that can be added to a TTP to reflect the real-world characteristics. Although one might not be able to find direct applications of TTP in the real world, the problem reflects one of the main complexity sources (interdependency between sub-problems) in real-world problems. Also, it is quite unclear which approaches can be more beneficial for solving TTPs with different types of interdependencies.

There are n cities, and the distance matrix $D = \{d_{ij}\}$ is given. Also, there are m items each of them having a value p_k and weight w_k. There is a thief who is going to visit these cities exactly once and pick some items from the cities and fill his knapsack. The maximum weight for the knapsack is W. The aim is to find a tour that visits all of the cities exactly once and gets back to the starting city, optimizing objective function(s) while the total weight of the knapsack is not violated.

Objectives

The study aims to maximize the net income for an illegal food delivery service inside a university campus. The net income is defined as the difference between the gross profit (i.e., the product between the unit cost of hamburger and demand catered) and the total costs incurred (i.e., the product between the unit cost associated with the vehicle and total distance travelled).

In order to achieve this, the model must be able to determine the most optimal way – which route must each vehicle traverse? Which mode of transport shall be ideal, provided that each mode presents a tradeoff when it comes to capacity, time, and cost? – that will yield the greatest net profit.

While the problem was formulated in a bizarre and odd way to spark interest, the researchers believe that the study has real-life implications underneath. For instance, the proposed model has the potential to help small-time food business owners kickstart their expansion through a local delivery service.

Since it would be too costly for small businesses to partner with existing online food platforms, these types of optimization models would aid the owners maximize the efficiency of their limited fleet. In addition, rural and remote areas are not covered by existing food platforms and our model has the potential of helping them come up with the design of a small-scale delivery service within their neighborhood. Moreover, student organizations planning to sell food within their campus for their fundraising initiatives would seamlessly benefit from our program and may solve their routing issues with a simple changing of the parameters.

Similar to the initial problem, real-life food delivery services would be bounded by time constraints. However, instead of the possibility of being caught by the police, these will be in the form of (1) the maximum acceptable waiting time for customers and/or (2) the maximum time before the quality of food diminishes. Furthermore, the study is deemed to be more relevant now that cities are actively pushing for sustainable modes of transport, and bicycles have now been utilized for delivery services.

Methodology

The researchers aim to capture a routing process that is as close as possible to the real-world situation. As such, extensive research was initially done to identify the required parameters (e.g., capacity, unit cost and velocity of bikes and cars) in the program. For example, (1) the researchers computed the distance between each node using Google Maps, and (2) the university speed detector was observed to find reasonable speeds for each type of vehicle.

Once the model has been formulated and programmed in *Python* and *Gurobi*, various test runs were conducted to (1) perform a baseline estimation of the validity and accuracy of the result, and (2) capture real-world implication using different what-if scenarios.

Finally, once the program has been fine-tuned, an actual simulation was conducted (albeit without the chasing police officers) to verify the validity of the program. Should there be any further discrepancy, the parameters will then be updated. The iterative process is shown in **Figure 3** below.

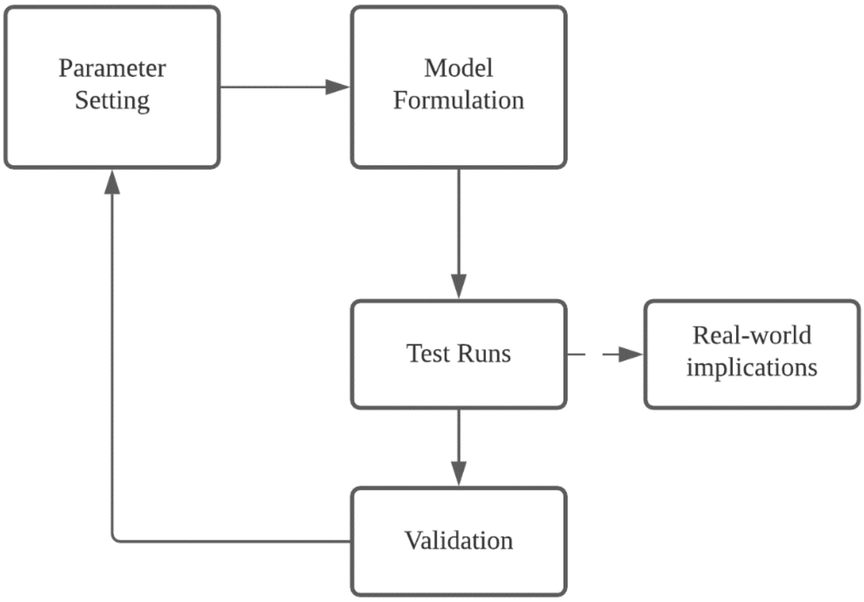


Figure 3. General framework adopted by the study

<u>Mathematical Model</u>

Parameter, Variable, and Model Description

A) Parameter Description

- $P_k : Vehicle\ Capacity$
  $(P_k = [P_{kc}, P_{kb}], \qquad \text{k} \in [\text{car}, \text{bike}])$
- $\text{C}_\text{k} : Vehicle\ Cost\ per\ Unit\ Distance$
  $(C_k = [C_{kc}, C_{kb}], \qquad \text{k} \in [\text{car}, \text{bike}])$
- $V_k : Vehicle\ Velocity$
  $(V_k = [V_{kc}, V_{kb}], \qquad \text{k} \in [\text{car}, \text{bike}])$
- $D_{ij} : Distance\ Between\ Node\ i\ and\ j$
  $\left((i, j) \in (N, N), \qquad D_{ij} \in \text{Z}\right)$
- $P : Price\ of\ a\ Burger\ (P = 120)$
- $Q_i : Demand\ at\ Node\ i$
  $(i \in N, \qquad Q_i \in \text{Z})$
- $T : Vehicle\ Maximum\ Allowable\ Travel\ Time$
  $(T = [T_{kc}] = [T_{kb}], \qquad \text{k} \in [\text{car}, \text{bike}])$
- $L = Maximum\ Allowable\ Node\ Visit\ for\ a\ saleman$

- $K = Minimum\ Allowable\ Node\ Visit\ for\ a\ saleman$
-

B) Variable Description
- $X_{ij} : Route\ From\ Node\ i\ to\ j\ Passed$
  $\left((i,j) \in (N,N), \qquad X_{ij} \in (0,1)\right)$
- $uX : Subtour\ Elimination\ Variable$
  $\left((x) \in (N), \qquad uX \in (0,1)\right)$

C) Model Description
- Objective Function

$$\left(Max\ Z = \sum_i \sum_j \left(X_{ij} \cdot (Q_i \cdot P - (D_{ij} \cdot C_k)\right)\right)$$

- mTSP Constraint

$$\begin{cases} \sum_i X_{ij} <= 1\,, for\ j \in (1,N] \\ \sum_j X_{ij} <= 1\,, for\ i \in (1,N] \\ \sum_i X_{0i} == 2\ for\ i \in (1,N] \\ \sum_i X_{i0} == 2\ for\ i \in (1,N] \end{cases}$$

- Miller-Tucker-Zemlin Subtour Elimination Constraint
- $(uX_i + L - 2 \cdot X_{0i} - X_{i0} \leq L - 1, \quad 1 \leq i \leq N)$
  $(uX_i + X_{0i} + (2 - K) \cdot X_{i0} \leq 2, \quad 1 \leq i \leq N)$
  $\left(uX_i - uX_j + L \cdot K \cdot X_{ij} + (L - 2) \cdot X_{ji} \leq L - 1, \qquad 1 \leq i \neq j \leq N\right)$

- Time Constraint

$$\sum_j \sum_i X_{ij} \cdot D_{ij}/1000 \leq \sum_k T_k \cdot V_k$$

- One In and One Out except Entry

$$\sum_{j=1} \sum_i (if\ \ X_{ij} == 1\ \ then \sum_i Xji == 1)$$

- Vehicle Capacity Constraint

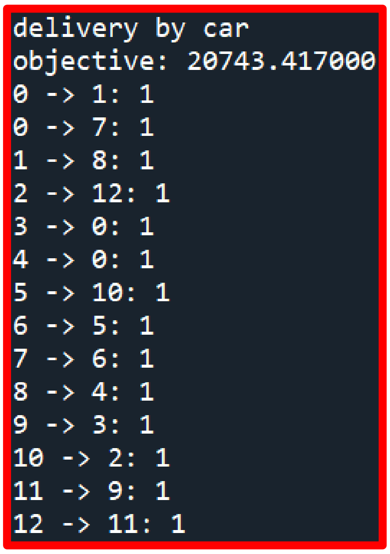
$$\sum_j \sum_i X_{ij} \cdot Q_i \leq \sum_k P_k$$

For this project an objective function to maximize the profit obtained through the sale of hamburgers was chosen. The idea behind the algorithm is to allow sufficient flexibility without compromising the obtention of a solution caused by the addition of excessive number of constraints. The researchers consider useful for future users, other than Mario and Luigi, to develop a program that is open to obtain the best solution from operations between two different modes and their attributes (speed, capacity, and operational costs) in an environment where time constraints can be set and adjusted along different distance matrixes.

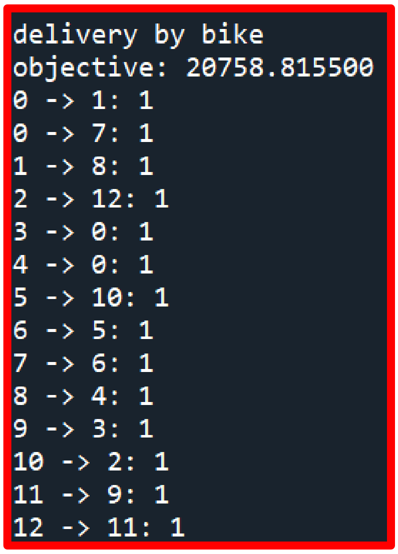
Modelling Results and Discussion

The algorithm not only served its purpose that was to calculate the best route and vehicle for the designed problem, but it ended up being a useful tool to calculate many what-if scenarios. Some are described below:

- Interestingly, by using the following parameters for cars, $P_k = 500$, $C_k = 0.0035$ (to account for distance in meters), $V_k = 20$, and for bikes $P_k = 200$, $C_k = 0.00025$, $V_k = 15$, as well as the general $P = 120$ and $T = 20/60$, the following results were obtained:



```
delivery by car
objective: 20743.417000
0 -> 1: 1
0 -> 7: 1
1 -> 8: 1
2 -> 12: 1
3 -> 0: 1
4 -> 0: 1
5 -> 10: 1
6 -> 5: 1
7 -> 6: 1
8 -> 4: 1
9 -> 3: 1
10 -> 2: 1
11 -> 9: 1
12 -> 11: 1
```

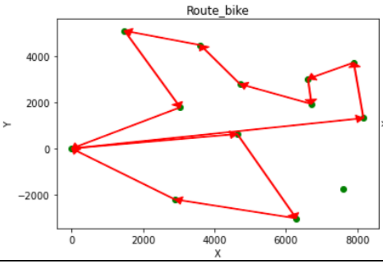


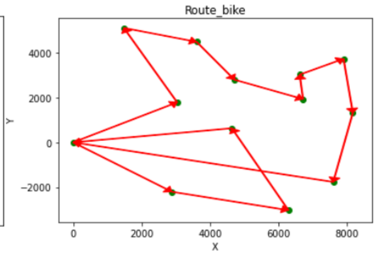
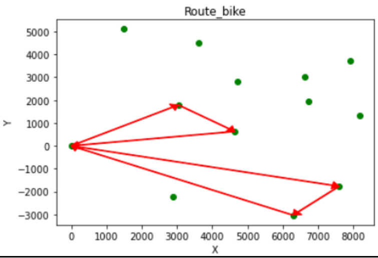
```
delivery by bike
objective: 20758.815500
0 -> 1: 1
0 -> 7: 1
1 -> 8: 1
2 -> 12: 1
3 -> 0: 1
4 -> 0: 1
5 -> 10: 1
6 -> 5: 1
7 -> 6: 1
8 -> 4: 1
9 -> 3: 1
10 -> 2: 1
11 -> 9: 1
12 -> 11: 1
```

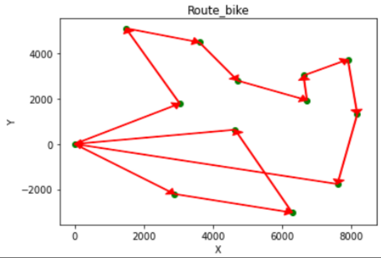
The similar results can be attributed to the small operational costs due to the limited space and the comparable speeds as a result of a in-campus restriction of 20 km/h stablished for cars. In a real life situation, other factors will be included such as the risk of having the car confiscated after police investigation.

- By playing with the time constraints, different best routes were discovered. **Table 2** summarized the effect of different parameters.

Table 2. Summary of the effects of different parameters adopted by the study

<table>
<tr><th>Parameter</th><th>Effect</th></tr>
<tr><td>Operational Cost Ck</td><td>If time and capacity constraints are eliminated by setting them very high, the NTU farm was left out of the route when Ck is 1 per unit distance, but it was included when Ck was reduced to 0.1 as shown below.



</td></tr>
<tr><td>Vehicle Capacity Pk</td><td>Capacity influenced decision making as well. The following pictures represent route choice with setting Pk to 40 and 200 respectively. Mario and Luigi can use this information to make changes to improve the capacity of their fleet. An old-fashioned front basket for their bikes looks like a good investment for the business.



</td></tr>
<tr><td>Maximum Allowable Travel Time Tk and Vehicle Speed Vk</td><td>Values for Tk and Vk can be adjusted to reflect every scenario. For this project, speeds of 15km/h and 20 km/h were chosen for bicycles and cars respectively. Smaller values were tested to represent the time lost during the transaction.</td></tr>
</table>

- The elimination of the time constraint allows the interested parties to obtain the best route to satisfy all the customers. This information may motivate our infamous brothers to obtain the permit to operate legally and maximize their earnings be serving all the interested customers without worrying about being caught.
- An excel file supplies the algorithm the distances between all the nodes. Different modes of transportation may have their own distances to account for dissimilarities in assigned routes. Excel matrixes are easy to update for future operations.
- A decreased operational speed was tested to account for the transaction time lost with satisfactory results. Additionally, the brothers are smart enough to avoid wasting time waiting for customers, there is the assumption that the customers are already waiting at the door for a smooth operation.

Once the problem has been solved, the students conducted an actual simulation of the delivery service to validate the results.

Validation

Extensive work was done by the researchers to ensure that the demand and travel time between each point were closer to real-life values. The demand was estimated based on the population serviced by the building (e.g., faculty members, staff, and students) and its adjacent areas. The distance between points were estimated using Google Maps, while the travel speed of bikes and cars were approximated based on the readings of the university's speed detector.

(a) speed of a passing car *(b) speed of a passing bike*

Figure 4. Sample output from the university speed detector

The vehicle parameters initially inputted in the program (for validation purposes) are shown in **Table 3** below.

Table 3. Parameters inputted for the validation of the program

Mode of vehicle	Capacity	Cost	Speed	Time	Distance
	# of burgers	*NTD/km*	*km/hr*	*hr*	*m*
	Pk	Ck	Vk	T	1000*T*Vk
Car	400	3.5	20	0.33	6666.66
Bike	200	0.25	10	0.33	3333.33

The justification and thought process behind the parameter setting are as follows:

- **Capacity** – Capacity is a major difference between cars and bikes. The value inserted in Pk is the total capacity of both vehicles combined, it can be assumed that the bikes can transport an increased number of burgers thought the installation of additional equipment such as bags and baskets.
- **Cost**
    - For the car option: the cost of gasoline in Taiwan is estimated at 30 NTD/liter, while the fuel mileage of vehicle is approximately 10 km/liter. As such, the estimated unit fuel cost for a car is 3.0 NTD/km. An additional 0.5 NTD/km is added to account for miscellaneous car costs (e.g., maintenance cost, parking cost).

- For the bike option: only maintenance cost, estimated at 0.25 NTD/km, was considered

- **Speed** – the vehicle speeds were deliberately lowered because of the (1) university speed limits and the (2) need to account for the time spent at the nodes (I.e., in real-life, there will be additional time associated with the actual delivery of the products.

Given these parameters, the program yielded the following as the most optimal routes for delivery (**Figure 5**).

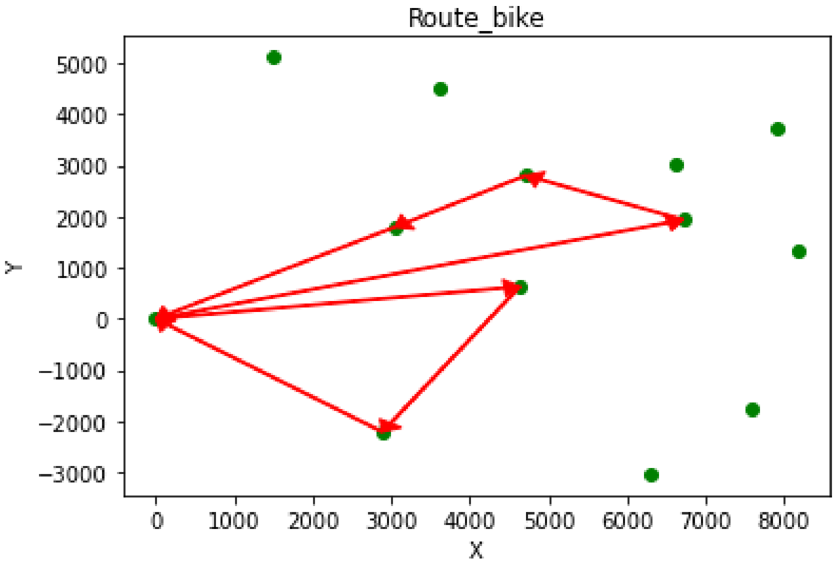


Figure 5. Bicycle routes for validation

Validation is an important part of every project, and this was the case in the team's experience. At first, the algorithm gave us a proposed route for the 20 minutes span, and it looked correct. However, the real test resulted in the following delivery time for the two loops, each representing the trajectory of one of the brothers (**Table 4**)

Table 4. Tabulation of actual biking time during validation

Loop	Link Description	Time (minutes, seconds)
1	NTU Main Gate – C.E. Building	2 min
	C.E. Building – 1st Women's Dorm	3 min, 10 sec
	1st Women's Dorm – NTU Main Gate	2 min, 15 sec
	Total	**7 min, 25 sec**
2	NTU Main Gate – Department of Mechanical Engineering	3 min, 30 sec
	Department of Mechanical Engineering – Freshman Classroom Building	1 min, 30 sec
	Freshman Classroom Building – Liberal Arts Building	1 min, 30 sec
	Liberal Arts Building – NTU Main Gate	2 min
	Total	**8 min, 30 sec**

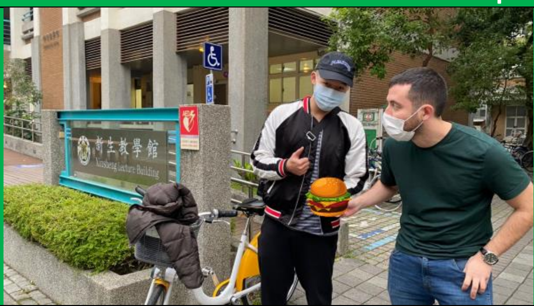
As can be seen, the program added the total time of both loops (total 15 min, 55 sec) and compared it with the time constraint of 20 min when in real life every brother would have 20 min on its own.

Start

Mario and Luigi planning to execute their illegal scheme

Loop 1: Mario's Adventure

Loop 2: Luigi's Path



End: The Great Escape

Mario and Luigi very happy after a successful day at the office – they were able to gain profit and fill students' tummies while evading the police

Lessons Learned and Future Work

The students learned to utilize the programming skills acquired in class to successfully solve a modified version of the multiple travelling salesman and multiple vehicle routing problem.

Initially, the researchers dreamt big and proposed the problem in a highly complex manner. The travelling thief problem became the starting point, mainly because it could serve as a good culmination to connect what we've learned in the TSP and knapsack problem. From this, a city-wide bank robbery problem was conceived, but was later revised due to concerns in validation and scale.

Even in its current context, the problem was gradually simplified due to limitations in time. Some of the initial (and interesting) constraints were believed to be achievable given enough time, but the researchers decided to focus on the constraints which were deemed essential to the problem context. The removed/adjusted constraints are as follows:

- **Time constraint**: time begins as soon as the first delivery is made (i.e., a bystander will report to the police once an illegal transaction is conducted)
- **Entry/exit point:** the characters may begin and end at various identified points around the campus
- **Accessibility constraint:** some areas around the campus are completely inaccessible to cars
- **Type of vehicle constraint:** the characters may choose a combination of bike and car (Case 3: Bike + Car)

The following are suggested by the authors as future extension/s of the problem:

- **Introduction of more demand points** – the current nodes represent the demand not just for a specific building, but for an identified cluster (e.g., the 1st male graduate dorm node represents the demand for all adjacent dorms). Disaggregating the nodes will represent a more realistic spatial distribution of demand.
- **Consideration of the effect of weight carried to the bicycle velocity** (i.e., the greater the weight carried, the slower the bicycle will be)
- **Consideration of other modes of transport including, but not limited, to horses** – horses may lead the brothers to being caught faster (i.e., riding horses will definitely catch a lot of attention), but at the same time will attract greater demand from the students (students love horses) which opens the possibility to increase gross profit.

Work Distribution

The work distribution is shown as follows:

Table X. Work distribution among the group members

Luis (19%)	**Joshua** (19%)	**Sukrit** (19%)	**Jun-Ping** (19%)	**Christian** (14%)	**Budiman** (10%)
Problem Context	Problem Context	Parameter Setting	Mathematical Model	Literature Review	Mathematical Model
Objectives	Objectives	Literature Review	Python/Gurobi Programming	Validation	
Validation	Validation	Validation	Results and Discussion	Mathematical Model	
Results and Discussion	Methodology				
Lessons Learned and Future Work	Lessons Learned and Future Work				
Compilation and Proofreading	Compilation and Proofreading				

Extra Credits

The researchers believe that they deserve extra credit because of two things:

- The problem context was creative and interesting
- Extensive validation was pursued to verify the program result

References

1. Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, *34*(3), 209-219.
2. Bonyadi, M. R., Michalewicz, Z., & Barone, L. (2013, 20-23 June 2013). The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. 2013 IEEE Congress on Evolutionary Computation
3. Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, *20*(1), 255-282.
4. Faulkner, H., Polyakovskiy, S., Schultz, T., & Wagner, M. (2015). *Approximate Approaches to the Traveling Thief Problem* Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain.
5. Hansknecht, C., Joormann, I., & Stiller, S. (2021). Dynamic Shortest Paths Methods for the Time-Dependent TSP. *Algorithms*, *14*(1), 21.
6. Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(3), 345-358.
7. Laporte, G., & Nobert, Y. (1980). A Cutting Planes Algorithm for the m-Salesmen Problem. *Journal of the Operational Research Society*, *31*(11), 1017-1023.
8. Oberlin, P., Rathinam, S., & Darbha, S. (2009, 10-12 June 2009). A transformation for a Heterogeneous, Multiple Depot, Multiple Traveling Salesman Problem. 2009 American Control Conference
9. Osman, M. F. S. (2018). Assignment-based MIP Modeling for Solving a Selling Firm mTSP with Time Limit Constraints. In *ICORES* (pp. 164-169).
10. Vali, M., & Salimifard, K. (2017, August). A constraint programming approach for solving multiple traveling salesman problem. In *The Sixteenth International Workshop on Constraint Modelling and Reformulation* (pp. 1-17).
11. 優化┃TSP中兩種不同消除子環路的方法及callback實現（Python呼叫Gurobi求解） - 知乎, MdEditor, https://www.gushiciku.cn/pl/p3hL/zh-tw