**Route Optimization Using the School Bus Routing Problem (SBRP): A Quantitative Case Study on School A**

AP Research

May 20, 2021

Word Count: 5498

**Abstract**

The school buses at School A, an international school in Tianjin, China, often arrive at school late in the mornings. Due to the traffic congestion level during rush hours and inefficiency in the current route planning process at School A, the researcher of this paper realizes that a quantitative case study using Python may help the school optimize its bus route in terms of cost and time. This study’s purpose is to collect data on the current bus route and improve it using the School Bus Routing Problem (SBRP), a routing method that considers the maximum capacity and travel time of each bus. Through computer programming and manually adjusting the routes from the computer program, two optimized routes were generated. For the first route, nine buses were required and the average travel time for each bus was 47 minutes. For the second route, the average travel time per bus was 50 minutes, but it only required eight buses, which reduced the bussing cost for School A. In addition, the method used in this paper also attempts to improve inefficiencies in the current route planning process by creating a computer program that automatically calculates the most optimized routes, making it easier to add or modify stops in the future. Furthermore, other international schools in China may also refer to this paper and implement the method to help them achieve bus route optimization. Future work may consider additional factors such as travel and safety regulations to further build upon this research project.

*Keywords*: route optimization, School Bus Routing Problem (SBRP), Vehicle Routing Problem (VRP)

**Introduction**

Public transportation has gone through significant improvements over the years, but traffic jams during rush hours are still major issues in urban areas. As traffic congestion level rises during rush hours, students take longer to get to school and sometimes even arrive late. Increasing the number of buses may reduce travel time for students, but it would also increase the cost. Therefore, school bus route planners need to minimize the amount of time students spend commuting while also balancing the cost required to run the school buses. To further complicate the problem, additional constraints need to be considered when planning for public transportations: travel distance, maximum capacity of the vehicle, varying travel time, etc.

The problem becomes infeasible to optimize by hand when the number of stops is large. For example, if a bus is to visit seven stops, there are, theoretically, 7! = 5,040 possible combinations. Due to the amount of calculation needed, a computer program written in Python is used in this paper to assist the researcher in computing an optimized route since the computer program can generate and run through all possible scenarios with consideration of variables such as travel time and maximum capacity.

In this project, the researcher was given 24 stops and the goal is to find the optimal number of school buses as well as their routes that minimize students’ travel time and school bus management cost. This problem can be described by a well-researched computer science problem called the Vehicle Routing Problem (VRP). It is a routing method that studies the optimal assignment of customers to a number of vehicles with specific maximum capacities on an optimized route (Gournaris et al., 2011). For school buses, however, adding constraints such as maximum travel times of each bus would transform this method into the School Bus Routing Problem (SBRP), a branch of the VRP that studies the distribution of students, bus routes, bus stops, and use of transportation policies (Ellegood et al., 2020; Lewis et al., 2018). For the optimized route computed from this project to be implemented at School A, bussing policies have to be considered; thus, the SBRP method is more suitable for the circumstances of this project. For the purpose of anonymity, “School A” is used instead of the school’s actual name.

**Problem statement**

There is a problem with the arrival time of school buses in the morning at School A, an international school in Tianjin, China. Buses are supposed to arrive at school before classes start, but buses are arriving late. This problem has negatively impacted students and teachers because students arriving in the middle of the class can distract others. A possible cause for this problem is that the travel times of the school buses are not optimized. If other international schools across China face a similar problem and have the same cause, a quantitative case study could answer the question of whether or not using Python can help optimize the school bus routes in terms of both time and cost.For further information,a quantitative case study analyzes the data collected for one subject to confirm or reject a hypothesis. In this study, the hypothesis is that if the optimized route proves to be time and cost efficient, then the route from this paper is considered to be optimized.

**Cornerstone work**

Like mentioned above, due to the amount of calculation needed, optimization problems are almost impossible to address without technologies’ help. Y.L. Li and L.F. Yan (2018) specifically study and simulate the school bus scheduling model at Yanshan University, using tools like Matlab (Li & Yan, 2018). Although this paper does not directly implement the SBRP, Li & Yan (2018) analyze Yanshan University’s current scheduling model and improve it using computer software. This is especially similar to the structure of this paper since this paper also collects data on the current situation and uses a tool, Python, to help optimize the situation. Other than this, another study by Bertsimas et al. (2019) provides evidence that optimization problems in urban settings can be addressed to provide benefits to schools, such as saving money and changing schools’ start times (Bertsimas et al., 2019). Bertsimas et al. (2019) then implement the SBRP to not only optimize the bus routes, but to also save money and even reform the school starting time (Bertsimas et al., 2019). This study is also crucial to the field of optimization problems since the VRP often gets more complicated in an urban setting due to uncertainties such as car accidents, weather conditions, and traffic jams. These uncertainties are hard to predict and may influence the effectiveness of routing optimization projects. This study then tries to consider the effects of these uncertainties during the data collection process.

**Argument and gap**

In the scholarly community studying route optimization techniques, there are not a lot of arguments on which method is absolutely correct; however, there are differences in the method/algorithm selection process depending on the constraints and goals of each project. In addition, since School A is an international school, the author acknowledges a gap in the field that studies the SBRP at international schools.

Research has been done on international schools’ bus management style and their service satisfaction level, but there has not been a paper that specifically focuses on the optimization of an international school’s bus route (Liu, 2016; Zhao, 2011). In addition, regarding the topic of SBRP, it is especially important to focus on School A because of the geographic location of School A. According to the United Nations Department of Economic and Social Affairs (UN DESA, 2018), there are six megacities (cities with a population of more than 10 million) in China (megacity, n.d.; UN DESA, 2018, p. 77). Since School A is located in one of those six megacities, this paper may have a significant impact on how routes are planned at international schools across China (UN DESA, 2018, p. 77). The reason that this paper emphasizes “international school” is that local Chinese schools are divided into school districts (Chen J. & Chen J.H., 2020). Since students within the districts are physically closer to their schools, they do not have as much need to take the bus as some students at international schools do. For instance, at School A, students are spread out in multiple districts in Tianjin (see Appendix A for coordinates and names of all the stops).

Furthermore, School A’s bus route planning system is not as “scientific” as it could be. Before conducting the research, the author of this paper interviewed the bussing coordinator at School A to collect more information on how bus routes at School A are planned. The researcher was informed that, each summer, School A’s bussing coordinator plans the next school year’s bus route using his/her personal experience at the job. However, purely using experience is not as effective as using a computer program due to the large number of possible routes. For these reasons, this paper aims to find an optimized school bus route for School A that may be generalized to other international schools across China.

**Literature Review**

This paper focuses on the routing problem, and there are several methods that could be used to address this. These methods, such as the Dijkstra algorithm, are used under different circumstances to serve different purposes depending on the research. Researchers often select the algorithm that best fits their research objective and constraints. Because almost every researcher has a different objective, it is uncommon to find debates on other researchers’ selection of a certain algorithm. Therefore, this section of the paper mainly focuses on how different circumstances and objectives may affect the algorithm/method selection. Five most popular and important methods are analyzed in the following order: the flooding algorithm, Dijkstra algorithm, robust optimization method, Vehicle Routing Problem (VRP), and School Bus Routing Problem (SBRP).

**The Flooding algorithm**

The Flooding Algorithm is an algorithm that runs through all possible routes and selects the shortest one (Xue et al., 2016). Although this may generate duplicated routes, this algorithm computes the shortest path after comparing it with all other paths (Xue et al., 2016). Weilian Xue et al. (2016) have optimized one of the school bus routes at Dalian Jiahui Sunshine Elementary School using the flooding algorithm. Considering roads’ lengths, characteristics, traffic congestion level, and bussing policies, Xue et al. (2016) prove the efficiency of the flooding algorithm (Xue et al., 2016). This paper is important to mention because Xue et al.’s work considers factors that are similar to those that this paper covers. This is possibly because both studies are done on schools. Since both School A and the elementary school in Xue et al.’s paper have specific bussing policies, consideration of such policies may also improve the effectiveness of the optimized route.

**The Dijkstra algorithm**

Although the flooding algorithm is effective, other methods should be evaluated in order to have a more thorough understanding on the existing methods. The Dijkstra algorithm, according to K. Mehlhorn and P. Sanders (2008), is used in finding the shortest route from a starting point to a target point on a weighted graph (Mehlhorn & Sanders, 2008). Using C# programming language for the Dijkstra algorithm, Zhou et al.’s research successfully optimizes a Chinese school’s bus route (Zhou et al., 2018). The Dijkstra algorithm is also useful for this project since a weighted graph should be used to consider the effects of traffic congestion level on busses’ travel times. For further information, a weighted graph is a graph that assigns a certain “weight” to each node on the graph (Fletcher et al., 1991). In this project, the weight would be the travel time needed from one stop to the next.

**Robust optimization method**

Another approach that is considered to be useful is the robust optimization method, which is a relatively recent method in comparison with other more traditional methods like the Vehicle Routing Problem and Dijkstra algorithm (Garcia et al., 2016). The robust optimization method provides solutions with consideration of worst-case scenarios, which includes traffic jams, bad weather, and traffic accidents (Garcia et al., 2016). By considering such events, this method, according to Sun et al., can be used “to generate a most cost-reliable route for school buses” (Sun et al., 2018). A drawback to using this method is that it is too “conservative” due to the evaluations of such rare events (Garcia et al., 2016). This indicates that it may lead to the school buses arriving unnecessarily early if none of the uncertainties happen. At the same time, because it is a relatively new method, it has not been implemented as much as the more traditional methods have.

**The Vehicle Routing Problem and School Bus Routing Problem**

Lastly, since the School Bus Routing Problem (SBRP) is a branch of the Vehicle Routing Problem (VRP), both of them are examined in this section (Lewis et al., 2018). Like mentioned before, the VRP includes variables such as travel time and vehicle capacity (Gounaris et al., 2011). However, unlike the VRP, the SBRP studies specifically how bus stops are allocated on each route with consideration of a maximum riding time (Lewis et al., 2018, p. 296). For example, one of the bus stops in this study has more than 100 students, which requires multiple buses to travel through that same stop to pick up students (Figure 1). On top of that, some schools may limit the maximum bus riding time of the students. While the VRP does not consider such constraints, the SBRP does.

**Summary**

After this analysis on the few possible methods used to address the routing problem, some key variables are emphasized. For example, the flooding algorithm and robust optimization method mainly consider situations on the road (congestion level, road lengths, weather, etc.). The Dijkstra algorithm finds the shortest path between one vertex and all other vertices. Both the VRP and SBRP look at the distribution of buses to different bus stops with consideration of the maximum capacity for each bus type, but the SBRP also focuses on additional variables such as maximum riding time. With this being said, the next section of the paper provides a more in-depth analysis and explanation on the specific method and procedures used to conduct this research.

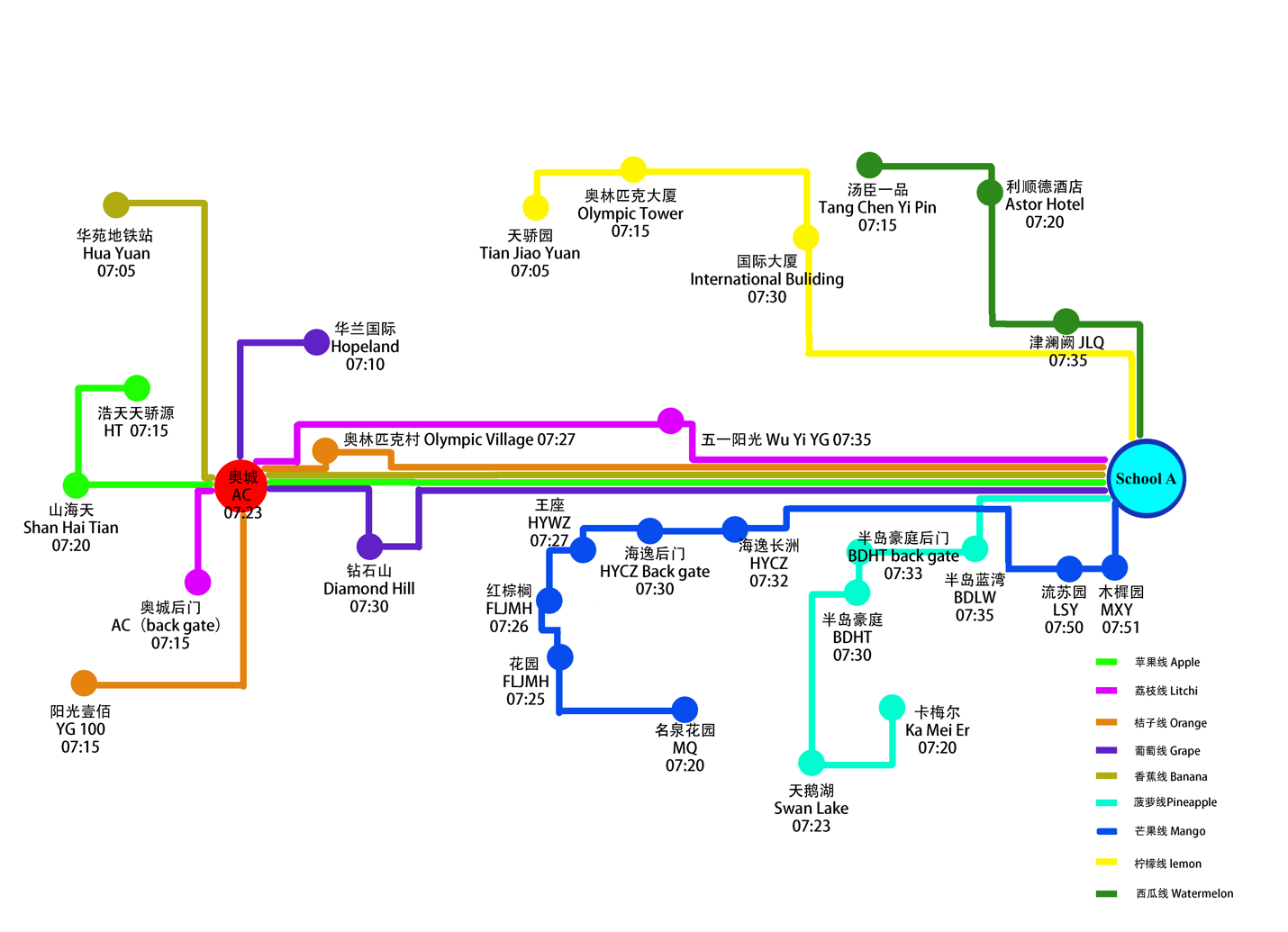
**Method**

**Method selection and analysis**

In order to select the most appropriate method/algorithm for this project, the current circumstances should be described. According to Figure 1, at the beginning of the 2020-2021 school year, there are nine buses in the morning running through multiple bus stops (thirty stops). Since the time spent traveling from one stop to the next stop varies based on distance and traffic flow, a weighted map should be used to model the problem and effectively minimize the travel time. In addition, the current bus route might not be optimized. Thus, durations from one stop to all other stops are recorded to ensure the optimized route is as close to being truly optimized as possible. In summary, this project considers several variables: school busses’ maximum capacity, duration from stop to stop, and maximum riding time allowed for each bus. Since these variables match the variables in SBRP, SBRP is the most appropriate method for this research paper.

**Figure 1**

*School A’s school bus routes from August, 2020 to June, 2021*



*Note*. The researcher received this figure from School A’s bussing department after the researcher physically interviewed the bussing coordinator while collecting information.

**Procedure**

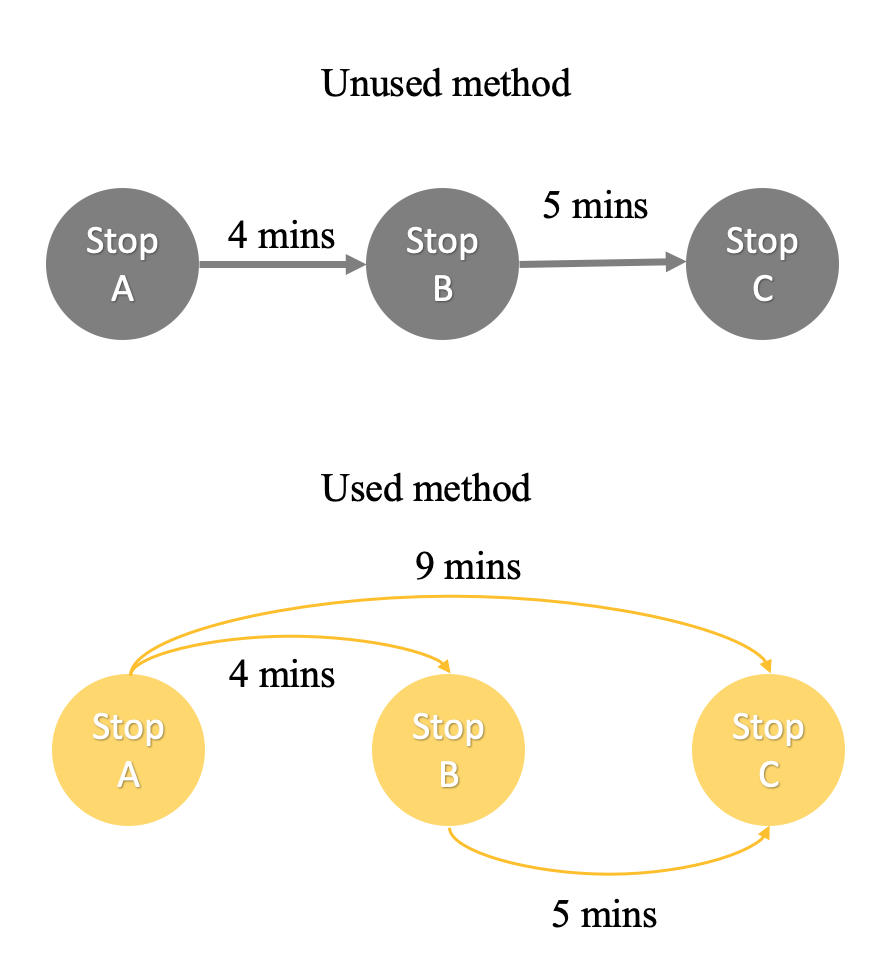
***Data Collection (First Part of Coding)***

Travel times from each stop to all other stops are collected to accurately construct a weighted map. A Chinese mapping software, Amap (Gaode), was used due to its current usage at School A. According to the bus coordinator at School A, each bus has a monitor, who checks Amap each morning to decide on which route to take to best avoid traffic jams. Amap not only has the ability to find the time needed for a vehicle to travel from one stop to the other, but it can also predict the time needed to travel at any time of the day for the entire week. At first, this tool was thought as a helpful tool for this project due to its ability to predict travel time with consideration of the traffic situation at a specific time. However, after physically testing this tool by taking car rides during rush hours, the researcher has noticed prediction errors by the tool. Therefore, data is collected in real-time and the process is described below.

With the researcher’s previous experience in Python, Python (Version 3.8.5) is used to help collect the travel time data. Since it is also unsure whether or not the current bus route is planned efficiently, the researcher has decided to collect the travel times from one stop to all other stops (Figure 2). This way, instead of finding the best combination of routes based on the current data from one stop to the next, the data collected from one stop to all other stops allows the computer to calculate the shortest total travel time for each bus on an optimized route.

**Figure 2**

*Explanation of the Used Method for Data Collection*

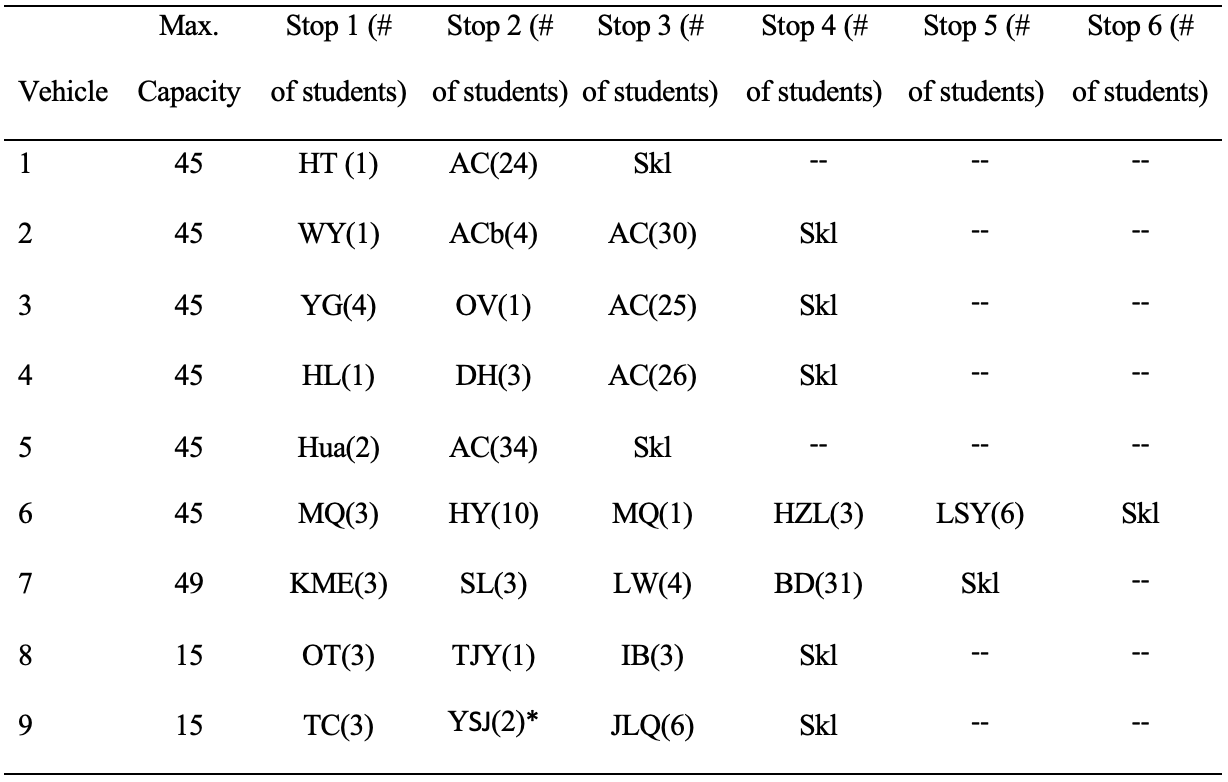


Specifically, the code that the author used to collect this data is called the “durations collector”, and it automatically collects the “navigation information” for all bus stops. Coordinates for each bus stop are first collected using Version 1.0 of the “Coordinates Picker” (2021), a tool on Amap’s Application Programming Interface (API) open platform. According to Meriam Webster, API is a program that allows programmers to develop software for a specific operating system (Application Programming Interface, n.d.). The code in Appendix B is then used to input the coordinates of two bus stops and get an estimated duration between these two stops using Amap’s current navigation data (see complete code in Appendix B; see sample results in Appendix C). This process is then repeated for all the stops, and a total of 465 durations are collected after each trial. Each trial is run twice every weekday in the morning at 7:24 and 7:47. The reason that these two times are chosen is that the average pick-up time for all the stops is 7:24 (Figure 1). In addition, according to School A’s Bussing Policy, busses are supposed to arrive before 8:10 (see Appendix D for bussing policy). Therefore, 7:47 is the average mid-point time. In addition, since the “durations collector” uses an online mapping software to help estimate the durations, such estimations may change every minute as traffic flow changes. Getting another estimation of travel times at 7:47 not only creates a larger sample size to minimize variations, but it also takes into account for the actual traffic situation while the busses are traveling. Because relatively low variations in the data collected through the “durations collector” are predicted, the collector is only run for three school weeks to collect enough data for this project. After the data is collected, it is put into an Excel document to visually show the variations from day 1 to day 15 (see Appendix E for the complete data).

***Constraints: Maximum Capacity***

Since this paper focuses on the School Bus Routing Problem (SBRP), maximum capacity of each bus should be considered (Lewis et al., 2018). Table 1 shows each bus’s maximum capacity and the number of students at each stop. This information has to be considered because buses must stay within their capacity limits.

**Table 1**

*Current Bus Route with Number of Students at Each Stop and Maximum Capacities of Each Vehicle* 

*Note*. YSJ is a stop that was added mid-year and was not informed to the researcher prior to the change. This stop was then not included in the results because there was not enough time to recollect the travel time data for that stop.

***Computing the result (second part of coding)***

The second part of coding is where the optimized route is computed. In order to do so, the SBRP is used[[1]](#footnote-1). Within the code, duration data is imported into Python (see Appendix F for the complete code). Duration data is taken from average travel times of all the trials for 15 days during the data collection process mentioned before. These average travel times are then put into a data table in an Excel document (Appendix E). While coding, two constraints are also included: maximum capacities and durations (Appendix F). In other words, the SBRP code essentially tries all possible combinations of routes and selects the solution that best meets the capacity limits and has the shortest total duration. Because of the large number of possible solutions, the code could run indefinitely. Therefore, a five-second running time limit is set because that is the shortest time it takes to compute the “optimized” result. Executing the code for four seconds or less computes a time that is slightly longer than that from running it for five seconds (Appendix G) but running the code for more than five seconds gives the same result as the one that is run for five seconds (Appendix H) Furthermore, the computed route from the code is displayed in the Results section.

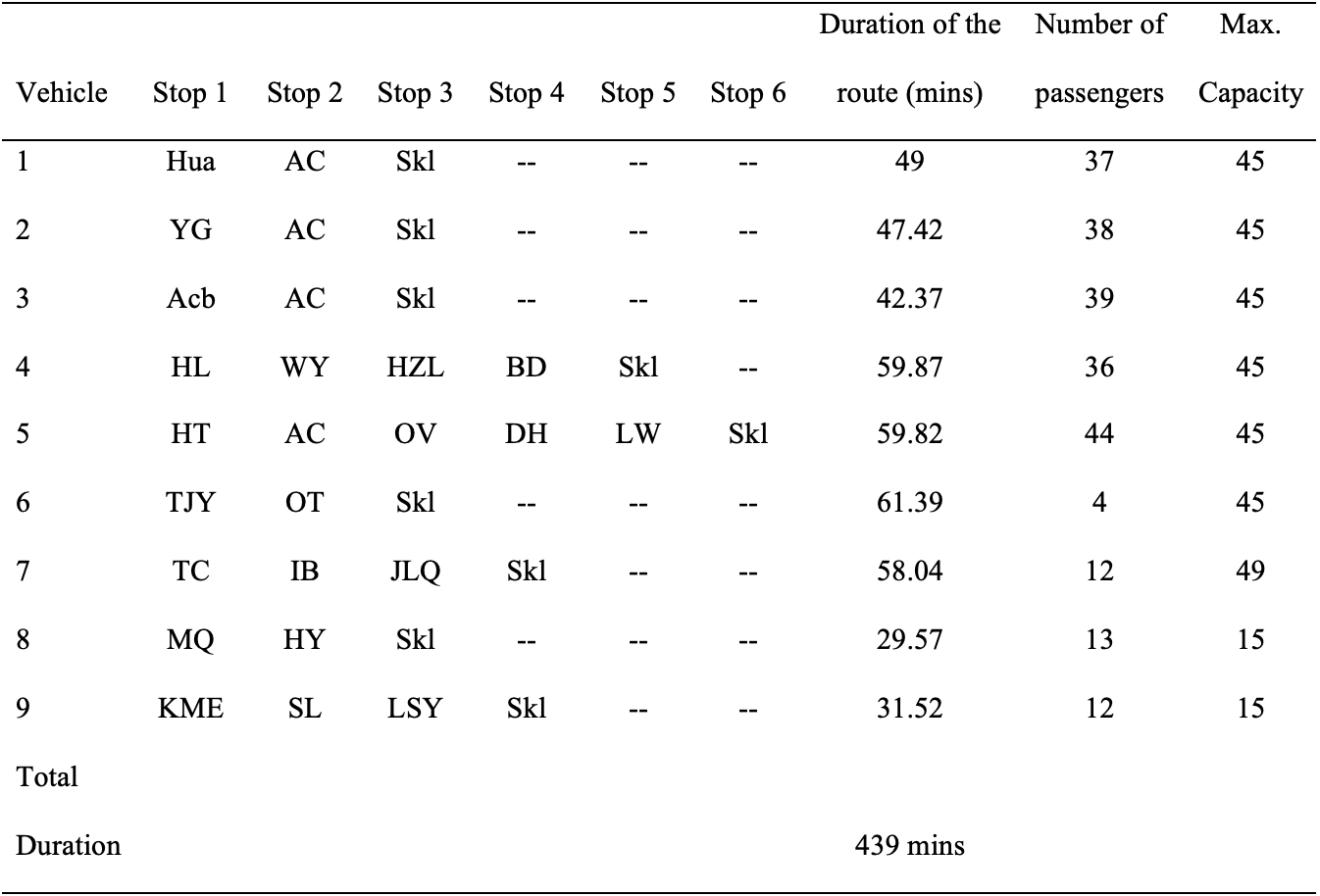
**Results**

**Findings**

Table 2 displays the output from the SBRP code. To help visualize the results, Figure 3 displays the optimized route on a map. It is found that the average travel time for all buses on the optimized route is around 439/9 ≈ 48.78 minutes.

**Table 2**

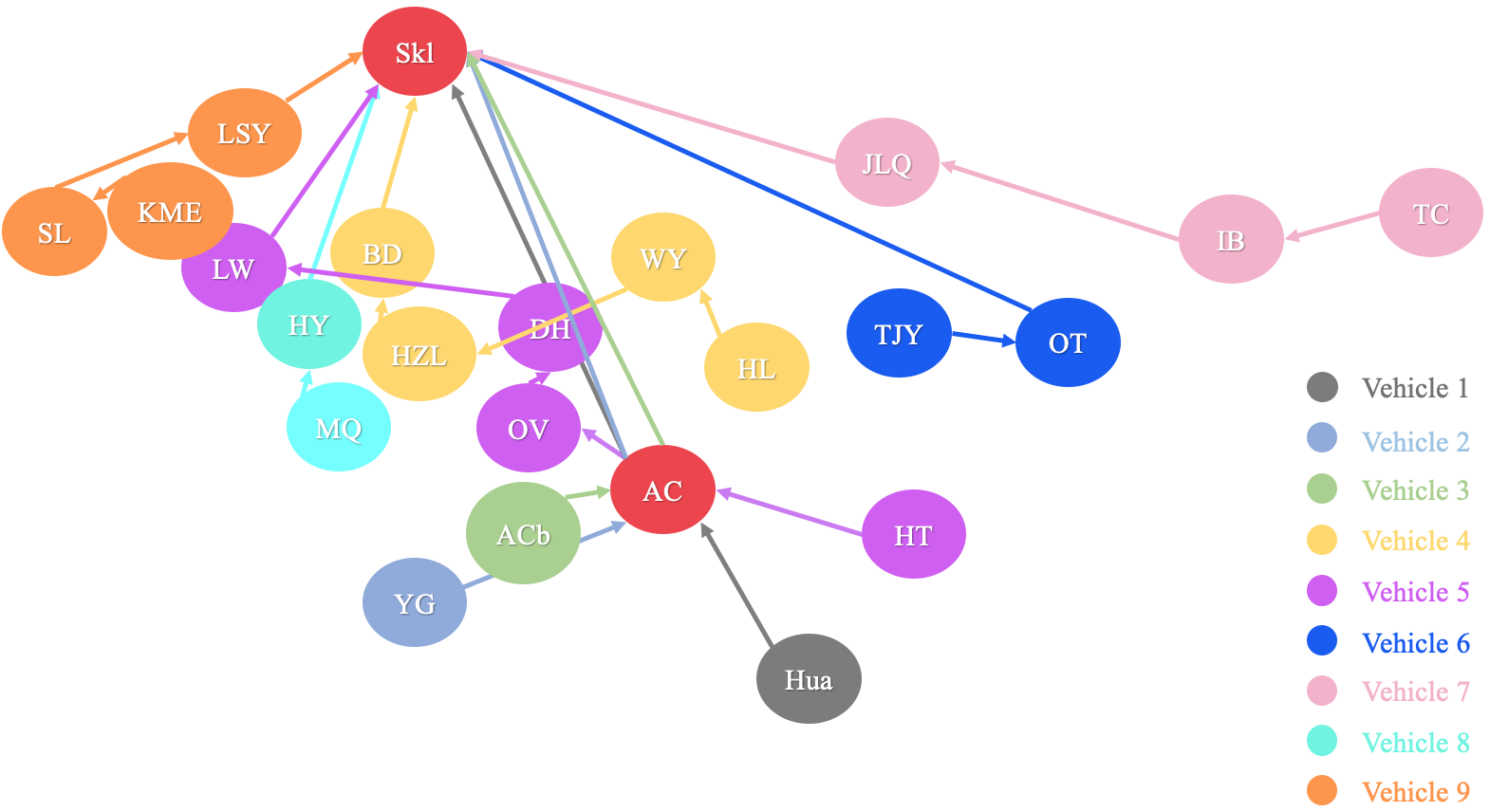
*Information of the “Optimized” Route from the SBRP Code*

****

*Note*. Blank columns mean that those vehicles visit less than 6 bus stops.

**Figure 3**

*Map of the Result From the SBRP Code*



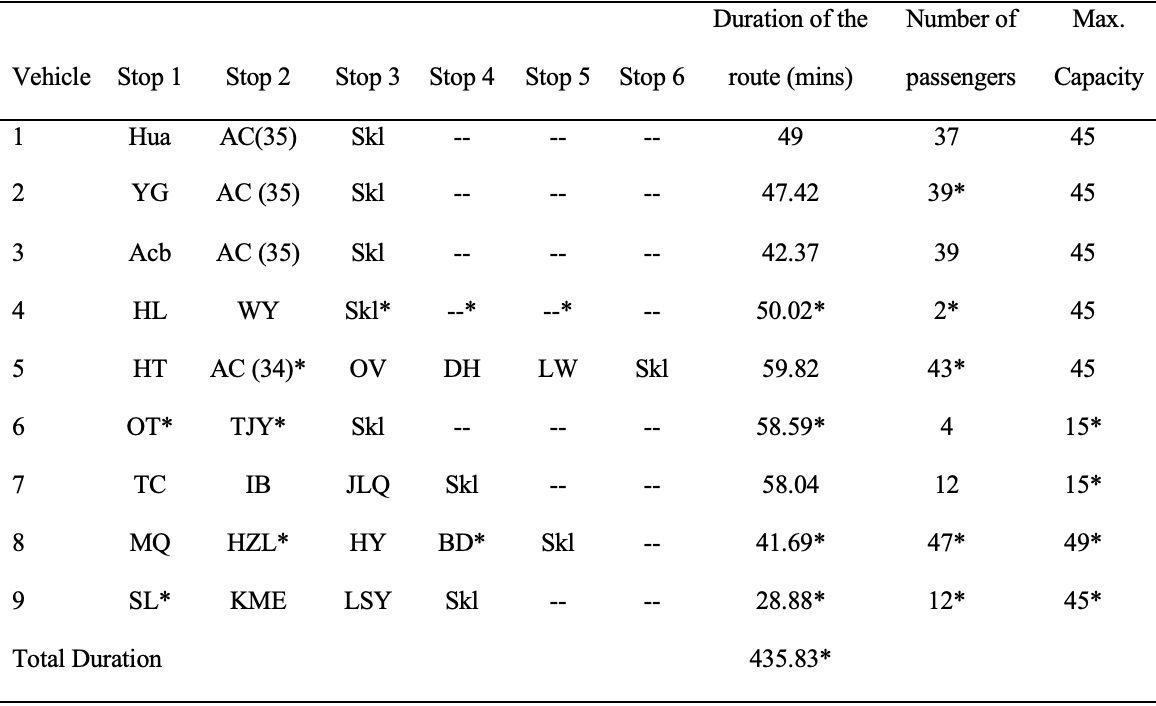
The result from Table 2 is not the finalized route information since the VRP is considered to be an NP-hard problem (Toth & Vigo, 2002). Therefore, the results computed are only approximations of what an actually optimized route might look like because, in NP-hard problems, the time required to solve a certain problem increases exponentially with the number of possible solutions, n! (Leeuwen, 1998). In this case, *n* is the number of bus stops, and the required time to run the code and compute a truly optimized result increases as *n* increases. In other words, the approximated optimized route computed from the SBRP code should be manually adjusted to provide a relatively more optimized route. Through manually adjusting the route, two different routes are generated based on average travel time and cost. Route 1 is adjusted based on the result in Figure 3, and Route 2 is adjusted based on Route 1.

**Route 1 (Optimized Average Travel Time)**

Since all starting stops in the code are set as the original starting stops in Figure 1, a few stops in Figure 3 can be adjusted to make Route 1 more optimized (Figure 1; Figure 3). Namely, TJY, OT, and KME’s routes seem odd. The path for Vehicle 6, which starts at TJY then goes to OT and Skl (School A), takes 61.39 minutes (Figure 3; Table 2). However, using the data table (Table E.1), if OT is set as the starting stop, it takes 10.5 + 48.09 = 58.59 minutes to complete the route, which is almost 3 minutes less than the route computed from the SBRP code (Figure 4; Table 3; Appendix E).

**Table 3**

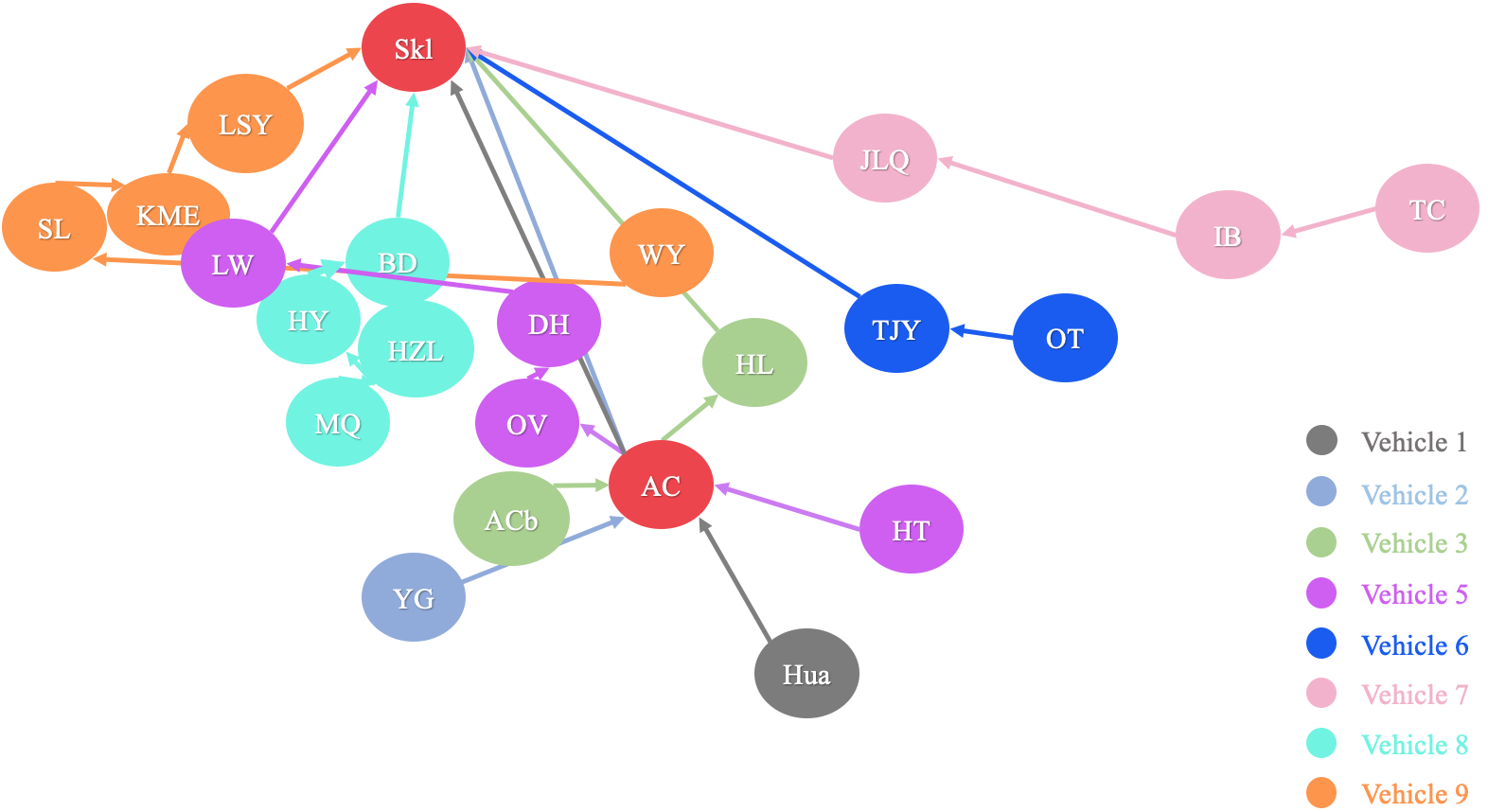
*Route 1 (Information of the Manually Adjusted Bus Route with 9 Buses in Total)*



*Note*. All changes from Table 2 are marked with asterisks.

**Figure 4**

*Map of the Manually Adjusted Optimized Bus Route*

**

*Note*. Vehicle 4’s bus route has been removed as shown in Table 4.

In addition, although the SBRP code considers vehicle capacity as a constraint while it computes the optimized route, it does not consider the physical sizes of the vehicles. For example, vehicles with a maximum capacity of 15 are smaller than the other vehicles (Table 2). A smaller vehicle size may reduce the travel time because it travels more freely in traffic than a bigger bus may; therefore, the researcher decides to replace Vehicle 6 and 7 with vehicles that have a maximum capacity of 15. This is because the stops that these two vehicles have to travel through­­—TJY, OT, TC, IB, and JLQ—are all relatively further away from School A (Table 2; Appendix E). Although this change does not theoretically change the travel times, it still may result in a reduction in actual travel times.

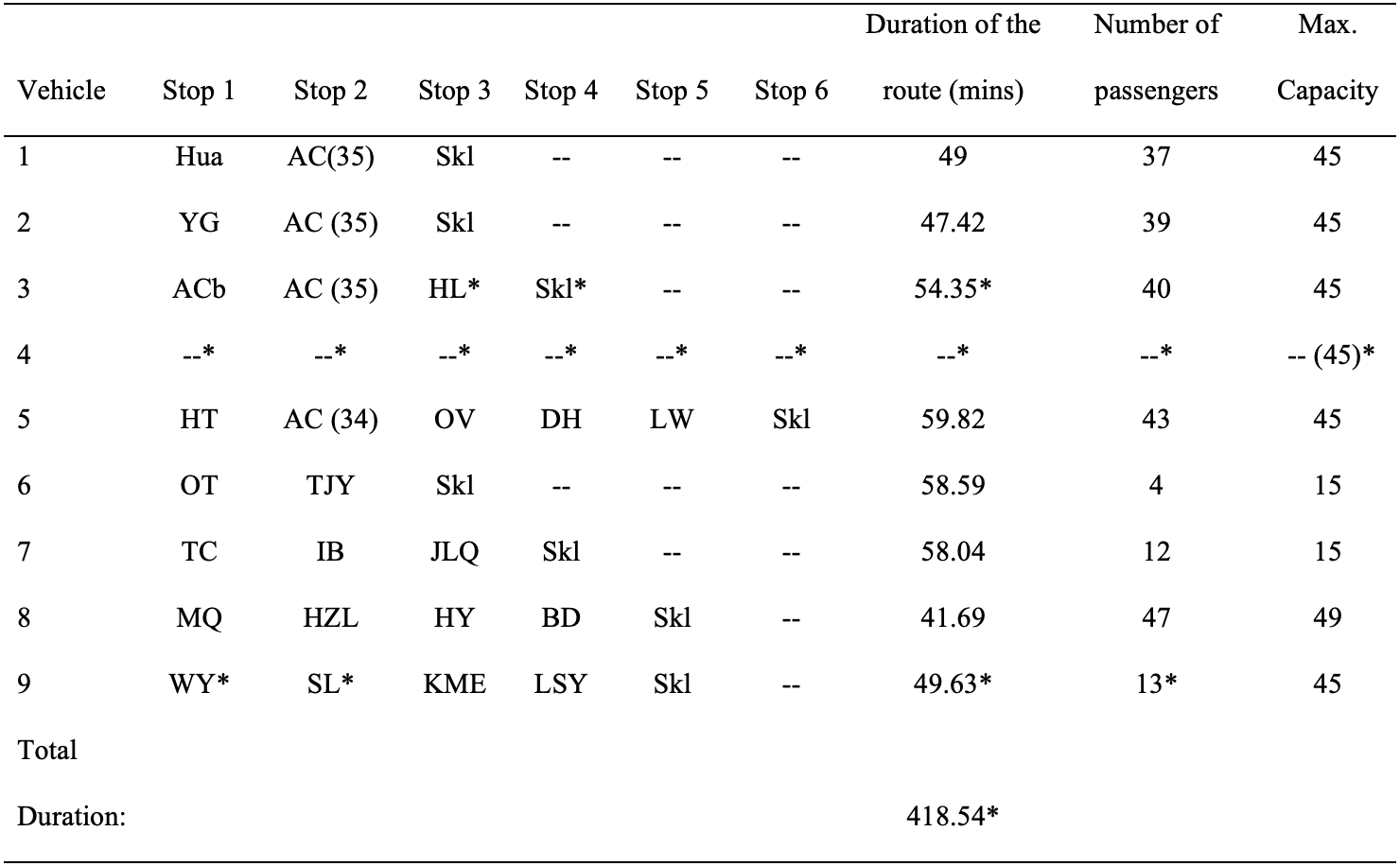
Furthermore, the starting stop for Vehicle 9 is changed from KME to SL (Table 3). After noticing that HZL and BD, which originally are stops for Vehicle 4 (Figure 3), are closer to the HY and MQ cluster for Vehicle 8 (Figure 3), HZL and BD are allocated to Vehicle 8 (Figure 4). In summary, with 9 busses, Route 1 has a total travel time of 435.83 minutes and an average travel time of 435.83/9 ≈ 48.43 minutes (Table 3).

**Route 2 (Optimized Cost)**

From Route 1, after allocating HZL and BD to Vehicle 8, the researcher realize that Vehicle 4 now has 2 passengers in total (Table 3). Because it would not be a great use of the resources to only pick up two students, the researcher decides to further allocate stops for Vehicle 4 to other vehicles. The researcher sets HL as the third stop for Vehicle 3 and WY as the first stop for Vehicle 9 (Table 4). This change then eliminates the need of Vehicle 4 and decreases the total duration by 17.29 minutes (from 439 minutes in Table 2 to 418.54 minutes in Table 4). However, this increases the average duration, from Route 1, 435.83/9 ≈ 48.43 minutes to 418.54/8 ≈ 52.32 minutes (Table 3; Table 4).

**Table 4**

*Information of the Manually Adjusted Bus Route with 8 Buses in Total (After the Removal Vehicle 4)*

**

*Note*. All changes from Table 3 are marked with asterisks. Cells for Vehicle 4 are blank because stops on this route are allocated to other vehicles. Other cells that remain empty mean that those vehicles have less than 6 bus stops.

**Summary**

With everything said, Route 1 is more optimized in terms of durations since the average travel time is only 48.43 minutes, but this may not seem to be the most cost-efficient solution. In comparison, the average travel time for Route 2 in Figure 4 is 3.89 minutes longer, but it does require one less vehicle than the first solution. Reducing one vehicle would allow School A to spend less on bussing each year. Implications for the results are discussed in the next section.

**Discussion**

**Implications**

Before discussing the importance of these results, it should be made clear that the results from this paper cannot be directly compared with the current bus route at School A (Figure 1). This is because, after collecting the data, the bussing coordinator at School A has made changes to the bus route. Some students decided to transfer to other schools, some transferred to School A mid-year, and some decided not to take the buses anymore

***Reduced Cost***

At school A, the bussing coordinator elaborated that School A’s bussing cost depends on multiple variables. For example, School A is in charge of paying for all buses, drivers, and seats. Although there are other variables, such as insurance expenses, the most important variable is the number of vehicles. Thus, decreasing the number of vehicles would reduce the number of bus drivers and seats, which then lowers the bussing cost. Therefore, to optimize the bussing cost and travel times to some extent, the second route with one less vehicle benefits School A the most.

***Assistance in Future Planning***

After completing the project, the author realizes that not only does this project optimize bus routes based on travel times and costs, but it may also assist the bussing coordinator with his/her job. At School A, the author was told by the bussing coordinator he/she plans the bus route every summer using his/her personal experience. The method/algorithm used in this project then has another purpose: help plan future bus routes. For example, when new students arrive at School A and need new bus stops to pick them up at the right locations every morning, the computer program used in this paper can be easily used to accommodate such changes and provide the bussing coordinator with a close-to-optimized route in a short time.

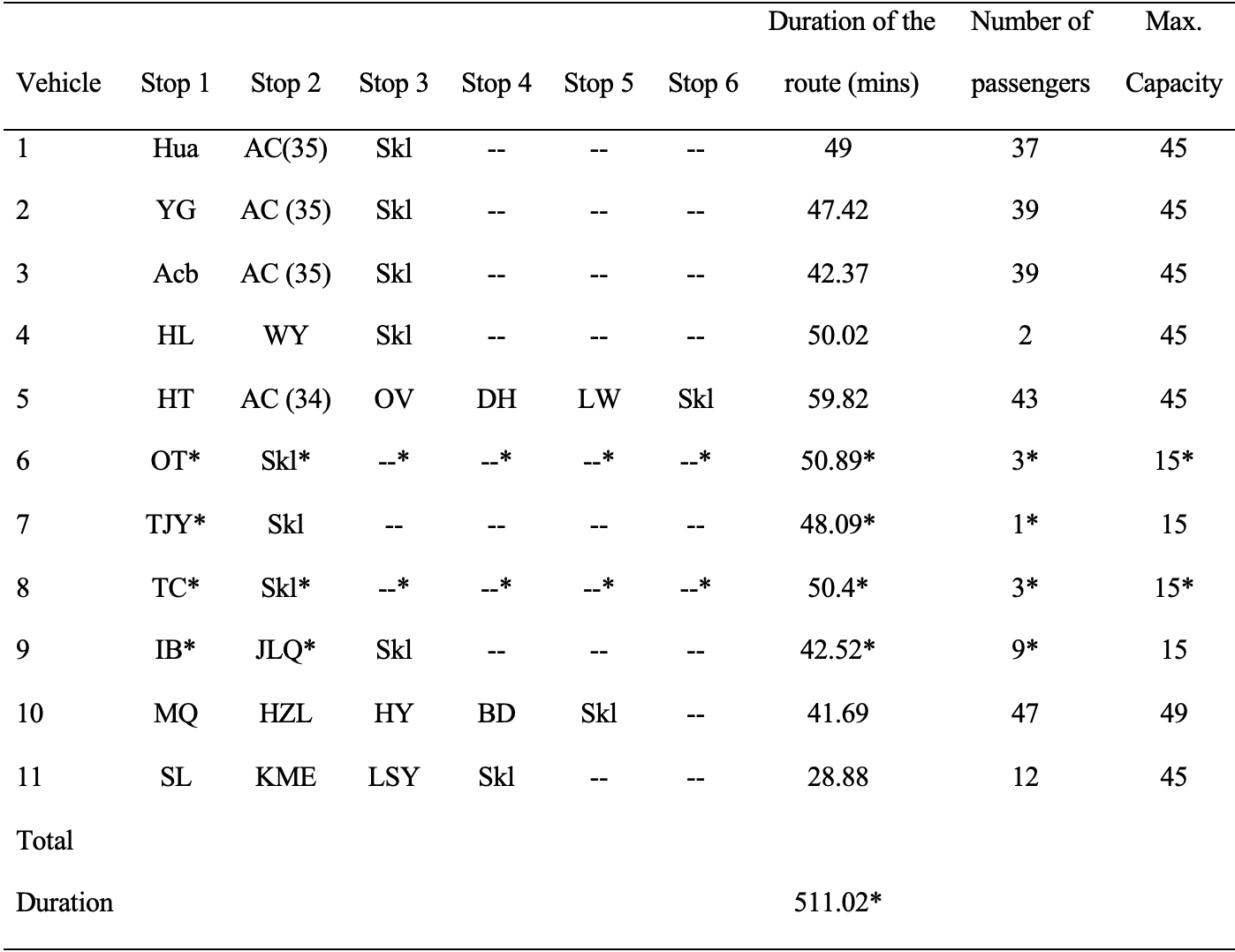
Other than its potential implications at School A, the method used in this paper may also be implemented at other international schools in China. The reason that the results from this paper cannot be generalized to all schools in China is that Chinese local schools do not need to use school buses because most students live in the same school district (Chen J. & Chen J.H., 2020). However, at international schools, instead of having most students live in the same district, students are relatively more spread out from the school. Due to such similarities within the international schools’ system, other international schools with similar problems and goals as School A may adopt the method used in this paper to help them achieve route optimization.

**Limitations**

The result computed in this paper does not consider the bussing policies at School A, Tianjin’s travel regulations, and parents’ opinions.

The SBRP code used in this paper includes a time contraint to ensure the result computed from the code is optimized based on travel times (Appendix F). However, the result still does not abide by all the bussing policties at School A. For instance, the first policy states that “the maximum time frame of each bus route should be no longer than 50 minutes on a normal day” (Appendix D). This policy, however, does not seem plausible in this project because traveling directly from OT or TC to Skl takes just above 50 minutes (Appendix E). It is essentially already impossible to abide by the policy even with two separate vehicles traveling to OT and TC (Table 5). Trying to follow the policy of having all travel times under 50 minutes would mean adding at least two vehicles, which also increases the bussing cost. This also means the route would not be considered as optimized in this paper since an optimized route is previously defined as being both cost and time efficient.

**Table 5**

*Route Information with OT and TC as Separate Stops*

*Note*. This requires using 11 vehicles instead of 9 vehicles.

Besides not being able to completely abide by the bussing policy, it is also challenging for this project to consider all travel regulations for the vehicles used. For example, according to the Regulation on the Implementation of the Road Traffic Safety Law of the People's Republic of China (2004), buses are also not allowed to make U-turns at every intersection. Making U-turns not only has a chance of creating a traffic jam, but there are also safety concerns. Therefore, if such regulations were considered, there could be an increase in actual travel times for some vehicles as the vehicle may have to find another way to arrive at the next stop. Some may say that, to prevent from making U-turns and still have a shorter travel time, students can simply cross the road to get on the bus; however, according to Regulations of Tianjin Municipality on the Administration of Passenger Transport and Public Transport (2015), students are required to board the bus at, and only at, their designated bus stops.

Furthermore, the bussing coordinator has described that multiple parents have concerns and opinions with the number of students on each bus. While I was getting some bussing information from the bussing coordinator, he explained that some parents wish that their kids could sit alone so that they could place their instruments, backpacks, and other personal items on the seats next to them. Although this may increase comfortability and satisfactory level of the parents and students, having one half of the bus empty would indicate needing more buses to travel and pick up all the passengers. This would significantly increase the bussing cost. Therefore, in order for School A to actually implement the method used in this study, further communication is needed to ensure that most parents are fine with increasing the number of students on each bus to optimize the routes and lower the bussing cost.

Last, but not least, the result from this paper cannot be directly tested to find out whether or not the result is optimized. School A’s bussing coordinator has told me that completely changing the bus route in the middle of the year is not allowed. Therefore, the computed optimized route is only theoretically optimized. It requires actual implementation and testing to justify whether it is truly optimized for School A.

**Future direction**

Next, future research can also be done on this topic to explore its potential. For example, other international schools in China can actually implement this algorithm to test the effectiveness of this method. There is a probability that the type of school and its location may not have any relationship with the effectiveness of the algorithm used in this paper. In addition, more algorithms could be tested or even created to best solve this sort of vehicle routing problem. Doing so may create an opportunity to consider safety regulations and bussing policies, which are not completely achieved in this project. However, this could lead to a decrease in the algorithm’s ability to be implemented at other schools or organizations since safety regulations and bussing policies for specific groups may differ depending on the characteristics of their customers, such as age group and geographical location. For example, China’s travel regulation may vary significantly from that in the U.S. Furthermore, customers aged 7-12 may also require additional specific safety regulation that is not required if the customers are aged 16-18. These variables can be evaluated in the future to create functional algorithms for all customers.

**Conclusion**

The study aims to optimize the school bus route for School A so that travel times of the buses are as short as possible. After spending much time choosing the suitable algorithm, collecting data, coding, and adjusting and analyzing the result, the School Bus Routing Problem (SBRP) is used to compute two optimized routes for School A. The total travel times for 24 bus stops are approximately between 415 and 435 minutes with an average travel time of around 47 to 50 minutes for each bus. By increasing the average travel time from 47 minutes for 50 minutes, one bus could be removed, which reduces the cost of bus route planning. Although limitations, such as inconsideration of uncertainties, China’s travel regulations, and parents’ opinions, may affect the feasibility of the results from this paper, this study still hold important implications to School A and other international schools in China. Some other variables can also be considered to improve the effectiveness of this result, but this paper has examined thoroughly on ways to achieve route optimization based on travel times and cost.

**References**

Application programming interface. (n.d.) In *Merriam-Webster.com dictionary*. Retrieved March 10, 2021, from <https://www.merriam-webster.com/dictionary/application%20programming%20interface>

Bertsimas, D., Delarue, A., & Martin, S. (2019). Optimizing schools’ start time and bus routes. *Proceedings of the National Academy of Sciences of the United States of America*, 114 (13), 5947-5948. <https://doi.org/10.1073/pnas.1811462116>

Chen, J. & Chen, J.H. (2020). 从“管理”到“治理”:学区政策的演进逻辑和发展趋势 [From “management” to “governance”: a study of policy changes and logic in school districts]. *Nanjing Journal of Social Sciences*, *21*(8), 133-140. <https://doi.org/10.15937/j.cnki.issn1001-8263.2020.09.018>

Coordinate Picker [Computer software]. (2021). Retrieved from <https://developer.amap.com/tools/picker>

, W.A., Solomon, S., North, J., & Campbell, J.F. (2020). School bus routing problem: Contemporary trends and research directions. *Omega*, *95*. Retrieved December 04, 2020, from <https://www.sciencedirect.com/science/article/abs/pii/S0305048318305127?via%3Dihub>

Fletcher, P., Hoyle, H., & Patty, C. W. (1991). *Foundations of discrete mathematics*. PWS-KENT Pub. Co.

Garcia, D. J., Gong, J., & You, F. (2016). Multi-stage adaptive robust optimization over bioconversion product and process networks with uncertain feedstock price and biofuel demand. *Elsevier*, *38*, 217-222. <https://doi.org/10.1016/B978-0-444-63428-3.50041-2>

Google OR-Tools [Computer software]. (2021). Retrieved from <https://developers.google.com/optimization/routing/cvrp>

Gounaris, C. E., Repoussis, P. P., Tarantilis, C. D., & Floudas, C. A. (2011). A hybrid branch-and-cut approach for the capacitated vehicle routing problem. *Elsevier*, *29*, 507-511. <https://doi.org/10.1016/B978-0-444-53711-9.50102-4>

Leeuwen, J. van. (1998). *Algorithms and complexity*. Elsevier.

Lewis, R., Smith-Miles, K., & Phillips, K. (2018) The School Bus Routing Problem: An analysis and algorithm. *IWOCA*, 10765, 287-298. <https://doi.org/10.1007/978-3-319-78825-8_24>

Li, Y.L., & Yan, L.F. (2018). 校车调度优化问题的研究——以燕山大学校车为例 [School bus route optimization problem–Case study of Yanshan University]*. Statistics and Management, 33*(4), 47-53. Retrieved 2020, from <http://lib.cqvip.com/Qikan/Article/Detail?id=7000756955&from=Qikan_Search_Index>

Liu, L. (2016). *国际学校校车服务满意度现状研究 [Research on satisfaction status of international school bus service—a case study on International School of Tianjin]* (Publication No. 1423030015) [Master’s thesis, Tianjin Normal University]. Retrieved April 21, 2021, from <https://cdmd.cnki.com.cn/Article/CDMD-10065-1017187934.htm>

Megacity. (n.d.) In *Cambridge dictionary*. Retrieved April 20, 2021, from <https://dictionary.cambridge.org/dictionary/english/megacity>

Mehlhorn, K., & Sanders, P. (2008). Shortest Paths. *Algorithms and data structures: the basic toolbox* (pp. 191–215). Springer.

Regulation on the Implementation of the Road Traffic Safety Law of the People's Republic of China, art. 4 (2004), Zhonghua Renmin Gongheguo Guowuyuan.

Regulations of Tianjin Municipality on the Administration of Passenger Transport and Public Transport, art. 4 (2015), Tianjin Shi Renda Chang Wei Hui.

*School A’s school bus routes from August, 2020 to June, 2021* [Current bus routes with departure times] (2020)

Shareia, B.F. (2016). Qualitative and quantitative case study research method on social science: Account perspective. *International Journal of Economics and Management Engineering, 10*(12), 3839-3844. Retrieved May 11, 2021, from <http://scholar.waset.org/1307-6892/10005863>

Sun, S., Duan, Z., & Xu, Q. (2018, January 01). School bus routing problem in the stochastic and time-dependent transportation network. *Public Library of Science*, *13*(8). Retrieved November 01, 2020, from <https://doaj.org/article/78d7428d8bae48728ab0fd1fbc2237be>

Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Society for Industrial and Applied Mathematics

United Nations, Department of Economic and Social Affairs, Population Division (2018). World urbanization prospects: the 2018 revision. *United Nations*.

Xue, W., Yu X., Zhou, F., & Li, Q. (2016). 基于洪泛算法的单线校车路径规划问题研究 [Study on single-track school bus route planning problem based on flooding algorithm]. *Logistics Technology, 35*(10), 48-52. Retrieved November 02, 2020, from <http://lib.cqvip.com/Qikan/Article/Detail?id=670578074&from=Qikan_Search_Index>

Zhao, Y. (2011). *国际学校的校车管理研究 [Study on school bus management in international school: a case study from the perspective of cross culture]* (Publication No. 62070116056) [Master’s thesis, East China Normal University]. Retrieved April 21, 2021, from <http://cdmd.cnki.com.cn/Article/CDMD-10269-1011130284.htm>

Zhou, F., Yang, Z., & Chi, A. (2018). 基于GIS和Dijkstra算法的校车路线方案优化 [School bus route optimization based on GIS and Dijkstra algorithms]. *Geospatial Information,* *16*(6), 60-63. Retrieved November 01, 2020, from <http://lib.cqvip.com/Qikan/Article/Detail?id=675793044>

**Appendix A**

**Coordinates of All Bus Stops and Names**



**Appendix B**

**Code Used to Get the Time From One Stop to All Other Stops**

**import** **requests**

**import** **datetime**

*#this code is used to find the durations from every bus stop to all other stops*

*#it is ran on every weekday morning at 7:24 and 7:47 to get average travel times with consideration of traffic*

*#constants for requesting a URL to get a duration of each segment*

KEY = '7487ef48b2a8734006f9c9106dca8299'

URL\_constant = 'https://restapi.amap.com/v3/direction/driving?'

*#opens the file with all the coordinates of the stops*

*#each line represents two bus stop's coordinates at origin and destination*

stops\_file = open('//Users/stellenshun.li/Desktop/12th Grade/AP Research/Paper/Codes/stops/stops\_copy.txt','r')

stops\_names = open('//Users/stellenshun.li/Desktop/12th Grade/AP Research/Paper/Codes/stops/stops\_test.txt','r')

*#variables and lists used to calculate and store all the durations*

orig = []

dest = []

durations = []

durations\_mins = []

secToMin = 60

all\_coordinates = []

stop = []

NAMES = []

names\_segments = []

*#a loop to get all the coordinates of each stop*

**for** line **in** stops\_file.readlines():

*#splits each line into an array in the format of [origin, destination]*

stop = line.strip()

all\_coordinates.append(stop)

*#builds the request URL for this segment; the URL gets the duration for each segment and puts the duration(sec) into the durations list*

**for** i **in** range(len(all\_coordinates)):

**for** j **in** range(i, len(all\_coordinates)):

orig = all\_coordinates[i]

dest = all\_coordinates[j]

URL = "https://restapi.amap.com/v3/direction/driving?key=7487ef48b2a8734006f9c9106dca8299&" + "origin=" + orig + "&destination=" + dest

URL2 = URL + "&originid=&destinationid=&extensions=base&strategy=10&waypoints=&avoidpolygons=&avoidroad="

resp = requests.get(URL2)

durations.append(resp.json()['route']['paths'][0]['duration'])

**for** k **in** range(0, len(durations)):

durations[k] = int(durations[k])

*#converts all durations from seconds to minutes rounded to the nearest 1 decimal place*

durations\_mins = [num/secToMin **for** num **in** durations]

**for** i **in** range(len(durations)):

durations\_mins[i] = float(round(durations\_mins[i], 1))

durations\_mins[i] = str(durations\_mins[i])

*#gets current time and date (month day year)*

time = datetime.datetime.now()

time = time.strftime('%a-%m-%d-%y')

*#loads the text file with all stops' names and puts all names into segments in a format of "(stop 1) to (stop 2): "*

**for** line **in** stops\_names.readlines():

name = line.strip()

NAMES.append(name)

**for** i **in** range(len(NAMES)):

**for** j **in** range(i, len(NAMES)):

names\_segments.append(NAMES [i] + " to " + NAMES [j] + ": ")

*#creates a text file with current date as the name of the file and all stops' names and durations of that day*

**with** open(r'//Users/stellenshun.li/Desktop/12th Grade/AP Research/Paper/Codes/Data/'+time + '.txt', 'w') **as** filehandle:

**for** i **in** range(len(names\_segments)):

filehandle.write(names\_segments[i] + durations\_mins[i] + '**\n**')

**Appendix C**

**Results of a Test Run From the Code in Appendix B**

Hua to Hua: 0.0

Hua to AC: 9.0

Hua to Skl: 25.9

Hua to YG: 14.4

Hua to ACb: 8.4

Hua to OV: 10.0

Hua to DH: 12.8

AC to AC: 0.0

AC to Skl: 25.8

AC to YG: 14.0

AC to ACb: 5.9

AC to OV: 6.9

AC to DH: 9.5

Skl to Skl: 0.0

Skl to YG: 32.7

Skl to ACb: 22.2

Skl to OV: 23.5

Skl to DH: 24.2

YG to YG: 0.0

YG to ACb: 8.2

YG to OV: 10.2

YG to DH: 13.4

ACb to ACb: 0.0

ACb to OV: 9.0

ACb to DH: 11.7

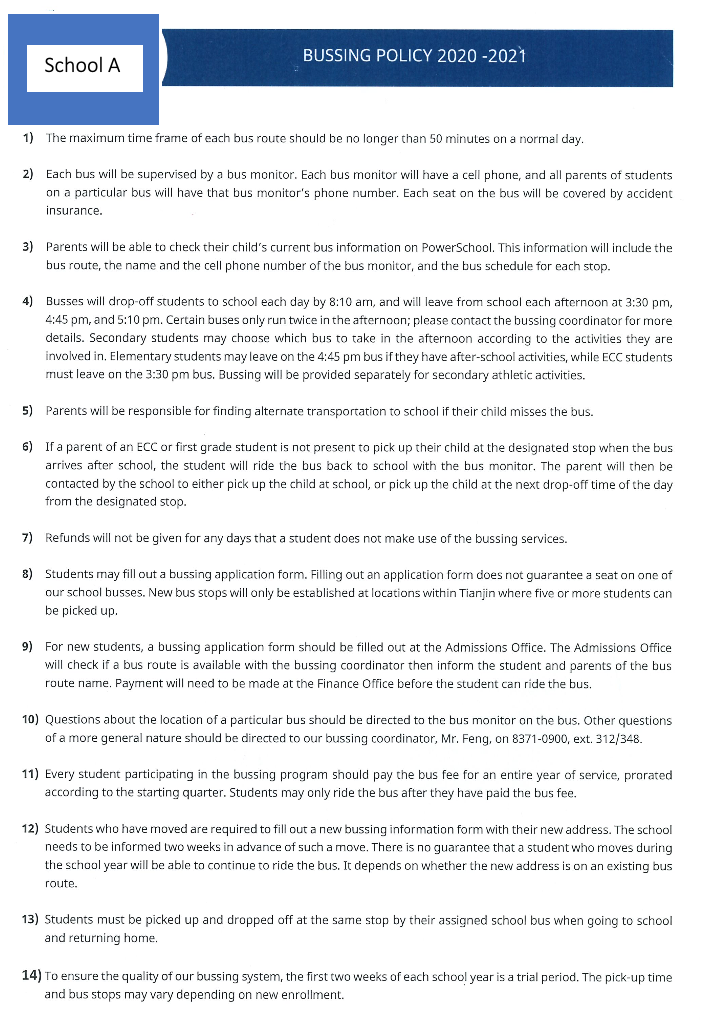
OV to OV: 0.0

OV to DH: 8.6

DH to DH: 0.0

**Appendix D**

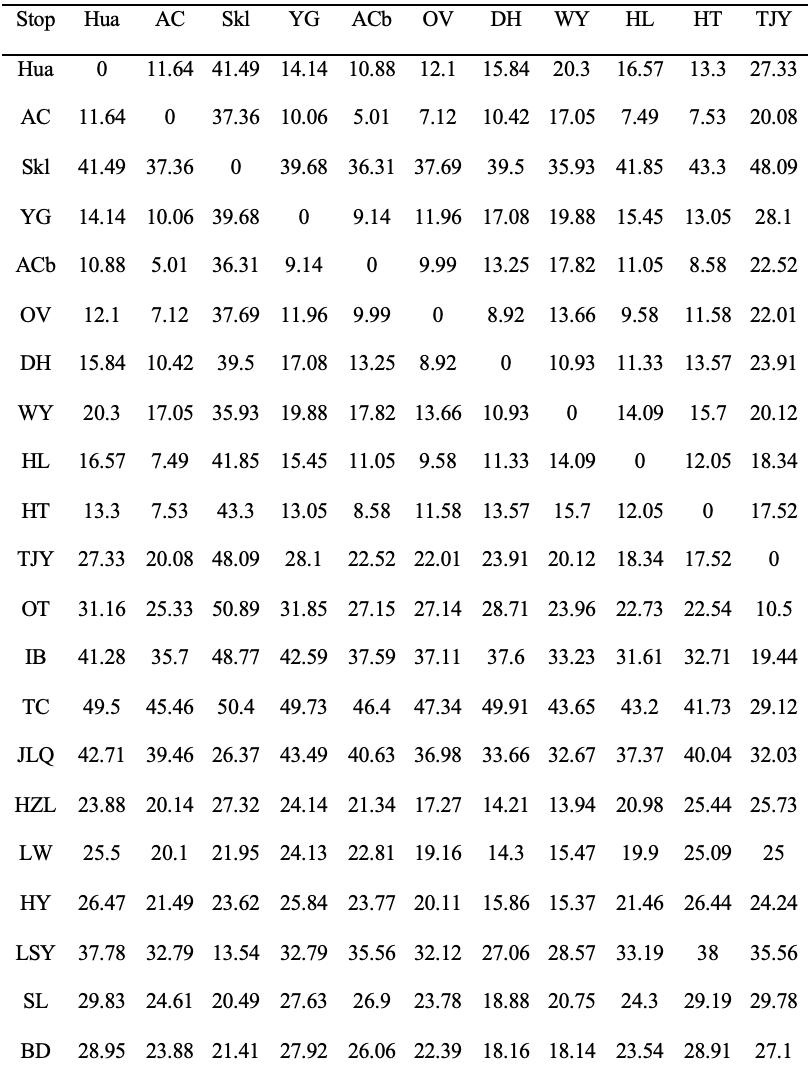
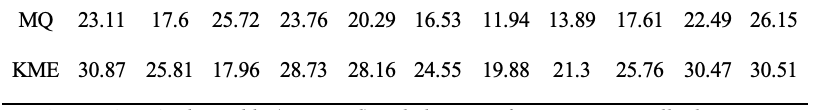
**Bussing Policy at School A**



**Appendix E**

**Table E.1**

*Data Table With Durations From One Stop to All Other Stops*

**

**Table E.2**

*Data Table (Continued) With Durations From One Stop to All Other Stops*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stop | OT | IB | TC | JLQ | HZL | LW | HY | LSY | SL |
| Hua | 31.16 | 41.28 | 49.5 | 42.71 | 23.88 | 25.5 | 26.47 | 37.78 | 29.83 |
| AC | 25.33 | 35.7 | 45.46 | 39.46 | 20.14 | 20.1 | 21.49 | 32.79 | 24.61 |
| Skl | 50.89 | 48.77 | 50.4 | 26.37 | 27.32 | 21.95 | 23.62 | 13.54 | 20.49 |
| YG | 31.85 | 42.59 | 49.73 | 43.49 | 24.14 | 24.13 | 25.84 | 32.79 | 27.63 |
| ACb | 27.15 | 37.59 | 46.4 | 40.63 | 21.34 | 22.81 | 23.77 | 35.56 | 26.9 |
| OV | 27.14 | 37.11 | 47.34 | 36.98 | 17.27 | 19.16 | 20.11 | 32.12 | 23.78 |
| DH | 28.71 | 37.6 | 49.91 | 33.66 | 14.21 | 14.3 | 15.86 | 27.06 | 18.88 |
| WY | 23.96 | 33.23 | 43.65 | 32.67 | 13.94 | 15.47 | 15.37 | 28.57 | 20.75 |
| HL | 22.73 | 31.61 | 43.2 | 37.37 | 20.98 | 19.9 | 21.46 | 33.19 | 24.3 |
| HT | 22.54 | 32.71 | 41.73 | 40.04 | 25.44 | 25.09 | 26.44 | 38 | 29.19 |
| TJY | 10.5 | 19.44 | 29.12 | 32.03 | 25.73 | 25 | 24.24 | 35.56 | 29.78 |
| OT | 0 | 14.21 | 20.86 | 28.42 | 33.44 | 35.82 | 35.66 | 43.71 | 37.92 |
| IB | 14.21 | 0 | 15.52 | 16.15 | 32.61 | 33.93 | 34.08 | 38.86 | 37.65 |
| TC | 20.86 | 15.52 | 2.04 | 15.85 | 35.53 | 35.15 | 35.37 | 40.16 | 39.09 |
| JLQ | 28.42 | 16.15 | 15.85 | 0 | 23.53 | 22.8 | 22.83 | 29.26 | 26.6 |
| HZL | 33.44 | 32.61 | 35.53 | 23.53 | 0 | 9.86 | 8.93 | 23.89 | 14.12 |
| LW | 35.82 | 33.93 | 35.15 | 22.8 | 9.86 | 0 | 4.83 | 16.6 | 9.11 |
| HY | 35.66 | 34.08 | 35.37 | 22.83 | 8.93 | 4.83 | 0.35 | 17.43 | 8.02 |
| LSY | 43.71 | 38.86 | 40.16 | 29.26 | 23.89 | 16.6 | 17.43 | 0 | 14.72 |
| SL | 37.92 | 37.65 | 39.09 | 26.6 | 14.12 | 9.11 | 8.02 | 14.72 | 0.05 |
| BD | 37.15 | 34.59 | 34.56 | 21.62 | 10.43 | 7.27 | 4.38 | 19.44 | 11.07 |
| MQ | 34.92 | 34.92 | 38.8 | 26.08 | 6.97 | 7.67 | 5.95 | 20.5 | 11.47 |
| KME | 37.44 | 36.25 | 37.24 | 24.27 | 14.17 | 7.45 | 8.22 | 12.08 | 3.26 |

**Table E.3**

*Data Table (Continued) With Durations From One Stop to All Other Stops*

|  |  |  |  |
| --- | --- | --- | --- |
| Stop | BD | MQ | KME |
| Hua | 28.95 | 23.11 | 30.87 |
| AC | 23.88 | 17.6 | 25.81 |
| Skl | 21.41 | 25.72 | 17.96 |
| YG | 27.92 | 23.76 | 28.73 |
| ACb | 26.06 | 20.29 | 28.16 |
| OV | 22.39 | 16.53 | 24.55 |
| DH | 18.16 | 11.94 | 19.88 |
| WY | 18.14 | 13.89 | 21.3 |
| HL | 23.54 | 17.61 | 25.76 |
| HT | 28.91 | 22.49 | 30.47 |
| TJY | 27.1 | 26.15 | 30.51 |
| OT | 37.15 | 34.92 | 37.44 |
| IB | 34.59 | 34.92 | 36.25 |
| TC | 34.56 | 38.8 | 37.24 |
| JLQ | 21.62 | 26.08 | 24.27 |
| HZL | 10.43 | 6.97 | 14.17 |
| LW | 7.27 | 7.67 | 7.45 |
| HY | 4.38 | 5.95 | 8.22 |
| LSY | 19.44 | 20.5 | 12.08 |
| SL | 11.07 | 11.47 | 3.26 |
| BD | 0 | 10.43 | 10.06 |
| MQ | 10.43 | 0 | 12.57 |
| KME | 10.06 | 12.57 | 0 |

**Appendix F**

*Complete SBRP Code Used to Compute the Optimized Route*

*#@SBRP ()*

*"""School Bus Routing Problem (SBRP)."""*

*#import tools used in the code#*

pip install ortools

**from** **ortools.constraint\_solver** **import** routing\_enums\_pb2

**from** **ortools.constraint\_solver** **import** pywrapcp

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**import** **csv**

*#import the duration csv file into an array*

reader = csv.reader(open("/Users/stellenshun.li/Desktop/12th Grade/AP Research/Paper/Codes/Duration.csv", "r"), delimiter=",")

x = list(reader)

result = np.array(x).astype("float")

*#store the data for this problem#*

**def** create\_data\_model():

data = {}

data['distance\_matrix'] = result

*#All stops:Hua, AC1, AC2, AC3, AC4, skl, YG, ACb, OV, DH, WY, HL, HT, TJY, OT, IB, TC, JLQ, HZL, LW, HY, LSY, SL, BD, MQ, KME*

*#AC is divided into AC1-AC4 because AC has more than 100 students. But since the maximum capacity of a bus is at most 49,*

*#4 buses are needed to visit AC*

*#demands is the number of students at each stop*

data['demands'] = [2, 35, 35, 35, 34, 0, 4, 4, 1, 3, 1, 1, 1, 1, 3, 3, 3, 6, 3, 4, 10, 6, 3, 31, 3, 3]

*#vehicle\_capacities has the maximum capacity data for all 9 buses*

data['vehicle\_capacities'] = [45, 45, 45, 45, 45, 45, 49, 15, 15]

data['num\_vehicles'] = 9

*#starting points*

data['starts'] = [0, 6, 7, 11, 12, 13, 16, 24, 25]

*#set destination as School A*

data['ends'] = [5, 5, 5, 5, 5, 5, 5, 5, 5]

**return** data

*#prints solution on console*

**def** print\_solution(data, manager, routing, solution):

total\_distance = 0

total\_load = 0

**for** vehicle\_id **in** range(data['num\_vehicles']):

index = routing.Start(vehicle\_id)

plan\_output = 'Route for vehicle {}:**\n**'.format(vehicle\_id)

route\_distance = 0

route\_load = 0

**while** **not** routing.IsEnd(index):

node\_index = manager.IndexToNode(index)

route\_load += data['demands'][node\_index]

plan\_output += ' Stop {0} Students({1}) -> '.format(node\_index, route\_load)

previous\_index = index

index = solution.Value(routing.NextVar(index))

route\_distance += routing.GetArcCostForVehicle(

previous\_index, index, vehicle\_id)

plan\_output += 'Stop {}**\n**'.format(manager.IndexToNode(index))

plan\_output += 'Duration of the route: {} mins**\n**'.format(route\_distance)

plan\_output += 'Number of students of the route: {}**\n**'.format(route\_load)

print(plan\_output)

total\_distance += route\_distance

total\_load += route\_load

print('Total duration of all routes: {} mins'.format(total\_distance))

print('Total number of students of all routes: {}'.format(total\_load))

*#solve the SBRP*

**def** main():

*# Instantiate the data*

data = create\_data\_model()

*# Create the routing index manager.*

manager = pywrapcp.RoutingIndexManager(len(data['distance\_matrix']),

data['num\_vehicles'], data['starts'],

data['ends'])

*# Create Routing Model.*

routing = pywrapcp.RoutingModel(manager)

*# Create and register a transit callback.*

**def** distance\_callback(from\_index, to\_index):

*"""Returns the distance between the two nodes."""*

*# Convert from routing variable Index to distance matrix NodeIndex.*

from\_node = manager.IndexToNode(from\_index)

to\_node = manager.IndexToNode(to\_index)

**return** data['distance\_matrix'][from\_node][to\_node]

transit\_callback\_index = routing.RegisterTransitCallback(distance\_callback)

*#Define cost of each arc.*

routing.SetArcCostEvaluatorOfAllVehicles(transit\_callback\_index)

*# Add Capacity constraint.*

**def** demand\_callback(from\_index):

*"""Returns the demand of the node."""*

*# Convert from routing variable Index to demands NodeIndex.*

from\_node = manager.IndexToNode(from\_index)

**return** data['demands'][from\_node]

demand\_callback\_index = routing.RegisterUnaryTransitCallback(

demand\_callback)

dimension\_name = 'Capacity'

routing.AddDimensionWithVehicleCapacity(

demand\_callback\_index,

0, *# null capacity slack*

data['vehicle\_capacities'], *# vehicle maximum capacities*

**True**, *# start cumul to zero*

dimension\_name)

*# Add Distance constraint.*

dimension\_name = 'Distance'

routing.AddDimension(

transit\_callback\_index,

0, *# no slack*

100, *# minimize vehicle maximum travel duration*

**True**, *# start cumul to zero*

dimension\_name)

distance\_dimension = routing.GetDimensionOrDie(dimension\_name)

distance\_dimension.SetGlobalSpanCostCoefficient(100)

*# Setting first solution heuristic.*

**for** v **in** range(manager.GetNumberOfVehicles()):

routing.ConsiderEmptyRouteCostsForVehicle(**True**, v)

search\_parameters = pywrapcp.DefaultRoutingSearchParameters()

search\_parameters.first\_solution\_strategy = (

routing\_enums\_pb2.FirstSolutionStrategy.PATH\_CHEAPEST\_ARC)

search\_parameters.local\_search\_metaheuristic = (

routing\_enums\_pb2.LocalSearchMetaheuristic.GUIDED\_LOCAL\_SEARCH)

search\_parameters.log\_search = **True**

search\_parameters.time\_limit.FromSeconds(5)

*# Solve the problem.*

solution = routing.SolveWithParameters(search\_parameters)

*# Print solution.*

**if** solution:

print\_solution(data, manager, routing, solution)

**else**:

print('no solution')

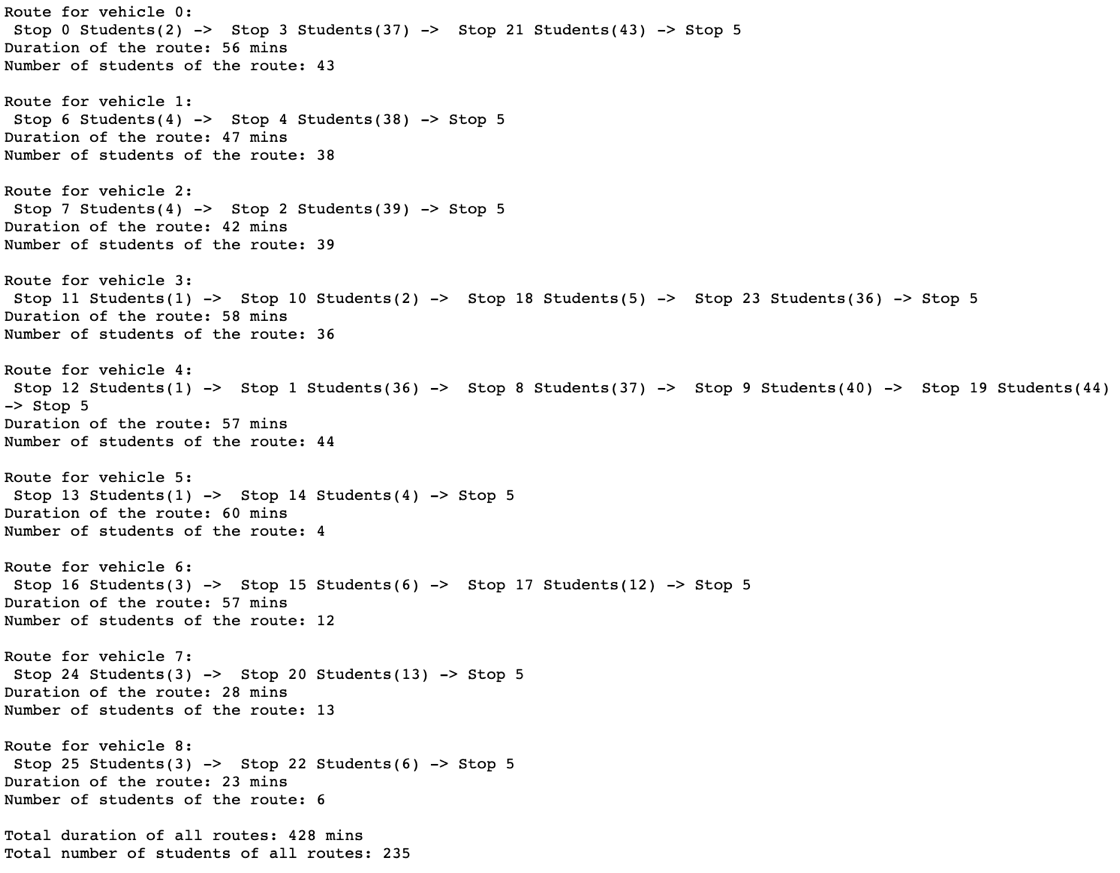
**if** \_\_name\_\_ == '\_\_main\_\_':

main()

data = create\_data\_model()

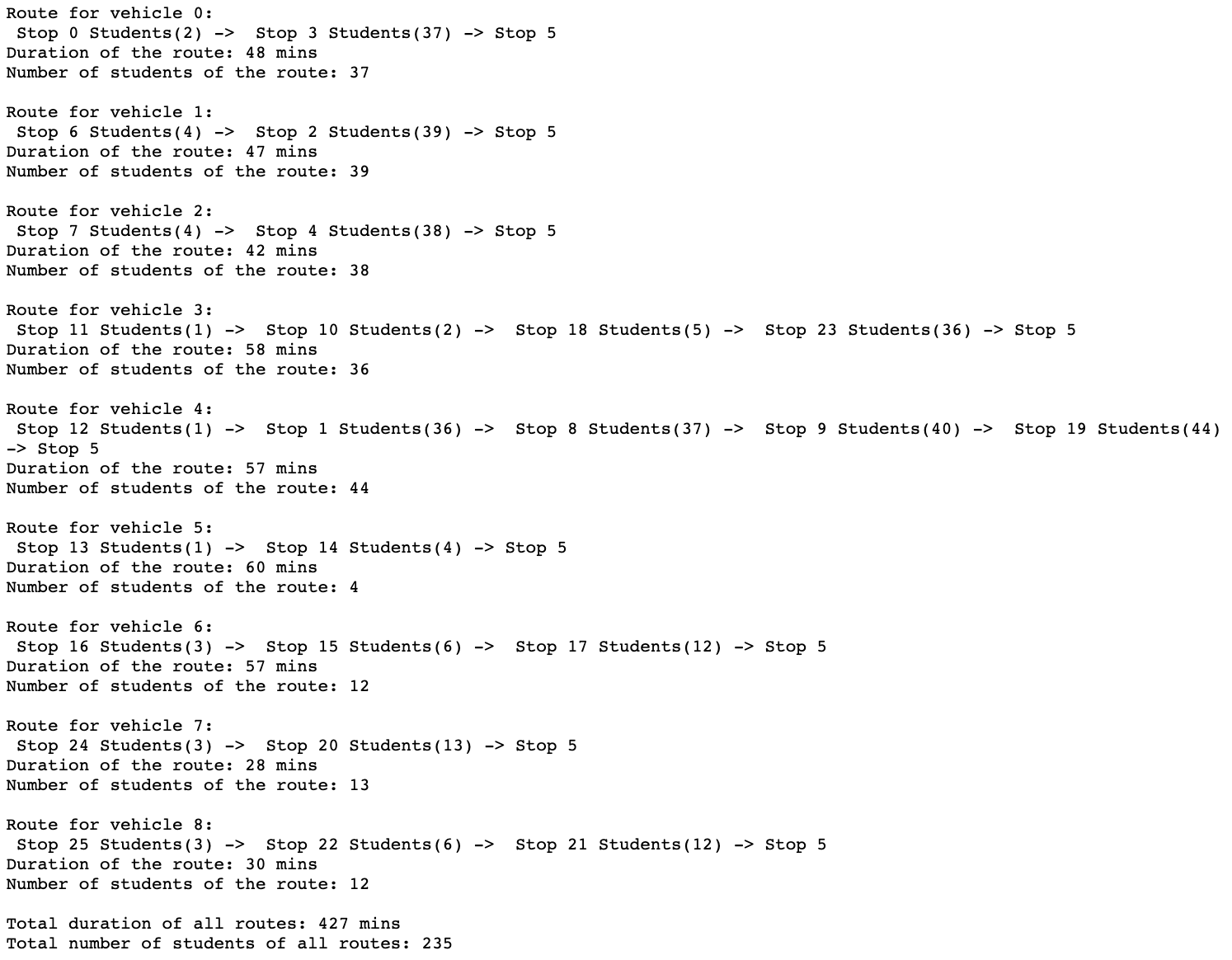
**Appendix G**

*Result From Running the SBRP Code for Four Seconds*

****

**Appendix H**

*Result for Running the SBRP Code for Ten Seconds*

****

1. The SBRP code is adapted from a VRP code in Google OR-Tools, which is a software used for optimization (Google OR-Tools, 2021). [↑](#footnote-ref-1)