Optimizing skip stop scheduling in the Washington Metro Silver Line using genetic algorithms

Anoop Hariharan

Loudoun Academy of Science

Abstract

Skip stopping scheduling is a transportation optimization technique developed to reduce overall travel costs. It is most commonly applied to metro systems that do not have a second rail for express trains to pass. Thus, skip stopping does not require much new investment to reduce overall travel costs. This project initially applied a continuous approximation model to the skip stopping model to identify the cost associated with skip stopping for the Washington Metro Silver Line. However, due to issues with dimensional analysis and derivations of the formulas used in the continuous model paper (Freyss, Giesen, & Muñoz, 2013), the project shifted to a genetic algorithm approach found in a paper applied to a Korean metro line. This approach not only provides a total cost associated with skip stopping, but also gives each station's type (A, B, AB) (Lee, Shariat, & Choi, 2014). Although results could not be collected due to errors with the program, the project did provide a valuable insight into the many factors that need to be considered when developing a realistic skip stop model.

Introduction

Skip stopping has been a major theme in transportation optimization research because it is a valuable tool to increase travel speed without developing much new infrastructure. As such, it has been discussed for many years now in transportation optimization papers. In fact, some of the papers addressing the AB skip stopping method are more than 40 years old (Vuchic, 1976). Though the AB skip stopping method has been found in much of the research on this topic, the exact method by which it is carried out has differed. For example, a continuous model was used to approximate the effect of an AB skip stopping pattern on travel costs in Chile. However, the

limitation of this model was that it could not actually output the specific stations to skip and did not take into account origin-destination data. Thus, it could only approximate the effectiveness of a skip stopping pattern but not show the exact method by which it could be accomplished most efficiently. This paper still found that shorter lines with few stations and low frequency lines would not benefit significantly from skip stopping (Freyss, Giesen, & Muñoz, 2013). Along with continuous models, there have also been combinatorics approaches to the problem of optimizing the skip stop pattern. These approaches usually take into account the different combinations of station types that can be created and use them as input variables for a fitness function of a genetic algorithm. Typically, these fitness functions are related to the total travel time of the passenger. Since the combinations of station types can be exceedingly high to compute using a brute force search, the genetic algorithm is used to arrive at an approximate answer. A paper applying the genetic algorithm approach to the Shenzhen metro in China found that the optimization produced a lower travel time than the traditional "all-stop" pattern. Of note, however, is that this paper utilized a Flexible Skip Stopping Scheme (FSSS) rather than the AB scheme (Zhang, Sun, & Liu, 2017). Genetic algorithms have also been used for the traditional AB skip stopping approach. In a paper utilizing data from a line on the Korean metro (Lee, Shariat, & Choi, 2014), the genetic algorithm took as inputs vectors with three possible values in each row, each corresponding to one of the three station types in an AB pattern. This model also output a set of stations to be skipped and stopped at. In addition, it took into account origin and destination demand (Lee, Shariat, & Choi, 2014). The research question for this project was whether a skip stopping pattern would be beneficial on the Washington Metro Silver Line. Based on previous research

that identified benefits from using a skip stop pattern, it seemed likely that a skip stop pattern would provide a decrease in travel cost (Lee, Shariat, & Choi, 2014)..

Materials and Methods

To develop this project, the Python Spyder application was used. This application provided an environment in which the python code could be written in a readable format and the output of the program could be seen. Within the Python spyder application, various addons were used from third party sources such as matplotlib and xlwings. Matplotlib provided the functionality to output graphs, though this was not possible in this project because the output could not be generated due to errors in the program (addressed in the discussions section). The xlwings addon provided the ability to connect Python to an Excel file, which was the format in which the metro system released their data. The genetic algorithm code was modified from a tutorial source which applied it to a different purpose (Nicolle, 2017). The origin destination data for this project was taken from the planitmetro.com website and modified in Excel to create a file from which the program could easily iterate over.

The genetic algorithm written in Python consisted of a few main functions: fitness, mutation, crossover, and random generation. The fitness function was written almost entirely from scratch and is essentially the same as the travel cost associated with a particular station vector. The other functions were modified from the tutorial source (Nicolle, 2017). The mutation function modifies a random row in the station vector (corresponding to a station) and then

randomly changes its type. This prevents the algorithm from settling at a local minimum on the fitness function. The crossover function chooses two chosen station vectors from the population, some of which are of high fitness and some of which are randomly chosen (to again prevent settling at local minimums). The function then merges these two station vectors to produce a more "evolved" station vector, which is ideally closer to the solution than the parent station vectors. The random generation function actually happens in the beginning, and it creates a population of random station vectors, with the first value of the vector always equal to 2 (the value for an AB station).

The data file that was downloaded from the planitmetro.com website gave the origin destination data for the month of October in 2014. From this data file, a filter was used to identify the average AM peak ridership on weekdays between the origin-destination pairs on the first five stations of the Silver Line. Only the first five stations of the Silver Line were used for this project because they did not share tracks with other lines. This is further elaborated upon in the discussion section. Along with the average AM peak ridership on weekdays for a given OD pair, the standard travel time was also included in another row. This was found by searching for the time taken to travel between the origin and destination on Google Maps. The transit function was used on Google Maps, which made it easy to find the standard travel time between a station pair.

The main component of the genetic algorithm is the fitness function, which took as an input a station vector and output a total travel time cost, also termed fitness. The fitness function took into account factors like transfer time and transfer time.

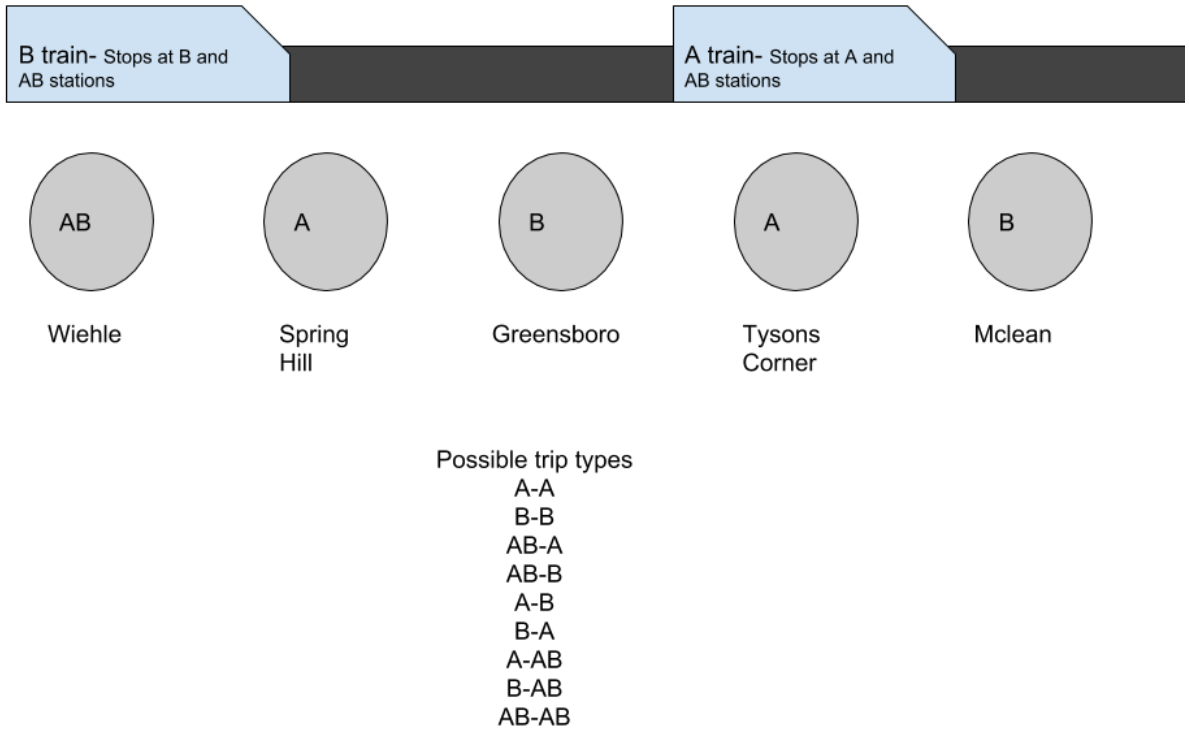


Figure 1: Outline of the AB skip stopping pattern for the five station considered on the silver line. This graphic was also used on the presentation board.

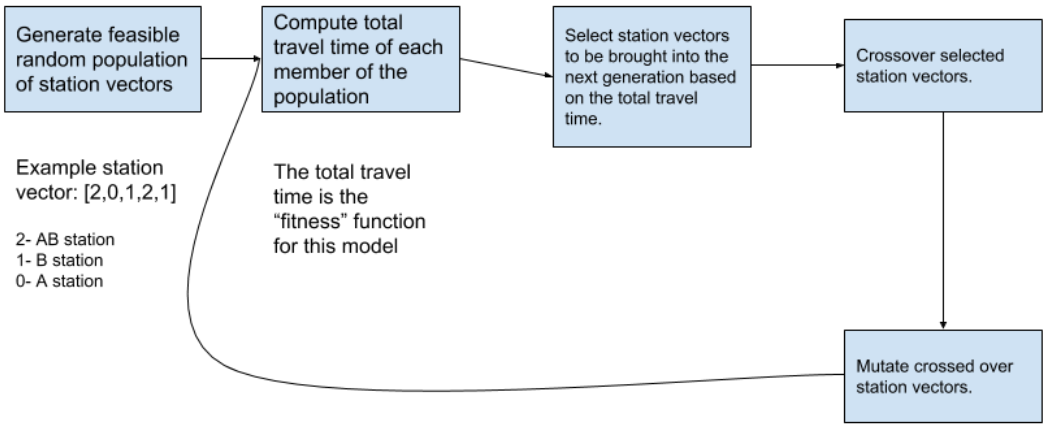
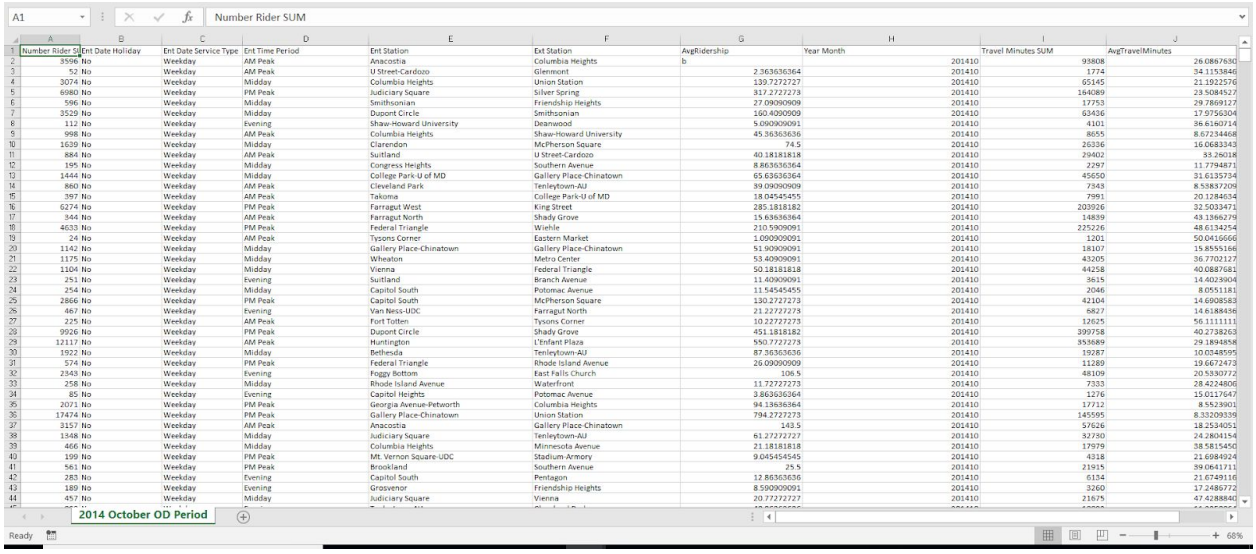


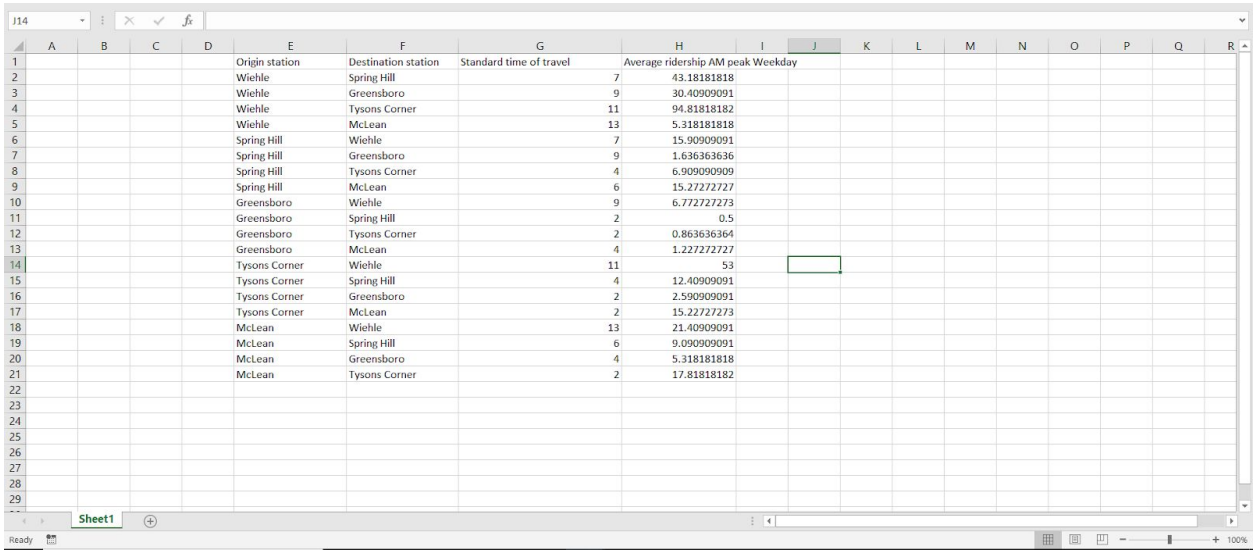
Figure 2: Outline of the genetic algorithm process as it relates to skip stopping. This graphic was also used on the presentation board.

Data



Number Rider Sum	Ent Date Holiday	Ent Date Service Type	Ent Time Period	Ent Station	Ext Station	AvgRidership	Year Month	Travel Minutes SUM	AvgTravelMinutes
3596	No	Weekday	AM Peak	Anacostia	Columbia Heights	b	201410	93808	26.0867630
52	No	Weekday	AM Peak	U Street-Cardozo	Glenmont	2.363636364	201410	1774	34.1153846
3074	No	Weekday	Midday	Columbia Heights	Union Station	139.7272727	201410	65145	21.1922576
6980	No	Weekday	PM Peak	Judiciary Square	Silver Spring	317.2727273	201410	164089	23.5084527
596	No	Weekday	Midday	Smithsonian	Friendship Heights	27.09090909	201410	17753	29.7869127
3529	No	Weekday	Midday	Dupont Circle	Smithsonian	160.4090909	201410	63436	17.9756304
112	No	Weekday	Evening	Shaw-Howard University	Deanwood	5.090909091	201410	4101	36.6160714
998	No	Weekday	AM Peak	Columbia Heights	Shaw-Howard University	45.36363636	201410	8655	8.6723468
1639	No	Weekday	Midday	Clarendon	McPherson Square	74.5	201410	26336	16.0683343
884	No	Weekday	AM Peak	Suitland	U Street-Cardozo	40.18181818	201410	29402	33.260188
195	No	Weekday	Midday	Congress Heights	Southern Avenue	8.863636364	201410	2297	11.7794871
1444	No	Weekday	Midday	College Park-U of MD	Gallery Place-Chinatown	65.63636364	201410	45650	31.6135734
860	No	Weekday	AM Peak	Cleveland Park	Tenleytown-AU	39.09090909	201410	7343	8.5383720
397	No	Weekday	AM Peak	Takoma	College Park-U of MD	18.04545455	201410	7991	20.1284634
6274	No	Weekday	PM Peak	Farragut West	King Street	285.1818182	201410	203926	32.5033471
344	No	Weekday	AM Peak	Farragut North	Shady Grove	15.63636364	201410	14839	43.1366279
4633	No	Weekday	PM Peak	Federal Triangle	Wiehle	210.5909091	201410	225226	48.6134254
24	No	Weekday	AM Peak	Tysons Corner	Eastern Market	1.090909091	201410	1201	50.0416666
1142	No	Weekday	Midday	Gallery Place-Chinatown	Gallery Place-Chinatown	51.90909091	201410	18107	15.8555166
1175	No	Weekday	Midday	Wheaton	Metro Center	53.40909091	201410	43205	36.7702127
1104	No	Weekday	Midday	Vienna	Federal Triangle	50.18181818	201410	44258	40.0887681
251	No	Weekday	Evening	Suitland	Branch Avenue	11.40909091	201410	3615	14.4023904
254	No	Weekday	Midday	Capitol South	Potomac Avenue	11.54545455	201410	2046	8.0551181
2866	No	Weekday	PM Peak	Capitol South	McPherson Square	130.2727273	201410	42104	14.6908583
467	No	Weekday	Evening	Van Ness-UDC	Farragut North	21.22727273	201410	6827	14.6188456
225	No	Weekday	AM Peak	Fort Totten	Tysons Corner	10.22727273	201410	12625	56.1111111
9926	No	Weekday	PM Peak	Dupont Circle	Shady Grove	451.1818182	201410	399758	40.2738263
12117	No	Weekday	AM Peak	Huntington	L'Enfant Plaza	550.7727273	201410	353689	29.1894858
1922	No	Weekday	Midday	Bethesda	Tenleytown-AU	87.36363636	201410	19287	10.0348595
574	No	Weekday	PM Peak	Federal Triangle	Rhode Island Avenue	26.09090909	201410	11289	19.6672473
2343	No	Weekday	Evening	Foggy Bottom	East Falls Church	106.5	201410	48109	20.5330772
258	No	Weekday	Midday	Rhode Island Avenue	Waterfront	11.72727273	201410	7333	28.4224806
85	No	Weekday	Evening	Capitol Heights	Potomac Avenue	3.863636364	201410	1276	15.0117647
2071	No	Weekday	PM Peak	Georgia Avenue-Petworth	Columbia Heights	94.13636364	201410	17712	8.5529901
17474	No	Weekday	PM Peak	Gallery Place-Chinatown	Union Station	794.2727273	201410	145595	8.3320933
3157	No	Weekday	AM Peak	Anacostia	Gallery Place-Chinatown	143.5	201410	57626	18.2534051
1348	No	Weekday	Midday	Judiciary Square	Tenleytown-AU	61.27272727	201410	32730	24.2804154
466	No	Weekday	Midday	Columbia Heights	Minnesota Avenue	21.18181818	201410	17979	38.5815450
199	No	Weekday	PM Peak	Mt. Vernon Square-UDC	Stadium-Armory	9.045454545	201410	4318	21.6984924
561	No	Weekday	PM Peak	Brookland	Southern Avenue	25.5	201410	21915	39.0641711
283	No	Weekday	Evening	Capitol South	Pentagon	12.86363636	201410	6134	21.6749116
189	No	Weekday	Evening	Grosvenor	Friendship Heights	8.590909091	201410	3260	17.2486772
457	No	Weekday	Midday	Judiciary Square	Vienna	20.77272727	201410	21675	47.4288840

Figure 3: Raw data file from planitmetro.com



Origin station	Destination station	Standard time of travel	Average ridership AM peak Weekday
Wiehle	Spring Hill	7	43.18181818
Wiehle	Greensboro	9	30.40909091
Wiehle	Tysons Corner	11	94.81818182
Wiehle	McLean	13	5.318181818
Spring Hill	Wiehle	7	15.90909091
Spring Hill	Greensboro	9	1.636363636
Spring Hill	Tysons Corner	4	6.909090909
Spring Hill	McLean	6	15.27272727
Greensboro	Wiehle	9	6.772727273
Greensboro	Spring Hill	2	0.5
Greensboro	Tysons Corner	2	0.863636364
Greensboro	McLean	4	1.227272727
Tysons Corner	Wiehle	11	53
Tysons Corner	Spring Hill	4	12.40909091
Tysons Corner	Greensboro	2	2.590909091
Tysons Corner	McLean	2	15.22727273
McLean	Wiehle	13	21.40909091
McLean	Spring Hill	6	9.090909091
McLean	Greensboro	4	5.318181818
McLean	Tysons Corner	2	17.81818182

Figure 4: Data file including standard travel time (from Google Maps) and average AM peak ridership on weekdays (taken from the filtered raw data file)

Results

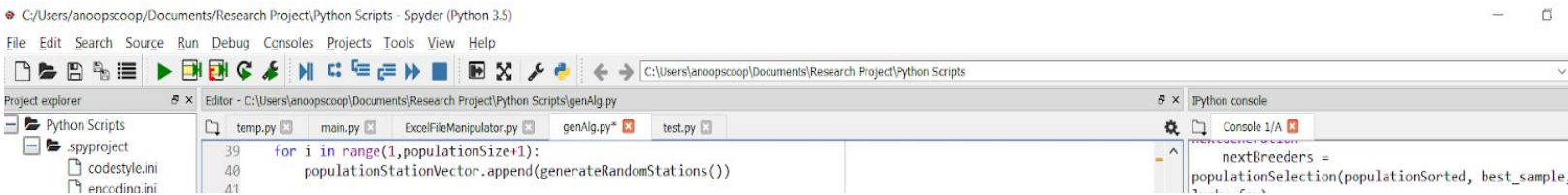


Figure 5: Snippet of fitness function code- travelTime()

Due to the errors in the program, an output could not be generated.

Discussion

It can be concluded that there is certainly the data available to conduct a genetic algorithm based skip stop optimization of the Washington Metro Silver Line. However, the multitude of different factors that have to be considered make its development time much longer than a continuous model like that applied to the Chile metro (Freyss, Giesen, & Muñoz, 2013). However, the benefit of being able to view the exact stops that can be skipped (which a continuous model cannot do) seems to be more useful.

Since results could not be collected, the effectiveness of the model could not be tested. However, the conceptual limitations of the model can still be addressed. The fitness function that was ultimately developed for the genetic algorithm had multiple limitations. First, it assumed a waiting time of zero when computing the total travel cost. Though this simplified the function, it is unrealistic because most people will not arrive exactly at the time that a train arrives at a station. This error would be seen especially in skipped stations, because the maximum headway of a passenger arriving at a skipped station would be higher than at an all stop station. The effect of this is that the total travel cost would appear lower than the actual value if waiting time were

taken into account. Indeed, the model used on the Korean metro addressed this issue (Lee, Shariat, & Choi, 2014). In addition, no collision constraints were used unlike the model applied to the Korean metro by past research (Lee, Shariat, & Choi, 2014). It was assumed that for this model, collisions would not be an issue since the maximum time that could be saved was four minutes (as only five stops were considered on the silver line). Since the highest frequency on the silver line is still 8 minutes, it was assumed that there would always be room between two trains even if 4 minutes were saved on a journey on the first five stations. Whether this assumption is mathematically and practically valid needs to be further verified. Another limitation of this project's work was that the number of station had to be limited to avoid track sharing with the Orange Line.

Future work on this project would include developing a way to incorporate more stations on the Silver Line that are shared with other lines. This would require additional constraints to prevent collisions and calculate overall travel cost. Other improvements to this project that could be made are simplification of the code to speed up the algorithm. This could be accomplished through the use of specialized libraries or even vectorization (if possible) to speed up the process of iterating through many loops. In addition, more factors such as delays could be accounted for in the model to increase its realism. The effect of this would be higher computation times, but given that this is not a real time skip stopping model, it would likely not be a significant hurdle for a train system operator.

References

Freyss, M., Giesen, R., & Muñoz, J. C. (2013). Continuous approximation for skip-stop operation in rail transit. Transportation Research Part C: Emerging Technologies, 36, 419-433.

Lee, Y. J. (2012). Mathematical modeling for optimizing skip-stop rail transit operation strategy using genetic algorithm. Morgan State University, Department of Transportation and Urban Infrastructure Studies: Baltimore, MD, USA.

Nicolle, L. (2017). Getting started with genetic algorithms: A tutorial. Retrieved May 30, 2018, from https://blog.sicara.com/getting-started-genetic-algorithms-python-tutorial-81ffa1dd72f9

Vuchic, V. R. (1976). Skip-stop operation: High speed with good area coverage. UITP Revue, 114.

Zhang, P., Sun, Z., & Liu, X. (2017). Optimized Skip-Stop Metro Line Operation Using Smart Card Data. Journal of Advanced Transportation, 2017.