Proceedings of the 2012 Joint Rail Conference
JRC2012
April 17-19, 2012, Philadelphia, Pennsylvania, USA

JRC2012-74188

OPTIMIZING SKIP STOP SERVICE IN PASSENGER RAIL TRANSPORTATION

Samuel L. Sogin
University of Illinois at Urbana-Champaign
B118 Newmark Civil Engineering Laboratory
205 N. Mathews Ave.
Urbana, IL 61801, USA

Brennan M. Caughron, EIT
University of Illinois at Urbana-Champaign
B118 Newmark Civil Engineering Laboratory
205 N. Mathews Ave.
Urbana, IL 61801, USA

Samantha G. Chadwick, EIT
University of Illinois at Urbana-Champaign
B118 Newmark Civil Engineering Laboratory
205 N. Mathews Ave.
Urbana, IL 61801, USA

ABSTRACT

Two-track passenger rail lines typically operate with all trains serving every station. Without additional infrastructure, transit planners have limited options to improve travel times. Service could be improved by operating a skip-stop service where trains only serve a subset of all the station stops. A skip-stop pattern must find an optimal balance between faster passenger travel times and lower service frequencies at each station. A mixed integer formulation is proposed to analyze this tradeoff; however, the mixed integer formulation could not scale efficiently to analyze a large scale commuter line. A genetic algorithm is presented to search the solution space incorporating a larger problem scope and complexity. In a case study of a Midwest commuter line, overall passenger travel time could be decreased by 9.5%. Both analyses can give insights to transit operators on how to improve their service to their customers and increase ridership.

INTRODUCTION

To improve service for passengers, rail service planners have limited options for reducing overall travel time. Most two track passenger rail and transit lines operate with trains stopping at all stations. Full express service is often not possible because of high traffic density coupled with infrastructure constraints. For passenger lines with frequent stops, enhanced train acceleration or maximum speed is only useful up to a point. One way of reducing passenger travel time is to introduce service that, on occasion, skips stops at stations with lower demand. By eliminating the delay incurred by decelerating, dwelling at a station, and then accelerating, average speed on a line can be increased. On a line with many trains and stations, it is a nontrivial problem to develop an optimal stopping pattern and schedule.

Two methods of developing an optimal skip-stop schedule are investigated. First, a mixed integer program is presented to solve the problem. The computational complexity of this method led to the development of a second heuristic method to search the solution space of lines with large numbers of stations and trains. After developing a random schedule, a genetic algorithm will iteratively crossover and mutate schedules with the goal of reducing overall travel time. A case study line was used to help develop both the mixed integer and heuristic search methods. The Metra Union Pacific (UP) North Line serving Chicago's north suburbs was selected because of the current lack of significant express service in the existing schedule. The subsequent analysis focuses solely on the morning inbound commute as this moves the most passengers in a short amount of time. In addition, the analysis was constrained to operating the same number of trains as the existing schedule in the case study time period. All methods proposed could be applied to the outbound commute or afternoon rush hour.

After inputting data from the case study line, the genetic algorithm was able to converge to a solution. The solution generated is dependent on the tradeoff between overall passenger travel time, and the convenience of having several trains to choose from. This tradeoff coefficient β could change depending on the passengers' value of time waiting for a train at a station. A sensitivity analysis is also presented showing the tradeoff between passenger convenience and total travel time.

The methods presented here could be applied to a variety of passenger rail transportation scenarios outside of the heavy rail commuter line in the case study. Although high speed and intercity passenger rail typically feature fewer stations than most commuter lines, skip-stop service could be a way of increasing passenger access to the line. New stations can be constructed along the line at low demand areas while stop-skipping preserves low travel times between high demand stations .

LITERATURE REVIEW

In this segment we present a literature review of other papers covering train routing and scheduling optimization problems.

Bodin et al. developed a mixed-integer program with the goal of optimizing operations within yards as well as between-yard movements [1]. The authors incorporate real-world constraints such as yard capacity and minimum block size. The model includes a delay function describing the length of time a car will wait at its origin before leaving for its destination, given the number of other cars going to that destination.

Ghoneim et al. analyzed zone scheduling for the Calgary Light Rail system using dynamic programming to determine the number and location of zones [2]. The authors incorporated headway constraints but did not consider the inconvenience to passengers of less O-D pair service.

Haghani developed a time space network model to examine the interaction between train routing, makeup, and empty car distribution decisions [3]. The routing network included both local and express links. The model incorporates a penalty cost for unfilled demand of empty cars. The model uses a heuristic decomposition algorithm to develop train routing and makeup decisions.

Bussieck et al developed models for both routing and scheduling passenger rail service [4]. The train routing problem is formulated as a mixed integer program with two alternate objective functions, one that minimizes the service cost of a capacity-adequate schedule and a second that maximizes the number of direct passengers not changing trains. For passengers without a direct train, it is assumed that they switch from a faster to a slower train at the earliest possible point and switch back to a slower train at the latest possible point. The scheduling model minimizes passenger waiting time in transfers.

Gorman analyzed the Santa Fe railroad network for improved operational cost and service to its customers [5]. The author used a combination of a genetic algorithm and tableau search to determine the number of necessary train frequencies on the intermodal network over a set of pre-generated routes that were both all-stop, and a subset of skip-stop routes.

Chang et al. analyzed the Taiwan High Speed Rail operating plan to select station stopping patterns using a linear approximation [6]. The proposed technique considers the dual objectives of minimizing passenger travel time and the operating cost to the railroad. The analysis assumed a pre-determined set of stopping patterns and did not consider the waiting time of passengers between train frequencies.

INPUT DATA

Data from an existing rail operation was first collected to serve as a baseline scenario for comparison with any new alternatives developed through the mixed integer or heuristic search methods. Ridership information from Chicago-area Metra lines was collected from the Chicago Regional Transit Asset Management System (RTAMS) website. RTAMS is maintained with the support of all major Regional Transit Authority (RTA) agencies, including Metra. Detailed passenger counts of both boardings and alightings for each station are available on the site. The most recent data for the Metra UP North Line is from the year 2006 [7].

Although information is available that shows the total boardings and alightings at each station, there is not information on the distribution of destinations for a given origin station. To develop this information, a gravity model ridership distribution was created using the product of boardings and alightings for an O-D pair divided by the segment distance raised to a coefficient. For this study, a distance coefficient of 0.5 was used. Using the gravity distribution, along with total line ridership from 2010, an updated ridership matrix was created [8].

$$\delta^{OD} = \frac{B^{O} A^{D}}{\sqrt{d^{OD}}}$$

δ^{OD} = OD travel demand
B^{O} = Total boardings at origin
A^{D} = Total alightings at destination
d^{OD} = Distance between points O and D

The equation above is used to provide relative weights describing where passengers are travelling at each station. The demand distribution was normalized by the number of people who boarded the train at the origin. This ensured that every person who boarded the train also disembarked at a downstream station.

In addition to passenger information, minimum travel times must be generated between any OD pair on the line. The present Metra schedule has some examples of skipped stops, but not enough to populate a full OD matrix of travel times [9]. Using the existing schedule, all cases of skipped stops were analyzed to determine an average stop delay τ_{avg} of 1 minute, 26 seconds. Ideally, τ should be a function of train speed, acceleration characteristics, and the number of boarding and alighting passengers. In the interest of simplicity we assumed the average station delay held true for all stations. Using a baseline schedule and τ_{avg}, a complete set of minimum express travel times was created. In the existing schedule, there is a travel time difference between inbound and outbound trains; the constructed table also reflects this difference.

MIXED INTEGER OPTIMIZATION

The Optimal Skip-Stop Optimization Problem could be formulated as a mixed integer program (MIP). The station pattern can be considered to be a directed network as shown in Figure 1. Each station is represented by a node, and each link represents the next downstream station at which a train could stop. The form of this problem is similar to a shortest path network problem. Each train will start at the exurb terminal and find the shortest path to the central business district (CBD) while minimizing the total time of passengers on the train.

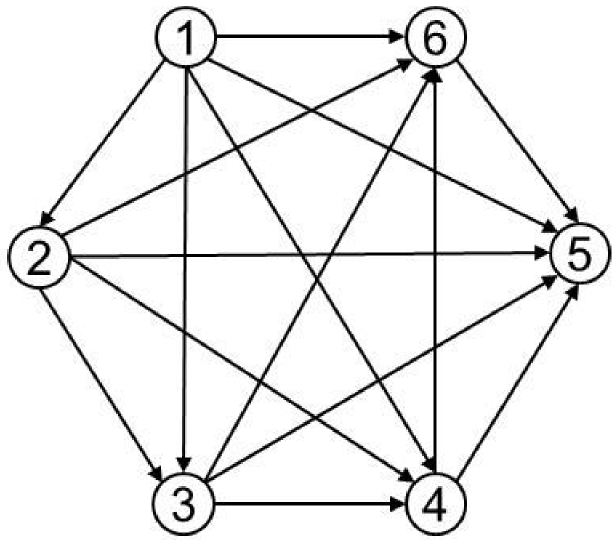

Figure 1: Skip-Stop Network Representation

Formulation

Let the set *{S}* represent the set of stations being considered and let the directed links be represented by $\{i,j\}$ where a train departs station i and arrives at station j. The set $\{O,D\}$ represents the origin and destination pairs. Lastly, let $\{k\}$ be the set of trains that are travelling in the morning rush-hour service where k corresponds to the k^{th} train to leave the suburbs for the CBD. Tij is the time to travel directly from station i to station j including acceleration, braking, and stop delay to load passengers. To simplify the problem, all trains are considered to originate at the furthest station from the CBD and terminate at the CBD.

Table 1: Definition of Sets Used in MIP

Set	Description
$\{i,j,O,D\}$	Set of nodes representing stations
$\{k\}$	Set of trains in morning rush hours
$\{ ij \}$	Links between stations
$\{OD\}$	Origination-Destination pairs

x_{ijk} is a binary variable that represents the choice of whether train k will provide direct service from upstream station i to downstream station j. y_{ijk}^{OD} represents the flow of passengers travelling between station i and station j on train k who boarded at station O and will disembark at station D. There is also the possibility of unfilled demand where a station will not receive any service from the scheduled trains. The unfilled demand for an origin destination pair is Z^{OD}.

Table 2: Decision Variables in MIP

Variable	Description
$x_{ijk} =$	{ 1 if transit service is provided for link ij for train k; 0 otherwise}
$y_{ijk}^{OD} =$	Flow of people on link ij using train k to travel from O to D
$Z^{OD} =$	Unfilled demand

The MIP Skip Stop Optimization can now be formulated as follows:

$$\text{Minimize: } \sum_O \sum_D \sum_k \sum_i \sum_{j:j>i} T_{ij} y_{ijk}^{OD} + \sum_O \sum_D \gamma Z^{OD} \qquad (1)$$

$$\sum_{j:j>i} x_{ijk} \leq 1 \qquad \forall i,k \qquad (2)$$

$$\sum_j \left(x_{ijk} - x_{jik}\right) = \begin{Bmatrix} 1 \text{ if departure} \\ -1 \text{ if terminal} \\ 0 \text{ otherwsie} \end{Bmatrix} \qquad \forall i,k \qquad (3)$$

$$\sum_k \sum_j y_{(i=O)jk}^{OD} = \delta^{OD} - Z^{OD} \qquad \forall O,D \qquad (4)$$

$$y_{ijk}^{OD} \leq \sum_j M x_{(i=O)jk} \qquad \forall i,j,k,O,D \qquad (5)$$

$$y_{ijk}^{OD} \leq \sum_i M x_{i(j=D)k} \qquad \forall i,j,k,O,D \qquad (6)$$

$$\sum_j y_{ijk}^{OD} - y_{jik}^{OD} = \begin{Bmatrix} \sum_j y_{ijk}^{OD} : i = O \\ \sum_j -y_{jik}^{OD} : j = D \\ 0 \end{Bmatrix} \qquad \forall i,k,O,D \qquad (7)$$

$$x_{ijk} = \{1,0\} \qquad \forall i,j,k,O,D \qquad (8)$$

$$Z^{OD} \geq 0 \qquad \forall O,D \qquad (9)$$

$$y_{ijk}^{OD} \geq 0 \qquad \forall i,j,k,O,D \qquad (10)$$

The objective function (1) serves to minimize total passenger travel time using the commuter rail service as well as any unfilled demand. The γ term is determined by the planning agency and is a cost coefficient to transform unfilled demand to units of time. γ should be set high enough such that it is more cost efficient to run trains than not serve passengers. With more morning trains, the estimation of γ is not as important as it becomes more likely for low demand stations to be served by one or more trains. Constraint (2) states that each train can leave its current station for only one downstream station. Constraint (3) is a mass conservation flow constraint for the trains travelling through the network. Constraint (4) defines the unfilled demand by setting the total actual flow of travelers for an origin-destination (O-D) pair across all trains equal to the theoretical demand minus the unfilled demand for that particular O-D pair. Constraints (5) and (6) guarantee that a train must service both the origin and destination of an O-D pair for passengers to utilize the commuter railroad. Constraint (7) is a mass conservation flow constraint for passengers. Lastly, constraints (8) through (10) define the variable types.

Case Study

A hypothetical commuter rail line is studied to demonstrate the potential of the proposed skip stop optimization framework. This line has 8 total stations including the terminals and 6 trains in the morning rush hour. The MIP was coded in GAMS and solved using CPLEX. The fastest run time from the suburban terminal to the CBD is 8 minute express service and the slowest run time is 14 minute local service. Table 3 shows the time between stations and Table 4 is the demand for each O-D pair.

Table 3: Travel Times Between Stations

		Origination Station						
		1	2	3	4	5	6	7
Destination Station	2	2						
	3	3	2					
	4	4	3	2				
	5	5	4	3	2			
	6	6	5	4	3	2		
	7	7	6	5	4	3	2	
	8	8	7	6	5	4	3	2

Table 4: Passenger Demand Between Stations

		Origination Station						
		1	2	3	4	5	6	7
Destination Station	2	10						
	3	10	5					
	4	10	10	5				
	5	50	100	25	15			
	6	10	10	5	0	5		
	7	10	5	5	0	20	2	
	8	200	150	25	175	125	90	50

Table 5: Optimal Schedule

Station ↓	Train #1	Train #2	Train #3	Train #4	Train #5	Train #6
STA 1	0	0	0	0	0	0
STA 2	2	2	-	2	-	2
STA 3	4	-	-	4	3	-
STA 4	6	-	-	6	5	-
STA 5	-	-	-	8	-	6
STA 6	-	-	-	10	-	8
STA 7	-	-	-	12	-	-
STA 8	11	9	8	14	10	10

Limitations

This model provides the optimal stopping pattern that minimizes travel time; however it does not address rider behavior. The model assumes that an individual passenger travelling from an origin to a destination will ride the train that takes him to his destination the fastest, not necessarily the most convenient. For example, there are six options to travel from Station 1 to Station 8. However, under this model, all the passengers who need to travel from Station 1 to Station 8 will take Train #3 because it is the fastest train. A skip-stop model should have passengers utilize all services between Station 1 and Station 8 with more passengers using the faster trains than the slower trains. In the hypothetical line studied, only one train served Station 7. These passengers do have their demand fulfilled by the schedule, but are still inconvenienced by only having the one service. A more advanced skip-stop model should incorporate the average waiting time between trains. These additions to the model quickly transform this into a non-linear mixed integer program. Lastly, the model becomes too large for GAMS and CPLEX to solve if more than 15 stations are considered.

HEURISTIC SEARCH

In order to address the limitation of the MIP model, a heuristic search based of a genetic algorithm was developed in PYTHON [10]. The stopping pattern of a train can be represented by a binary string. The length of the string is the number of stations. An entry of "1" indicates that the train stops at that station, and a "0" indicates that the station was skipped [6]. The strings for the proposed stopping pattern can then be merged into one string representing a potential schedule. Table 6 takes the last 7 stations on the Metra Union Pacific North Line in Chicago and encodes the binary representation of the schedule from the stopping pattern of 4 trains.

Table 6: Binary Encoding of Skip Stop Service

	Central	Davis	Main	Rogers Park	Raven-swood	Clybourn	Ogilvie	Binary String
Train #1	7:34	7:38	–	–	–	7:51	8:02	1100011
Train #2	7:41	7:44	7:47	7:50	7:55	8:02	8:12	1111111
Train #3	–	7:51	–	–	8:01	8:08	8:18	0100111
Train #4	8:13	–	8:18	8:22	–	8:31	8:41	1011011
Schedule Code: 1100011111111101001111011011								

Search Algorithm Formulation

The final search algorithm is described in Figure 2.

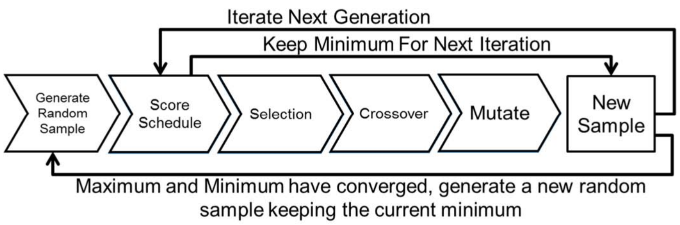

Figure 2: Proposed genetic algorithm to search for improved skip-stop service schedules

Generate Random Sample

An initial sample of potential schedules is generated randomly with a restriction that the train must serve the terminals in the farthest suburb and the CBD.

Score Schedules

The score of any schedule is based off total riding time of all the passengers riding the train plus any unfilled demand, and an inconvenience penalty experienced by the passengers. This inconvenience penalty for an O-D pair is derived from a tradeoff coefficient β, the passenger demand for that O-D pair, and the average headway for the O-D pair. The β coefficient in the inconvenience penalty incorporates both a time value of passengers waiting at stations and the degree that the algorithm is allowed to skip low demand stations. Total score of each schedule is calculated as follows:

$$\sum_{O}\sum_{D}\sum_{k} T_{ODK}Y_{ODK} + \beta\sum_{O}\sum_{D}\frac{\delta_{OD}H}{\pi_{OD}} + \gamma\sum_{O}\sum_{D} Z_{OD}$$

O = Originations
D = Destinations
K = Train Services
T_{ODK} = Time for train K to travel from O to D
Y_{ODK} = Number of passengers going from O to D on train K
π_{ODK} = Number of trains serving O to D on train K
δ_{OD} = Passenger demand from O to D
Z_{OD} = Unfilled demand between O and D
H = Length of rush hour
β = Inconvenience cost penalty
γ = Cost of unfilled demand

Unlike the MIP formulations, passengers are allowed to take any train to travel from their respective origins to their destinations. The passengers are not limited to just the fastest train. Passengers are split across all trains that serve their origin and destination with faster services carrying more passengers than slower services. The split of passengers between train services to the same destination is calculated as follows:

$$Y_{ODK} = \delta_{OD}\frac{(T_{ODK})^{-1}}{\sum_k (T_{ODK})^{-1}}$$

For example, consider if there were only four trains serving an O-D pair having run times 24, 20, 20, and 16 minutes. The ridership would be split at 20% for the slowest train, 25% each for the average trains, and the fastest train would carry 30% of the total ridership.

Selection

Schedules were selected randomly based on their total score determined in the scoring module. Initially, all schedules have a relatively equal chance of being selected for the next iteration. As the algorithm iterates, schedules that better minimize travel time and inconvenience are more likely to be selected. This selection behavior is modeled by using a fitness function to scale the raw scores of schedules. Initially, the fitness function equalizes the schedule score to explore more of the solution space early in the search process. Later in the process, the fitness function will increase the disparity between schedules. This allows for very small improvements to be exaggerated late in the search process to ensure that these small improvements are not overlooked by the algorithm. The fitness value is calculated as follows:

$$F = (M - X)^{\frac{1.5i}{L}}$$

F = Fitness score
M = Maximum score of sample
X = Current score of schedule being analyzed
i = Current iteration index
L = Iteration limit

The probability of selecting a schedule out of the current pool is then calculated based off the fitness scores of individual schedules. Higher fitness scores will be more likely to be selected.

$$P_j = \frac{F_j}{\sum F_j}$$

The schedule with the smallest score in any sample will automatically be selected for the next generation for further analysis.

Crossover

Once the schedules are selected, they will then be paired off. A random number is chosen to split the binary representation of the schedule. The pair of schedules will then swap their binary bits after this point. The crossover module creates new schedules based off the best of previous schedules. Figure 3 demonstrates how this works between the stopping patterns of two trains. Train #1 initially skips stations 3 through 5, and Train #2 stops at each station. The next generation of stopping patterns incorporates Train #1 skipping stations 4 and 5, while Train #2 stops at all stations except station #3.

Figure 3: Crossover of Schedules

Mutation

The new schedules created in the crossover module are then passed through the mutation module. There's a very small probability that the mutation module will flip a random station from being skipped to being served and vice versa. This allows the search algorithm to search for new schedules if the current sample becomes too homogenous. The new schedule is checked to make sure that all trains start and stop at terminals before it is allowed to continue on to the next iteration. If the maximum score of the sample converged to the minimum, then the mutation probability increases significantly to generate a new search pool. In this case, 98% of the current pool is thrown out and replaced with mutant schedules and the previous minimum is retained.

Proof of Concept

There are 2^{350} possible solutions for the Union Pacific North Line problem. One major drawback of using a genetic algorithm to find a solution is that it is difficult to verify if the search has actually found the optimal solution. One way to verify that the algorithm is correct is to input a problem with sufficient complexity but also with a known solution. To test the robustness of the algorithm, a hypothetical commuter rail line similar to that of the MIP formulation is analyzed. This problem differs slightly from the MIP problem because of the increased complexity of the assumptions. The GA considers the inconvenience to passengers and will distribute passengers across all trains that serve their origin and destination. Additionally, this problem simplifies the MIP problem by limiting the number of trains to only 3 trains. These trains will operate within a 60 minute period and the β tradeoff coefficient is assumed to be 0.25. The time between stations and origin-destination demand are summarized in Table 3 and Table 4 respectively. There are 262,144 possible solutions to this problem. Each of the solutions is evaluated for total passenger travel time and inconvenience time. The optimal minimum is 248 hours, where two trains will stop at all eight stations and one train will skip Station 3 and Station 4. There are three solutions that will incorporate this stopping pattern. The genetic algorithm was then applied to this simple problem with a sample size of 8. After 11 iterations, the genetic algorithm found one out of the three global minima of the solution space. In total, only 88 potential schedules were evaluated instead of the 262,144 that enumeration would evaluate. The genetic algorithm can significantly decrease the number of computations needed to arrive at a solution.

Case Study: Union Pacific North Line

The search algorithm was applied to the Metra Union Pacific North Line for the morning commute. The current schedule has 14 trains between 6 AM and 9 AM serving 27 stations with very limited skip-stop service. Each potential schedule for the line is represented by a 378-bit chromosome. The demand between stations was estimated using a gravity model based on the distance between stations, the number of boardings and number of alightings at each O-D pair. The time between stations was determined from Metra's published timetable for its passengers, and the distance was determined from Google Maps. The coefficient for the headway term was assumed to be 0.25, assuming that passengers on Metra are familiar with the published schedule and show up closer to when the train departs their origin station. After 10,000 iterations of the Genetic Algorithm, the best schedule found is shown in Appendix A.

The total travel times for all 14 trains are within four minutes of each other, indicating that average speed is similar across all 14 stopping patterns. Having similar average speeds implies that there is not a need for an overtake maneuver within the schedule. If there was an overtake, then there would have to be additional tracks added to accommodate the manoeuver. Ravinia Park is a station for a concert venue that is only in service for evenings in the summer. The search algorithm was correct in not providing any service to Ravinia Park. In comparison with the existing schedule [9], the new schedule would reduce total passenger travel time by 9.5%. However, the total inconvenience (headway) penalty for the new schedule is 24.4% higher than the existing schedule. Applying McFadden's utility from the BART feasibility study, demand for the rail line would increase 3.8% due to the time savings [11]. The total trip time from Kenosha to Chicago on average is 10 minutes faster than the existing schedule. This improvement may not be large enough to allow for lower cycle times to decrease the fleet size of the service.

Table 7: Comparison of Existing Schedule to Proposed Schedule with More Aggressive Skip Stop Service

	Existing Schedule	New Schedule
Passenger Travel Time (Hours)	6,845	6,194
Frequency Penalty (Hours)	1,036	1,289
Total Utility (Hours)	7,882	7,482

Computational Efficiency

For passenger lines with a level of complexity similar to the case study presented here, the genetic algorithm is able to converge on an optimal solution in about 1,000 iterations as shown in Figure 4. This problem is looking for very small improvements over the base case, so the initial proposed solutions are already within 3% of the final value. On problems with higher complexity, this number could be expected to increase. Using an Intel Core i5 CPU at 2.8 GHz the computation time can vary from 2 hours using a small sample size with 3-5 simultaneous populations to 48 hours using a large sample size with 10-20 simultaneous populations. Larger iteration limits will also increase the computation time.

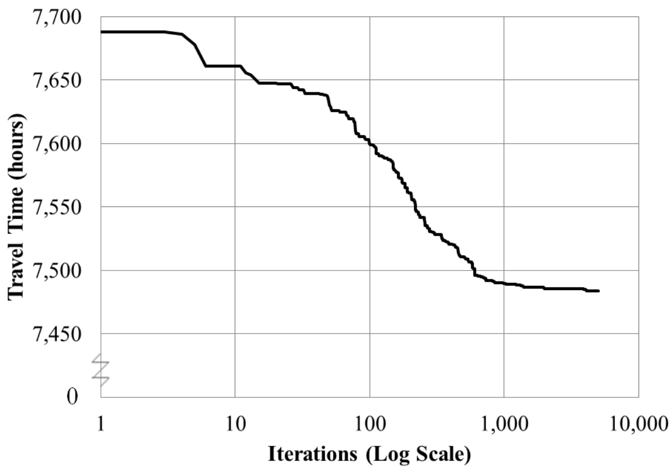

Figure 4: Computational Efficiency

Schedule Development

Before adopting a proposed schedule from the search algorithm, the service pattern must be tested for feasibility with existing conditions. To avoid building new infrastructure, the new schedule must be able to accommodate all planned trains without interference. A train schedule requiring overtaking moves on the adjacent 2nd main track would likely not be feasible given current levels of outbound morning commute traffic.

The Metra Union Pacific North Line uses a three block, four aspect signal system. To ensure that a train does not reduce speed from a free flow condition, a minimum spacing of three block lengths must be maintained between all trains on the schedule. Block lengths vary slightly along the line to compensate for different stopping distances expected with changes in speed and grade. Using aerial photography, an average block length of 0.88 miles and minimum train spacing of 2.64 miles was determined [12].

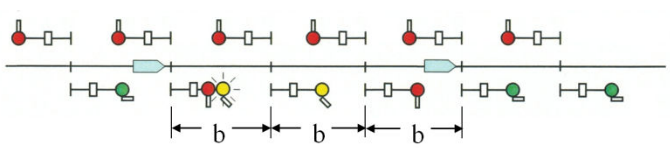

Figure 5: Illustration of minimum train spacing

Using the proposed stopping pattern determined by the search algorithm, a time space diagram is plotted and shown in Figure 6. Initially, a schedule was developed where train departures were uniformly spaced from the station of origin such that all of the trains completed their trips from 0600 hours to 0900 hours. With this initial schedule pattern, some of the trains experienced interference with each other along the line. Two methods can be used to alter the schedule and eliminate train interference. The first method is to change the order of departures from the initial station. The second method is to add additional time between departures at the initial station. In both methods it is important not to create service gaps at stations that are more frequently skipped. As much as possible, the headways at any given station should be uniform throughout the schedule. Using a combination of transposing stopping patterns and adding as many as 5 minutes to the uniform departure times, a final proposed schedule was developed.

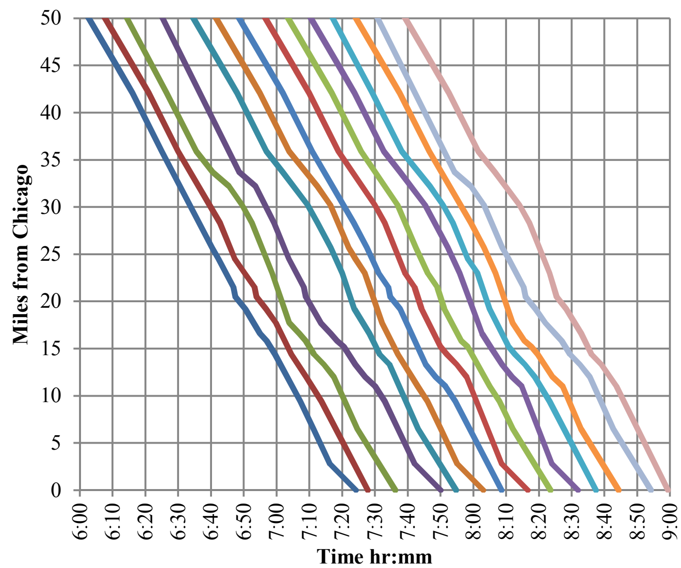

Figure 6: Time Space Diagram of Proposed Schedule

Sensitivity Analysis

There are various sources of error within the inputs to the genetic algorithm. There is sample error from the measurements of the total boardings and alightings at each station. This error is difficult to correct for because there is no data on the variation of passengers at each station. Additionally, the gravity model assumed that ridership was proportional to the square root of distance between stations. If a larger exponent on the distance term was chosen, passengers would travel shorter distances. A smaller exponent will indicate that distance is not as strong of a factor between stations and more people will ride the line from their home to the terminal CBD station. The most sensitive input that a planner has to choose is the tradeoff coefficient, β, between travel time and convenience. Faster travel times will occur with smaller β coefficients but will result in more inconvenience to the passengers. Within the results from the genetic algorithm, the inconvenience penalty has a higher magnitude that inputs a larger β coefficient. The inconvenience penalty is normalized by the β coefficient to compare solutions that use different β coefficients. Figure 7 shows the nonlinear relationship between travel time and inconvenience with respect to the β coefficient. The greatest improvements in travel time are when β is small, in the 0.05-0.40 range.

The existing schedule on the Metra Union Pacific North Line is denoted by the star in Figure 7. While not on the proposed optimal frontier, the current schedule does perform reasonably well. The current practices of Metra suggest that the β coefficient should be in the range of 0.40 - 0.55.

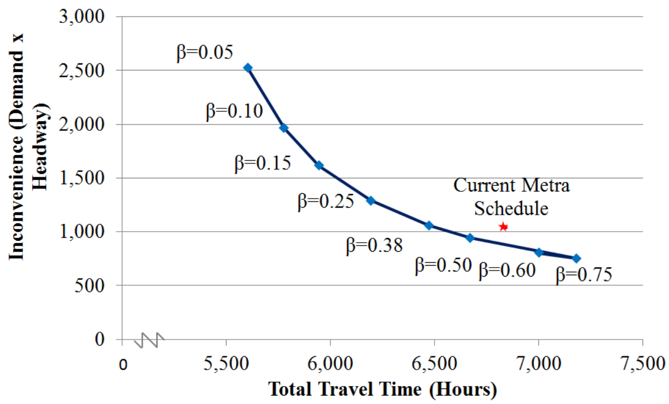

Figure 7: Sensitivity analysis with regard to the tradeoff coefficient β

Limitations

Trains are assumed to be stopped at each station for the same stop delay independent of the number of people who board or alight at the station. In reality, the rush hour trains experience longer dwells at stations for a greater number of passengers to board and alight the train. The off-peak trains actually have a lower average stop delay at each station than the rush hour trains. Additionally, all trains are assumed to originate in Kenosha and terminate in Chicago. Under the current operating plan, most inbound trains originate in Waukegan, with 9 starting in Kenosha, 5 in Winnetka, and 1 in Highland Park.

The proposed genetic algorithm is not guaranteed to find the optimal solution. The algorithm only searches the solution space for potential improved solutions. This scheduling problem is susceptible to local minimums. The algorithm can converge to a local minimum and not move towards the global minimum. This can be mitigated to some extent. Setting a higher iteration limit will allow for a higher probability of the mutation process finding a schedule or part of a schedule that is better than the current minimum. This type of search is more akin to a random search. Another option is to utilize a larger sample size which will explore more solutions. With a larger sample size, the algorithm will iterate more before converging to one value. However, a larger sample size will increase the computation time necessary to find a solution. The best way to mitigate this problem would be to run the algorithm in parallel with itself. Once values of each sub-population converge, they are combined into a master population for further analysis.

Lastly, there could be other operational or political reasons why a train stops at any given station that are beyond the scope of this research.

FUTURE WORK

There are a number of areas where the search algorithm could be further refined. In an effort to better incorporate train scheduling into how a set of stopping patterns is scored, a signal wake model could be incorporated into the heuristic search. This would ensure that any minimum stopping pattern developed by the search would be feasible to implement as a schedule for different existing and proposed infrastructure configurations. In addition to the search algorithm alterations, a scheduling program could be developed that would determine the ideal order of stopping patterns for a schedule. In this analysis, the stopping patterns would be ordered so that they accommodate headways as uniform as possible at each station while minimizing interference between trains.

The travel times between any origin-destination pair in Appendix C are based off an average stop delay from the Metra schedule. In reality, the stop delay should be related to train acceleration and braking rates as well as top speed. If subsequent stops are skipped, then the train will reach a higher maximum speed as well as a higher average speed. Using an average stop delay leads to conservative travel times in the analysis presented. The travel times can be updated to reflect the actual accelerating and braking characteristics using a train performance calculator (TPC).

The demand split between train services for each O-D pair was determined by the relative time savings between train services. This split between trains could also be modeled by a train choice logit model that incorporates the demand elasticities with respect to in-train-travel-time, waiting time at stations, and transfer time (if modeled). Incorporating a choice model will more accurately describe how transit riders in a particular region distinguish between train services.

In a further refinement, the heuristic search could be directed to modify existing schedules only to a certain degree instead of developing brand new schedules. In this process, an existing schedule would be fed into the algorithm along with constraints on how many additional stops could be added or skipped. Using this method may ease the difficulty of implementing a more efficient schedule over time by mitigating radical changes that may be initially unpopular with passengers.

Another possibility of decreasing travel times for passengers is to utilize a hybrid-local-express service where trains will stop at 5-10 stops in the suburbs and run express to the central business district. The Metra line from Chicago to Naperville uses this type of stopping pattern. This type of service is usually determined analytically to minimize passing conflicts. If there are significant numbers of passing conflicts then an additional track is required. This type of stopping pattern can achieve higher average speeds than a skip-stop service. Future work could compare a hybrid-local-express and a skip-stop service in terms of convenience and travel time

CONCLUSION

Skip-stop schedules can save a significant amount of passenger travel time by reducing the number of stops at stations with lower demand. Using a case study commuter rail line, two approaches to creating a skip stop schedule were explored. The mixed integer program presented is feasible for small problems, but as the number of stations and trains increase the number of possible solutions, a heuristic search may be a better method. The Metra Union Pacific North line morning inbound weekday scenario, with 27 stations and 14 trains, has a current passenger travel time of 6,845 passenger-hours and frequency penalty of 1,036 passenger-hours. The genetic algorithm presented was able to develop a schedule with 6,194 passenger hours of travel time and 1,289 passenger hours of frequency penalty, a 9.5% decrease and a 24.4% increase respectively. Passenger sensitivity to train frequency, expressed here as β, effects how aggressively skip-stop service can be implemented. Running a sensitivity analysis that compares the passenger travel time with frequency penalty for different values of β , it appears that Metra is running a close to optimal schedule for β = 0.40-0.55. Although the methods presented here were developed using a case study heavy rail commuter line, similar methods could conceivably be applied to other types of urban and interurban passenger rail transportation.

REFERENCES

[1] Lawrence D. Bodin, Bruce L. Goldstein, Allan D. Schuster, "A Model for the Blocking of Trains," *Transportation Research Part B,* vol. 14B, pp. 115-120, 1980.

[2] N.S.A. Ghoneim, S.C. Wirasinghe, "Optimum Zone Structure During Peak Periods for Existing Urban Rail Lines," *Transportation Research Part B,* vol. 20B, no. 1, pp. 7-18, 1984.

[3] A. E. Haghani, "Formulation and Solution of a Combined Train Routing and Makeup, and Empty Car Distribution Model," *Transportation Research Part B,* vol. 23B, no. 6, pp. 433-452, 1989.

[4] M.R. Bussieck, M.E. Lübbecke, T. Winter, U.T. Zimmermann, "Discrete Optimization in Rail Transport," 1998.

[5] M. F. Gorman, "Santa Fe Railway Uses and Operating-Plan Model to Improve its Service Design," in *Institute for Operations Research and the Management Sciences*, 1998.

[6] J.-C. Jong, C.-S. Suen and S. J. Chang, "A Decision Support System to Optimize Railway Stopping Patterns: Application To The Taiwan High Speed Rail," in *Transportation Research Board 91st Annual Meeting*, Washington D.C., 2011.

[7] Regional Transportation Authority, "Regional Transit Asset Management System," Chicago, 2006.

[8] Union Pacific Railroad, Chicago Area Timetable #4, 2009.

[9] METRA, "Union Pacific North Schedule," Chicago, 2011.

[10] M. Obitko, "Introduction to Genetic Algorithms," Dresden, 1998.

[11] D. McFadden, "The Measurement of Urban Travel Demand," *Journal of Public Economics,* pp. 303-328, 1974.

[12] Google Earth, "Chicago Northern Suburbs," 42°08'-08" 87°45'30", 2010.

[13] METRA, "Ridership Reports - System Facts," Chicago, 2010.

[14] C.-H. Y. C.-C. S. Yu-Hern Chang, "A Multiobjective Model for Passenger Train Services Planning: Applications to Taiwan's High-Speed Rail Line," *Transportation Research Part B,* vol. 34, pp. 91-106, 1999.

APPENDIX A

METRA UP NORTH LINE BASELINE 2006 RIDERSHIP

Station	Inbound				Outbound			
	Boardings		Alightings		Boardings		Alightings	
	#	%	#	%	#	%	#	%
Kenosha	317	3.1%		0.0%		0.0%	20	1.0%
Winthrop Harbor	77	0.8%		0.0%		0.0%	4	0.2%
Zion	130	1.3%		0.0%	1	0.0%	9	0.4%
Waukegan	630	6.2%	27	0.3%	2	0.1%	97	4.6%
North Chicago	68	0.7%	10	0.1%	2	0.1%	68	3.2%
Great Lakes	53	0.5%	20	0.2%	2	0.1%	109	5.2%
Lake Bluff	246	2.4%	20	0.2%	4	0.2%	160	7.6%
Lake Forest	283	2.8%	32	0.3%	7	0.3%	237	11.3%
Fort Sheridan	199	2.0%	10	0.1%	4	0.2%	36	1.7%
Highwood	98	1.0%	23	0.2%	32	1.5%	32	1.5%
Highland Park	580	5.7%	83	0.8%	17	0.8%	216	10.3%
Ravinia	219	2.1%	19	0.2%	8	0.4%	25	1.2%
Braeside	187	1.8%	21	0.2%	2	0.1%	96	4.6%
Glencoe	496	4.9%	44	0.4%	8	0.4%	93	4.4%
Hubbard Woods	255	2.5%	11	0.1%	8	0.4%	32	1.5%
Winnetka	395	3.9%	27	0.3%	14	0.7%	74	3.5%
Indian Hill	193	1.9%	115	1.1%	2	0.1%	44	2.1%
Kenilworth	306	3.0%	6	0.1%	20	1.0%	21	1.0%
Wilmette	1,126	11.0%	27	0.3%	21	1.0%	103	4.9%
Central St.	1,001	9.8%	35	0.3%	19	0.9%	54	2.6%
Davis St.	734	7.2%	237	2.3%	109	5.2%	444	21.2%
Main St.	618	6.1%	14	0.1%	99	4.7%	32	1.5%
Rogers Park	814	8.0%	17	0.2%	144	6.9%	19	0.9%
Ravenswood	1,064	10.4%	67	0.7%	540	25.7%	54	2.6%
Clybourn	109	1.1%	165	1.6%	379	18.1%	20	1.0%
Ogilvie		0.0%	9,168	89.9%	655	31.2%		0.0%

APPENDIX B

METRA UP NORTH LINE PROPOSED SCHEDULE

Train #	Train 1	Train 2	Train 3	Train 4	Train 5	Train 6	Train 7	Train 8	Train 9	Train 10	Train 11	Train 12	Train 13	Train 14
AM/PM	AM	AM	AM	AM	AM	AM	AM	AM	AM	AM	AM	AM	AM	AM
Kenosha	6:00	6:05	6:11	6:22	6:31	6:38	6:45	6:53	7:00	7:07	7:14	7:21	7:28	7:36
Winthrop Harbor	6:12	6:17	-	-	6:43	6:50	6:57	7:05	7:12	7:19	-	7:33	-	7:48
Zion	6:16	6:21	6:26	-	6:47	6:54	7:01	7:09	7:16	7:23	7:29	7:37	-	7:52
Waukegan	6:25	6:30	6:35	-	6:56	7:03	7:10	7:18	7:25	7:32	7:38	7:46	-	8:01
North Chicago	-	-	6:40	6:48	7:01	-	-	-	-	7:37	-	-	7:54	8:06
Great Lakes	-	-	6:45	6:53	-	7:12	-	-	-	-	7:46	-	7:59	-
Lake Bluff	-	-	6:49	-	7:09	7:16	-	7:29	7:36	7:45	7:50	-	8:03	8:14
Lake Forest	-	6:42	6:52	6:59	7:12	-	7:23	7:32	-	7:48	7:53	7:59	-	8:17
Fort Sheridan	6:40	-	-	-	7:16	7:21	7:27	-	7:42	7:52	-	8:03	8:08	-
Highwood	6:42	6:47	-	7:03	-	7:23	-	-	-	-	7:58	-	8:10	-
Highland Park	-	6:50	6:58	-	7:20	7:26	7:31	7:39	7:46	7:55	8:01	8:06	-	8:23
Ravinia	6:46	6:53	-	7:08	-	-	7:34	7:42	7:49	-	-	-	8:15	-
Ravinia Park	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Braeside	6:47	6:54	-	7:09	-	-	7:35	-	-	-	-	-	8:15	8:25
Glencoe	6:50	6:56	-	-	7:23	-	7:37	7:44	7:51	-	8:05	-	8:18	8:28
Hubbard Woods	-	6:59	7:03	7:13	-	7:32	-	-	-	-	-	8:12	8:21	-
Winnetka	6:54	-	7:06	7:16	7:28	-	-	-	-	8:02	-	-	8:24	8:32
Indian Hill	6:56	-	7:08	7:18	-	-	-	-	7:56	-	-	8:15	8:26	-
Kenilworth	-	-	-	7:20	-	-	-	7:49	7:58	8:05	8:10	8:17	-	-
Wilmette	6:59	7:04	7:11	-	7:31	7:36	-	7:51	-	-	8:12	8:19	8:29	8:36
Evanston Central Street	-	-	7:14	7:24	7:34	-	7:45	7:54	-	8:08	8:15	-	8:32	8:39
Evanston Davis Street	-	7:09	7:17	7:27	-	-	7:48	7:57	-	8:11	8:18	8:24	8:35	-
Evanston Main Street	-	-	-	7:30	-	7:43	7:51	-	8:05	8:14	-	8:27	-	8:43
Rogers Park	7:07	7:13	-	7:33	7:43	7:46	7:54	-	8:08	-	8:23	-	-	-
Ravenswood	7:11	-	7:24	7:37	-	7:50	-	-	8:12	-	-	8:32	8:42	-
Clybourn	7:16	-	-	7:42	-	7:55	-	8:08	-	8:23	-	-	-	-
Ogilvie Transportation Center	7:24	7:27	7:36	7:50	7:54	8:03	8:08	8:16	8:23	8:31	8:37	8:44	8:54	8:59

APPENDIX C

O-D TRAVEL TIMES (MINUTES)

	Origin																										
Destination	Kenosha	Winthrop Harbor	Zion	Waukegan	North Chicago	Great Lakes	Lake Bluff	Lake Forest	Fort Sheridan	Highwood	Highland Park	Ravinia	Ravinia Park	Braeside	Glencoe	Hubbard Woods	Winnetka	Indian Hill	Kenilworth	Wilmette	Evanston Central Street	Evanston Davis Street	Evanston Main Street	Rogers Park	Ravenswood	Clybourn	Ogilvie Transportation
Kenosha	-	12.0	14.6	22.1	25.7	29.3	31.8	33.4	36.0	36.6	38.1	39.7	39.3	38.8	40.4	42.0	43.5	44.1	44.7	45.2	46.8	48.4	50.0	51.5	54.1	57.7	64.2
Winthrop Harbor	8.0	-	4.0	11.6	15.1	18.7	21.3	22.8	25.4	26.0	27.6	29.1	28.7	28.3	29.8	31.4	33.0	33.5	34.1	34.7	36.2	37.8	39.4	41.0	43.5	47.1	53.7
Zion	9.6	3.0	-	9.0	12.6	16.1	18.7	20.3	22.8	23.4	25.0	26.6	26.1	25.7	27.3	28.8	30.4	31.0	31.5	32.1	33.7	35.2	36.8	38.4	41.0	44.5	51.1
Waukegan	17.1	10.6	9.0	-	5.0	8.6	11.1	12.7	15.3	15.8	17.4	19.0	18.6	18.1	19.7	21.3	22.8	23.4	24.0	24.5	26.1	27.7	29.2	30.8	33.4	37.0	43.5
North Chicago	18.7	12.1	10.6	3.0	-	5.0	7.6	9.1	11.7	12.3	13.8	15.4	15.0	14.6	16.1	17.7	19.3	19.8	20.4	21.0	22.5	24.1	25.7	27.2	29.8	33.4	40.0
Great Lakes	21.3	14.7	13.1	5.6	4.0	-	4.0	5.6	8.1	8.7	10.3	11.8	11.4	11.0	12.6	14.1	15.7	16.3	16.8	17.4	19.0	20.5	22.1	23.7	26.2	29.8	36.4
Lake Bluff	23.8	17.3	15.7	8.1	6.6	4.0	-	3.0	5.6	6.1	7.7	9.3	8.8	8.4	10.0	11.6	13.1	13.7	14.3	14.8	16.4	18.0	19.5	21.1	23.7	27.2	33.8
Lake Forest	25.4	18.8	17.3	9.7	8.1	5.6	3.0	-	4.0	4.6	6.1	7.7	7.3	6.8	8.4	10.0	11.6	12.1	12.7	13.3	14.8	16.4	18.0	19.5	22.1	25.7	32.2
Fort Sheridan	28.0	21.4	19.8	12.3	10.7	8.1	5.6	4.0	-	2.0	3.6	5.1	4.7	4.3	5.8	7.4	9.0	9.6	10.1	10.7	12.3	13.8	15.4	17.0	19.5	23.1	29.7
Highwood	29.6	23.0	21.4	13.8	12.3	9.7	7.1	5.6	3.0	-	3.0	4.6	4.1	3.7	5.3	6.8	8.4	9.0	9.6	10.1	11.7	13.3	14.8	16.4	19.0	22.5	29.1
Highland Park	31.1	24.6	23.0	15.4	13.8	11.3	8.7	7.1	4.6	3.0	-	3.0	2.6	2.1	3.7	5.3	6.8	7.4	8.0	8.6	10.1	11.7	13.3	14.8	17.4	21.0	27.5
Ravinia	32.7	26.1	24.6	17.0	15.4	12.8	10.3	8.7	6.1	4.6	3.0	-	1.0	0.6	2.1	3.7	5.3	5.8	6.4	7.0	8.6	10.1	11.7	13.3	15.8	19.4	26.0
Ravinia Park	32.3	25.7	24.1	16.6	15.0	12.4	9.8	8.3	5.7	4.1	2.6	1.0	-	1.0	2.6	4.1	5.7	6.3	6.8	7.4	9.0	10.6	12.1	13.7	16.3	19.8	26.4
Braeside	31.8	25.3	23.7	16.1	14.6	12.0	9.4	7.8	5.3	3.7	2.1	0.6	1.0	-	3.0	4.6	6.1	6.7	7.3	7.8	9.4	11.0	12.6	14.1	16.7	20.3	26.8
Glencoe	33.4	26.8	25.3	17.7	16.1	13.6	11.0	9.4	6.8	5.3	3.7	2.1	2.6	3.0	-	3.0	4.6	5.1	5.7	6.3	7.8	9.4	11.0	12.6	15.1	18.7	25.3
Hubbard Woods	35.0	28.4	26.8	19.3	17.7	15.1	12.6	11.0	8.4	6.8	5.3	3.7	4.1	4.6	3.0	-	3.0	3.6	4.1	4.7	6.3	7.8	9.4	11.0	13.6	17.1	23.7
Winnetka	36.5	30.0	28.4	20.8	19.3	16.7	14.1	12.6	10.0	8.4	6.8	5.3	5.7	6.1	4.6	3.0	-	2.0	2.6	3.1	4.7	6.3	7.8	9.4	12.0	15.6	22.1
Indian Hill	37.1	30.5	29.0	21.4	19.8	17.3	14.7	13.1	10.6	9.0	7.4	5.8	6.3	6.7	5.1	3.6	2.0	-	2.0	2.6	4.1	5.7	7.3	8.8	11.4	15.0	21.6
Kenilworth	37.7	31.1	29.5	22.0	20.4	17.8	15.3	13.7	11.1	9.6	8.0	6.4	6.8	7.3	5.7	4.1	2.6	2.0	-	2.0	3.6	5.1	6.7	8.3	10.8	14.4	21.0
Wilmette	38.2	31.7	30.1	22.5	21.0	18.4	15.8	14.3	11.7	10.1	8.6	7.0	7.4	7.8	6.3	4.7	3.1	2.6	2.0	-	3.0	4.6	6.1	7.7	10.3	13.8	20.4
Evanston Central Street	39.8	33.2	31.7	24.1	22.5	20.0	17.4	15.8	13.3	11.7	10.1	8.6	9.0	9.4	7.8	6.3	4.7	4.1	3.6	3.0	-	3.0	4.6	6.1	8.7	12.3	18.8
Evanston Davis Street	41.4	34.8	33.2	25.7	24.1	21.5	19.0	17.4	14.8	13.3	11.7	10.1	10.6	11.0	9.4	7.8	6.3	5.7	5.1	4.6	3.0	-	3.0	4.6	7.1	10.7	17.3
Evanston Main Street	42.0	35.4	33.8	26.2	24.7	22.1	19.5	18.0	15.4	13.8	12.3	10.7	11.1	11.6	10.0	8.4	6.8	6.3	5.7	5.1	3.6	2.0	-	3.0	5.6	9.1	15.7
Rogers Park	43.5	37.0	35.4	27.8	26.2	23.7	21.1	19.5	17.0	15.4	13.8	12.3	12.7	13.1	11.6	10.0	8.4	7.8	7.3	6.7	5.1	3.6	3.0	-	4.0	7.6	14.1
Ravenswood	47.1	40.5	39.0	31.4	29.8	27.2	24.7	23.1	20.5	19.0	17.4	15.8	16.3	16.7	15.1	13.6	12.0	11.4	10.8	10.3	8.7	7.1	6.6	5.0	-	5.0	11.6
Clybourn	51.7	45.1	43.5	36.0	34.4	31.8	29.2	27.7	25.1	23.5	22.0	20.4	20.8	21.3	19.7	18.1	16.6	16.0	15.4	14.8	13.3	11.7	11.1	9.6	6.0	-	8.0
Ogilvie Transportation Center	59.2	52.7	51.1	43.5	42.0	39.4	36.8	35.2	32.7	31.1	29.5	28.0	28.4	28.8	27.3	25.7	24.1	23.6	23.0	22.4	20.8	19.3	18.7	17.1	13.6	9.0	-