Vrije Universiteit Amsterdam

Bachelor Thesis

Public transport transfer time minimization in Luxembourg

Author: Thomas Nieuwenhuis (2641933)

1st supervisor:	Wan Fokkink
daily supervisor:	Wan Fokkink
2nd reader:	Femke van Raamsdonk

A thesis submitted in fulfillment of the requirements for the VU Bachelor of Science degree in Computer Science

July 5, 2021

Abstract

This thesis focuses on minimizing the average transfer time for the public transport network (PTN) in Luxembourg. In order to solve this NP-hard problem, timetable schedules are only allowed to be shifted backwards or forwards by several minutes. Three metaheuristic algorithms, namely two hill-climbing algorithms and one genetic algorithm are used to solve the optimization problem at hand. Every PTN is evaluated by simulating all the journeys between any two stations from Monday to Friday between 08:00 and 20:00 with Dijkstra's all-pairs shortest paths algorithm. In the end, a 19.3% decrease in the average transfer time in a reduced PTN and a 5.7% decrease in the complete PTN was achieved for journeys with at least one transfer. While the main goal of this study is to decrease the average transfer time, an increase of the general passenger satisfaction and convincing people to make more use of public transport is hoped to be a by-product of it.

Keywords: Transfer Time Minimization, Luxembourg, Public Transport, Hill-Climbing, Genetic Algorithm

1 Introduction

With the green party being one of the main forces in the Luxembourgish government, the question arises why Luxembourg is still one of the worst European countries when it comes to adapting to climate change. As can be seen from Table 1, Luxembourg's greenhouse gas emissions are nearly twice as big while its gross electricity generation from renewable sources is not even half than any of its neighbouring countries. Even though decisions such as making public transport free and reinstating the tramway are an effort to battle climate change, decisions such as subsidizing the acquisition of private electric cars for up to 8000€ and not increasing petrol taxes seem hardly justified as they do not force people to switch from their private cars over to public transport.

The increase of petrol taxes is a delicate topic in Luxembourg, on the one hand because of fuel tourism being an important economic factor and on the other hand simply because people prefer disregarding climate change for the luxury of being able to keep on using their cars. Ironically, Luxembourg has by far the biggest GDP in the EU while its petrol prices are amongst the lowest. Moreover, people might be more inclined to buy electric cars if petrol prices rise, which would only shift the problem because it would create a renewable energy bottleneck.

While private electric cars are being promoted by the media, the statistics are not very convincing. Not only is the production and recycling of electric vehicle batteries very derogatory to the environment, they also rely on scarcely available renewable energy to have an environmentally favourable impact compared to combustion engines. In addition to this, these batteries can take up to several hours to recharge and restrict the driver to a smaller driving distance than combustion engines. This does not mean that we should be satisfied with driving combustion engines as it merely indicates that neither of both are currently the perfect solution and that we should try to minimize the usage of both.

Even though electric cars emit much less CO2 during their use phase [1], their Life Cycle Analysis (LCA) shows that both their production phase and their end of life phase are a lot more detrimental to the environment than people realize. A single battery change can lead to an electric car losing all of its CO2 gains compared to an internal combustion engine [2]. If we only base the LCA on CO2 emissions, it will take an electric car 76.6km [2] in Europe to be environmentally friendlier than a gasoline car and 109.4km [2] to overtake a car running on diesel. However, if we also include other factors, such as acidification (+50%), human toxicity (+387%), particulate matter (+128%), photochemical ozone formation (+24%) and resource depletion (+32%) [3], it becomes apparent that private electric vehicles do not seem to be any environmentally friendlier than internal combustion vehicles. In none of these categories are electric cars able to break even with internal combustion engine cars considering the use of a European average electricity grid mix [3]. With this in mind, it is clear that the goal of subsidizing electrical cars for up to 8000€ helps in improving Luxembourg's greenhouse gas statistics because neither their production nor their destruction happens in Luxembourg, hence the much bigger and more toxic emissions during these processes will not be counted in within any statistics.

Another factor is that, even though Luxembourg's public transport is amongst the most reliable ones in Europe, it is still too unreliable for people to switch away from their cars. The CFL, Luxembourg's government-owned railway company reported a 90% punctuality in 2019 [4] which shows that on average every tenth train trip is delayed.

Country	LUX	BEL	FRA	GER	NED	EU-27
Gross Domestic Product (thousand Euro per capita) [5]	100.2	40.3	35.3	40.5	45.0	30.3
Greenhouse gas emissions (tonnes of CO2 equivalent per capita) [5]	20.3	10.8	6.9	10.7	11.6	8.7
Gross electricity generation from renewable sources (share of gross electricity consumption (%)) [5]	9.1	21.2	18.9	38.0	32.2	15.1
Energy from renewable sources used in transport (share of gross final energy consumption (%)) [5]	6.5	9.0	6.6	7.9	8.3	9.6
Passenger cars (number per thousand inhabitants) [5]	676	511	478	567	494	529
Diesel prices (in Euro) [6]	1.13	1.42	1.38	1.31	1.38	1.29
Super95 prices (in Euro) [6]	1.25	1.41	1.51	1.51	1.73	1.45

Table 1: Western EU countries statistical analysis

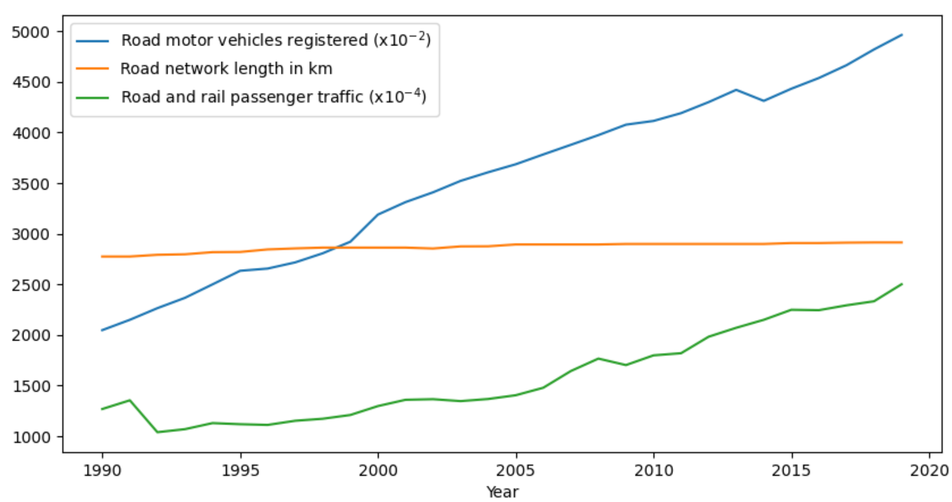

Figure 1: Traffic analysis [7], [8], [9]

Noticeable is also that Luxembourg is a very centered country with most activities happening in Luxembourg City while the north of the country is a quiet place. As an effect, public transport is very advanced around Luxembourg City while many public transport routes in the north are only being serviced maximum once an hour. As a consequence, many villages in the north are barely connected via public transport and people are forced to use their cars if they do not want to wait forever to finally catch a bus or a train.

Finally, traffic jams play a major role in greenhouse gas emissions in Luxembourg. As can be drawn from Figure 1, the number of cars registered in Luxembourg has increased faster than the number of people making use of public transport over the past 30 years. Furthermore, the road network has barely been extended, which has made the traffic congestion increase dramatically during recent years.

In the short run, we will have to make more use of public transport. Not only does the public transport operate anyway, it keeps us from riding alone in our cars and as a direct consequence from creating even bigger traffic jams. Nonetheless, one downside of public transport is that one often has to transfer along the journey, which adds aggravating travel time. Most public transport optimizations, such as duplicating rail tracks or increasing the vehicle fleet, trying to reduce afore mentioned travel time in order to increase customer satisfaction require serious planning or financial investment.

In this thesis, a solution that takes little effort and no financial backing is proposed. The goal is to computationally optimize Luxembourg's public transport time schedules such that the transfer times between any two stations are minimized. This can simply be achieved by independently shifting the time schedules of individual routes forwards or backwards by several minutes and observing the change in the average transfer time of the entire public transport network.

The data set provided by the Luxembourgish Ministry of Mobility will first be sanitized before it can be used in the experiment. Several constraints, such as not including passenger demand, are proposed in order to make the goal of this thesis achievable. From the four metaheuristic algorithms proposed in the literature study, the hill-climbing technique and the genetic algorithm approach are chosen to be used in the experiment. With only metaheuristic algorithms available, the main issue faced is to create an objective function with a reasonable computation time. However, reducing the network size by filtering out the most visited stations ends up to be the only viable option. After a brief description of the programmed framework, the implementations of the chosen algorithms will be explained before being executed on the DAS-5.

In the end, the results are very promising with transfer time reductions of up to 19.3%. Both the optimized hill-climbing algorithm and the genetic algorithm perform equally well, unlike the complete hill-climbing algorithm which takes too much time to compute. Some ethical aspects will be analysed before wrapping up the thesis and having a look at possible future work.

2 Literature Study

Public transport network (PTN) optimizations come in many different shapes and sizes and most papers related to this topic can be split into three categories. Some papers are related to PTN planning where ideas are being proposed which can be made use of during the planning of a PTN, transit route or time-schedule set-up. The input of such kinds of studies are most often the fixed number of headways to be used or a fixed vehicle fleet. Other papers treat optimization strategies such as holding [10], zone scheduling [10], short-turning [11] or deadheading [11] to enhance single routes or PTNs already in place. Finally, the last category of papers deals with real-time optimizations such as synchronizing public transport transit by using inter-vehicle communication [12] or making use of operational tactics such as holding, skip-stops or short-turning [13].

The topic of this thesis belongs to the second category as an existing PTN without any real-time enhancements is being treated. The optimization focus of this study will be on the transfer time minimization which has as objective the minimization of the average transfer time within the PTN. This optimization however shall not be confused with transfer minimization which focuses solely on the minimization of the number of transfers [14], disregarding the transfer times.

The most common objectives for improving a PTN are customer satisfaction, energy consumption reduction [15] or financial reasons. While some studies try to optimize a single objective, other studies try to optimize a set of correlated objectives such as timetable synchronization and user demand [16] or transfer times and operational costs [17]. Where possible, papers take passenger demand [15][18] into account in order to have a bigger influence in customer satisfaction.

A PTN optimization problem with a gigantic search space such as this one has been proven to be NP-hard [18][19] and makes it therefore nearly impossible to find the best solution. It thus requires a metaheuristic algorithm to find an approximated optimal solution to the optimization problem. The four metaheuristic algorithms found in the literature study and related to studies such as the one treated in this thesis were simulated annealing, sequential hill-climbing, tabu search and genetic algorithms.

A hill-climbing algorithm [20] starts at a random point in the search space. It then repeatedly evaluates all of the neighbour solutions and switches to the neighbour solution with the best objective function value. Once none of the neighbour solutions have a better value, the algorithm knows that it has found a local minimum and has to start a new search from a newly computed random point. Both simulated annealing and tabu search are very similar to hill-climbing. Simulated annealing [21] includes a probability factor which, when triggered, chooses a random neighbour solution and not the best one. This guarantees that the search does not get stuck in a single local minimum. Tabu search [22] on the other hand keeps track of a so-called tabu list. After every iteration, the best found solution that is not contained in the tabu list will be added to the list and won't be able to be revisited within a chosen number of turns. This ensures that the algorithm is being discouraged of back-tracking to solutions that have already been visited. Finally, the last option is a genetic algorithm [23] which starts with a randomly initialized pool of chromosomes. It then repeatedly performs a crossover strategy on two chromosomes chosen from its current pool by its selection strategy. Each new chromosome is then mutated according to the mutation strategy and the iteration ends once the new pool of chromosomes has the same size as the previous pool.

Both simulated annealing and sequential hill-climbing only occur in papers that compare the effectiveness of metaheuristic algorithms in such an environment. Simulated annealing has been outperformed in solution quality and computation time by sequential hill-climbing [24], a genetic algorithm [25] and tabu search [26], while sequential hill-climbing has been proven to outperform a genetic algorithm in both of these categories [24]. Questionable however is that the latter study takes into account only two bus lines, which creates a search space far smaller than the one discussed in this thesis. With a tabu search, 11.1% improvement in waiting times for one evening in the greater Copenhagen metropolitan area [26] and 5.1% improvement in the weighted waiting time [27] in entire Denmark have been computed in two different studies. Finally, the metaheuristic algorithm used in most papers is a genetic algorithm which consists of selection, crossover and mutation. In terms of selection strategy, roulette wheel is being proposed twice [28], [29] and elitism once [28]. While the general consensus is that a 50% crossover probability seems like the best choice, the optimal mutation probability found, namely 10% [30] and 90% [28], and the preferred population size, 80 and 20 respectively, are quite far apart. An improvement of 13.1% of the average transfer time was recorded on the Broward County Transit data [30] and a 14.6% reduction in transfer waiting times on the Mashhad City bus network [28].

While searching for the minimum average transfer time it might happen that journeys will be taken into account where travellers will gain for example a single minute by making an extra transfer instead of waiting a bit longer at a transfer station. To most travellers however this makes little sense as it is less convenient to switch between two vehicles than to travel for a little bit longer. Even though transfer count minimization [31] and transfer inconvenience [32] have been well researched, no study has documented how much travel time passengers are willing to give up in order to save on an extra transfer.

All of these studies require for the PTN to be evaluated at some point, yet the research done within this field is very scarce. The only paper found [33] calculates the estimated average waiting time at stops based on the arrival time of the passenger and the departure time of the public transport vehicle at the stop. However, as we consider no delay in this study and the passengers to always be on time at their departure station, we do not have to estimate any waiting times but are able to calculate the waiting time on existing data.

Finally, we also need to keep track of important satisfaction criteria. The walking time between any two stations should for example not exceed 10 minutes [34]. In addition to this, an average transfer time of less than 6 minutes should be attained to satisfy at least 90% of the passengers [35].

3 Research Question

The objective of this study is to find out whether the average waiting time in the PTN can be decreased using AI techniques. As the literature study shows, only metaheuristic algorithms seem applicable when solving such an optimization problem. The objective function, used in each metaheuristic algorithm to evaluate each PTN, needs to be optimized to require as little computation time as possible. After some thorough research, the decision is taken to implement one complete and one adjusted hill-climbing algorithm as well as a genetic algorithm. With 3 different algorithms, using only a single algorithm which performs poorly and thereby ruining the experiment can be avoided. The exact decision process and the detailed implementation are described in experimental method.

With 2799 stations and 537 public transport routes, Luxembourg's PTN is not too big for the calculations to take forever, yet it still poses a certain challenge to the optimization problem faced. Luxembourg's PTN makes use of buses, trains, tramways and a funicular, and after having added multiple routes to the PTN over many years, an optimal synchronization between all of the transport vehicles within the PTN has been lost. By minimizing the average transfer time we hope to increase passenger satisfaction and with it the general usage of public transport.

4 Experimental Setup

4.1 Data Set

The data set containing all the information of the PTN is provided by the Luxembourgish Ministry of Mobility on the platform www.data.public.lu [36]. The data set enforces the General Transit Feed Specification (GTFS) [37] format which is a universally adopted PTN specification. In addition to this, a map of Luxembourg has been retrieved from OpenStreetMap [38] in order to calculate the walking times between two bus stations with the Open Source Routing Machine [39].

4.2 Data Sanitization

First, we can get rid of the agency, calendar_dates, frequencies, shapes and transfers text files as they either do not contribute to this project or are provided without content. We then get rid of all the trips that are not being driven every day during the weekdays. This ensures that we do not make use of trips which only ride partially during the week or only during weekends. Once this step is done, we can filter out the remaining route_IDs, the remaining stop_times related to the trips that are left and the remaining stops that are being visited during these trips. The next step is to rewrite the route_IDs, the trip_IDs and the stop_IDs so that they all start at 0 and are incremented continuously without any gaps, in order to be able to index them faster in arrays later on. Furthermore, the times are converted to integer numbers so that it will be easier to read them and to compare them in the main program. Finally, we create a routes file which for every stop time in the PTN contains the route_ID, the maximum headway of the route, the trip_ID, the station_ID, the departure time and the arrival time. A second file called stations keeps track of the station_IDs and the walking times between two stations. This guarantees that we only have the important data left and that this data can easily be read into a C program.

4.3 Constraints

This project naturally enforces many constraints as the number of ideas that can be taken into account is very extensive. The following constraints have been set to lighten the project and make it achievable within the set time constraint. Firstly, the passenger demand cannot be taken into account as it is impossible to track enough passengers besides the actual project work. Next, the assumption has to be made that public transport companies are not necessarily willing to buy new vehicles or hire new employees, hence the constraint that the number of buses, lines and travels for a company shall remain unchanged has to be enforced. Moreover, only journeys where the planned arrival time at the destination station lies between 8:00 and 20:00 will be taken into account because

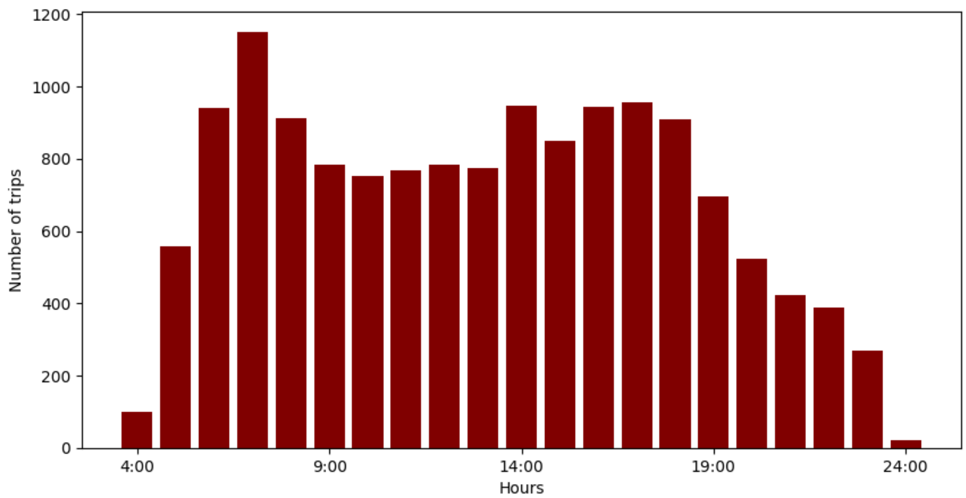

Figure 2: Trip starting times

All-Pairs Shortest Paths	Dijkstra	Floyd-Warshall	Bellman-Ford
Space Complexity	O(V)	O(V^2)	O(V)
Time Complexity	O(VE + V^2lg(V))	O(V^3)	O(V^2E)
V = 2779, E = 350629	1 062 749 996	24 461 775 139	2 707 852 016 989
V = 500, E = 180965	92 723 946	125 000 000	45 241 250 000

Table 2: Space & Time Complexity

some journeys that should arrive at 8:00 will have to start at 6:00 already and as one can see from Figure 2, most vehicles operate between 6:00 and 20:00. To limit the search space, each schedule is only allowed to be shifted between -20 minutes and +20 minutes of its original time. Even though this seems like a heavy restriction, with the PTN having 537 routes, the search space will still contain $8 * 10^{111}$ possible solutions. Finally, this thesis assumes that every public transport vehicle will always be on time and that travellers know the times and therefore do not have to wait when they get to their departure station. Assuming this was not the case would create a model closer to reality but at the same time increase the difficulty of this project.

5 Method

5.1 Objective Function

The objective function is the most important part of the algorithm as it allows to turn any PTN into a single value, in this case the average transfer time, and subsequently compare various PTNs to each other to find the PTN with the lowest value. It is however at the same time also the bottleneck of this project as a single evaluation requires approximately 23.5 hours of computing time with 2779 stations as well as 350629 stops and walks. In order to determine the average transfer time, all the journeys between any two stations where the planned arrival time lies between 8:00 and 20:00 are simulated. Dijkstra, Floyd-Warshall and Bellman-Ford, also known as the all-pairs shortest paths problem, are the 3 algorithms that can be used to solve the problem at hand. With the PTN having 2779 stations, it can be clearly seen from Table 2 that the optimal algorithm to be used in this scenario should be Dijkstra's algorithm. The algorithm starts at each station at every minute between 8:00 and 20:00 and uses Dijkstra's algorithm to find its way through the PTN to every other station in a backward manner. During the experiment a forward search was found to lead to problems because a person might have to wait at an intermediate station for such a long time that they could have also taken a later bus at the start station of their journey. This problem can be avoided by starting at the arrival station and searching backwards. Furthermore, as mentioned in the literature study, only walks between stations are included if they require less than 10 minutes of walking as travellers are not keen on walking much longer. Moreover, a 60 seconds mandatory transfer time was added, on the one hand to allow the passengers to switch from one vehicle to another and on the other hand to ensure that not all of the vehicles arrive at a station at the same time. Finally, if two different journeys between two stations can be established with the same travel time, the simulation will choose the journey with the least amount of transfers as this is a more comfortable choice for the traveller.

Dijkstra's all-pairs shortest paths algorithm is a custom implementation which is used to simulate every journey to each arrival station for every arrival time. During every iteration, the algorithm initializes an array for all the stations in which it keeps track of the current best journey from that station to the current arrival station. It then starts at the arrival station and calculates the latest departure time possible from every other station it is connected to based on the incoming vehicles and the walking times. Once this step is done, it searches for the station that holds the latest departure time and has not been visited yet and repeats both of these steps for every station in the PTN. Finally, the algorithm cycles through the array and accumulates the transfer times for the journeys with at least one transfer.

As mentioned before, the PTN evaluation is the bottleneck of this project. Options to reduce this gigantic computation time such as an alternative algorithm, reducing the PTN size or reducing the time frame for evaluating the PTN therefore had to be explored. The most important goal to keep in mind while searching for methods to decrease the computation time was that the search direction should remain the same. In other terms, when searching for the optimal PTN with an optimized objective function, it has to be ensured that the objective function of the original PTN also decreases.

5.1.1 Alternative Approach

The first idea that came up was to completely change the algorithm and to map every incoming vehicle at every station to every following outgoing vehicle visiting a different station. One would then be able to sum up the transfer time for each of these mappings and calculate the average transfer time. The algorithm's computation time of 353 milliseconds looked promising. However, as can be seen in Figure 3, such an algorithm ended up giving different values than the original algorithm and in addition did not have the same minimum indices. This idea therefore had to be disregarded and other algorithm improvements

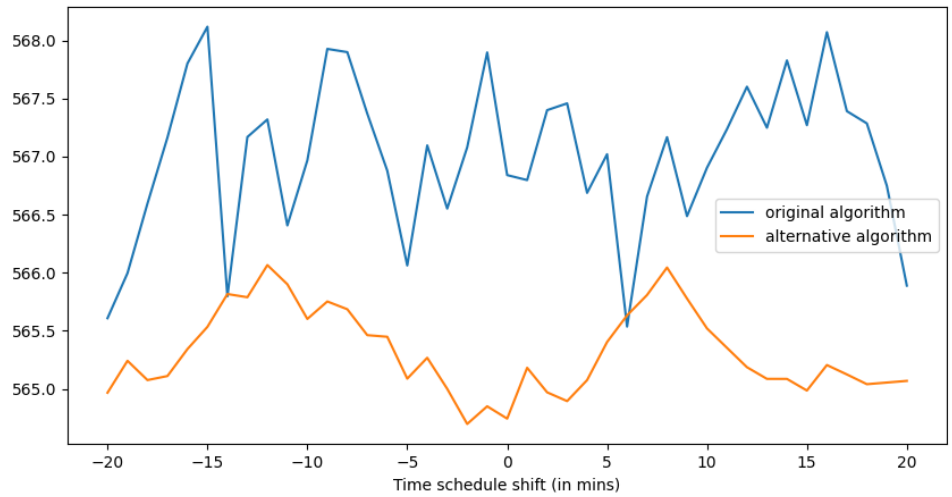

Figure 3: Faster Objective Function

needed to be considered.

5.1.2 Reduce PTN Size

The best way to decrease the computation time of Dijkstra's all-pair shortest paths algorithm is to reduce the number of vertices which in this case are the stations. However, one must also ensure that the biggest count of routes and stops possible shall remain within the PTN in order to not lose too much information. There are two possible ways to achieve this, either by filtering out the stations or by filtering out the routes with the most amount of vehicle stops. As can be derived from Figure 4, accumulating the stops of stations being visited the most leads to a quick increase in routes. In this case the PTN also includes more stops than if one was to retain the routes with the most stops. The number of routes increases heavily only for the top 500 stations, so if the PTN is limited to only these stations it will lead to retaining the highest number of routes (453, 84.4%) and stops (178049, 53.4%) within the PTN for the lowest amount of stations. Finally, Table 2 displays that this reduces the calculation time by an order of magnitude and Figures 7 and 8 shows that the scaled down PTN still represents the original PTN fairly well.

5.1.3 Reduce Time Frame

Reducing the time frame when the journeys are being simulated from 8:00 - 20:00 to a single hour or maybe a two or three hour frame could also increase the speed of the PTN evaluation by a factor of four. To check if this is a valid optimization, 5 routes in the scaled down PTN from section 5.1.2 were shifted backwards and forwards by 20 minutes, one minute at a time. For each of those 41 shifts of every route, the time frame is reduced to a one, two and a three hour frame instead of the regular 8:00 - 20:00. The goal is then to find out if there exists one time frame where the minimum objective value of the shifts of every route is the same as for the original time frame. However, as Figure 5 illustrates,

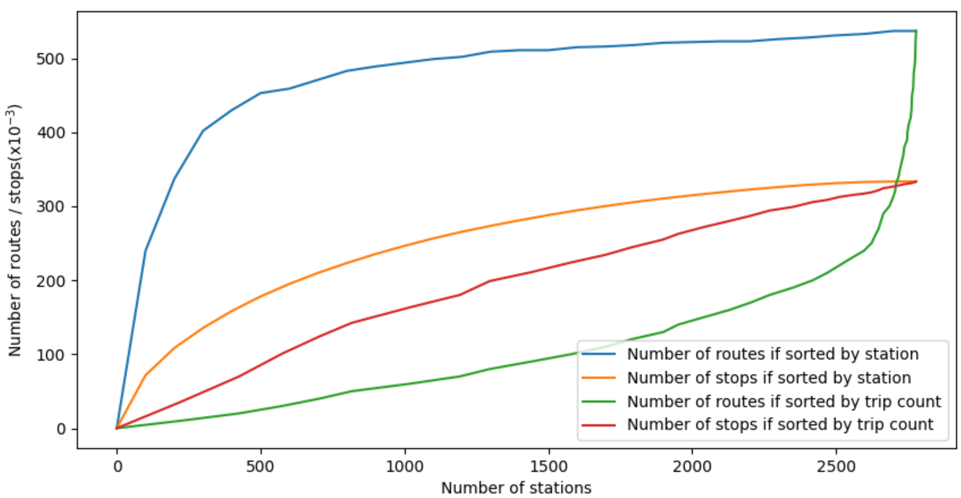

Figure 4: Number of routes and stops

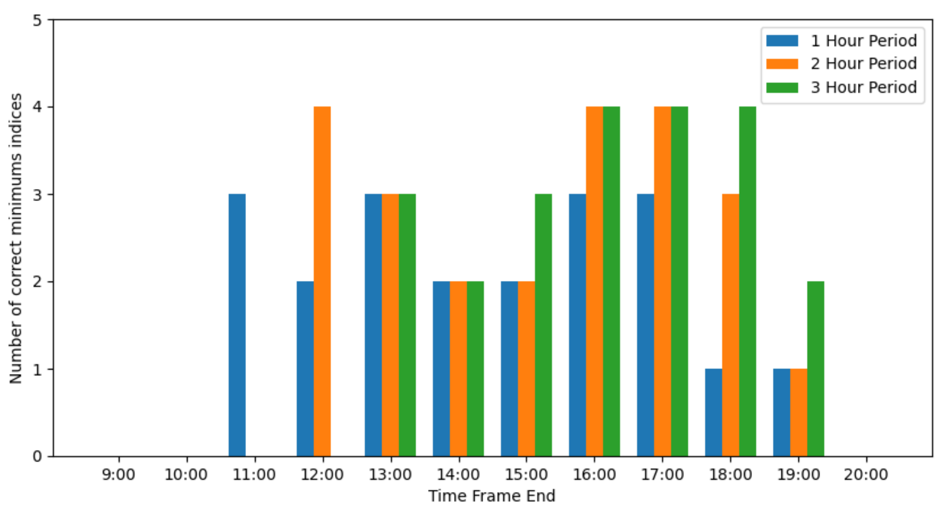

Figure 5: Number of correct minimums per period

there is not a single scaled down time period which managed to correctly predict all the minimum indices from these routes in the PTN. Moreover, there is no clear time frame that outperforms the other time frames, hence this optimization technique did not seem like a good choice.

5.2 Code Framework

All of the computations have been done on 16 nodes of the DAS-5 (The Distributed ASCI Supercomputer 5) [40]. Each node consists of dual 8-core CPUs and a memory capacity of 64 GB. In order to get all 256 cores to work concurrently on each algorithm, Open MPI [41] was used, which allows to specify one core as the master core. This master core then executes the algorithm and if it gets to the part where the newly discovered PTNs have to be evaluated by the objective function, the master core will make use of the other cores to multithread the computations.

Every implemented algorithm is based on a simple framework. First, the routes and the stations are read from the files created during the data sanitization. The program then initializes the MPI process's execution environment and gives each core an identifying number. The master core then takes over program control while the other cores wait for instructions to evaluate PTNs with the objective function. As long as the finish condition has not been reached, the master core will endlessly create new PTNs, evaluate them with the other cores and decide how to handle the previous PTNs depending on the algorithm.

The algorithms mpiHillClimbing.c, mpiOptimizedHillClimbing.c and mpiGenAlgo.c, written specifically for this experiment, are retrievable from GitHub [42]. This repository contains all the code written during this study, including the files used for data sanitization and for creating the graphs. The latter files were all written in Python for simplicity reasons while the algorithms used in the experiment had to be written in C to optimize efficiency.

5.3 Metaheuristic Algorithms

The first decision is to use a hill climbing technique to solve the problem at hand as it outperformed simulated annealing and a genetic algorithm as mentioned in the literature study. Furthermore, the algorithm keeps track of which states it has already visited which makes a tabu search algorithm redundant as there is no possibility to revisit older states. The only problem remaining is that with 453 routes, each one able to be shifted 1 minute forwards and backwards, 906 neighbour solutions need to be evaluated at every iteration which slows the algorithm down a lot. The second decision is therefore to also try out a genetic algorithm as this opens up the option to specify the size of the PTN pool and hence also the number of PTN evaluations being done during every iteration.

5.3.1 Hill Climbing

Even though a hill-climbing algorithm usually starts at a random point in the search space, the algorithm in this study starts with the original PTN. The time constraints of this study made it clear that only a single hill-climbing search is possible during the experiment. As the main goal of this study is to find an improved PTN, the decision to start from the original PTN was straightforward because the hill-climbing algorithm guarantees an immediate improvement. The algorithm first evaluates the initial PTN and adds it to the linked list where it stores all of the PTNs that have been visited. This list is sorted at all times with the PTN having the best value being the header node and each node in the list containing information about whether it has already been visited or not. In the endless loop the algorithm then chooses the best, not yet visited PTN, computes all of its neighbours and evaluates them. A neighbour solution in this case is a PTN where the time table of one of the routes has been shifted either forwards or backwards by a single minute. If a neighbour solution has a better value than the PTN it emerged from and is not yet in the linked list, it will be sorted into the list. The finish condition is met if the best PTN either improves by less than 0.01 seconds compared to the best PTN found in the previous iteration or if no improved PTNs have been found.

5.3.2 Hill Climbing (Optimization)

A trivial optimization to the previous hill climbing algorithm is reducing the amount of neighbour solutions being searched to 256 which is equal to the number of cores available. The 256 neighbour solutions are drawn randomly and it is made sure of that no solution is being drawn twice. While previously 906 neighbour solutions had to be evaluated, which implied that some cores had to evaluate 4 PTNs during a single iteration, each core now has to evaluate exactly one PTN, hence the computation time has been reduced by 75% from approximately one hour to 15 minutes.

5.3.3 Genetic Algorithm

In order to find the right parameters for the genetic algorithm the idea was to simply average the values from the papers found in the literature study. This means that the roulette wheel strategy, a crossover probability of 50%, a mutation probability of 50% and a population size of 50 PTNs should be used. However two adjustments had to be made as with 256 cores available from the DAS-5 it made more sense to use a population size of 256 PTNs. Additionally, a 50% mutation probability was questionable as this would likely prevent the population to converge to an optimal solution, therefore it was decreased to 10% as suggested in the paper of Cevallos and Zhao.

The algorithm starts with initializing a pool of 256 PTNs with randomly shifted schedules. It then enters the endless loop where it first evaluates all of the PTNs. Once this step is finished, the algorithm creates the roulette wheel by finding the highest objective value of all the PTNs and calculating the share on the roulette wheel of every PTN by subtracting its value from the highest objective value previously found. It then begins to create 256 new PTNs by electing two parent PTNs from the roulette wheel and merging the route schedules by choosing each route schedule from both parents with a 50-50 percent probability. Finally, before replacing the old PTN pool with the newly computed pool, the algorithm goes through every route of every PTN and randomly shifts the route's schedule if the mutation probability applies. The finish condition is reached if the value of the best PTN in the pool does not change during 10 iterations and the difference between the value of the best and the worst PTN is less than 1 second.

After the first 1400 minutes, it seemed like the algorithm did not converge to an optimal solution with a 10% mutation rate and it was therefore decreased to 5%. This same

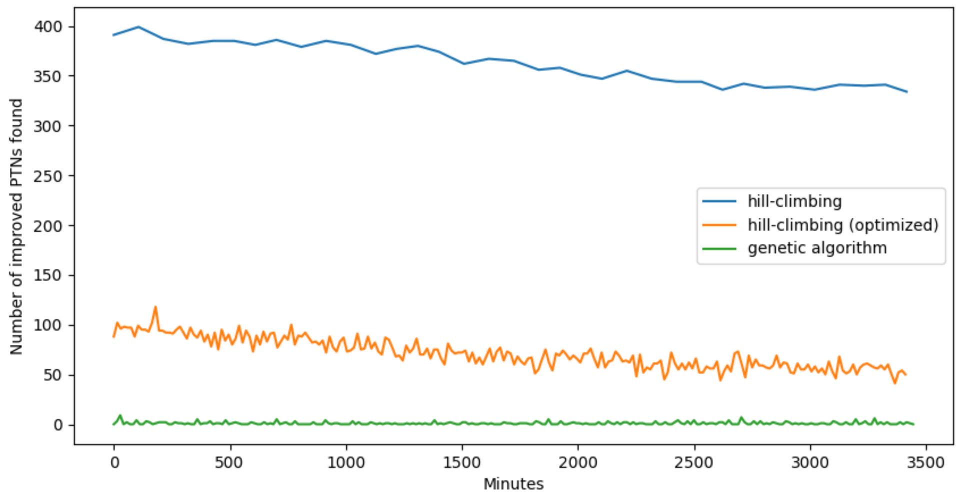

Figure 6: Number of better schedules found at each epoch

phenomenon happened again after another 1400 minutes at around 2800 minutes and the decision to further decrease the mutation rate to 1% had to be taken.

6 Results

Figure 6 illustrates for every iteration the number of PTNs that outperformed the best PTN of the previous iteration. The optimized hill-climbing algorithm for example found only 0.037 seconds of improvement within its last iteration even though it discovered 50 better PTNs. Considering the linear decrease of both hill-climbing algorithms, it is safe to say that a continuation of the search for the local minimum does not make sense as it results in the algorithm losing too much time.

It is noteworthy that the genetic algorithm does as well as the optimized hill-climbing algorithm and better than the original hill-climbing algorithm even though it does not discover any improved PTNs during many iterations. This implies that the number of improved PTNs found per iteration has no correlation with the strength of the algorithm.

Figure 7 displays the average transfer time of the best PTN for every iteration of every algorithm. We can derive from it that the optimized hill-climbing algorithm found the best PTN (457.5s, 19.3%) closely followed by the genetic algorithm (458.9s, 19.0%). The original hill-climbing algorithm's best found PTN (501.9s, 11.5%) on the other hand is far off from the other two algorithms which is mainly due to the fact that it takes approximately 60 minutes for a single iteration while the other two algorithms only need 15 minutes for one iteration.

The effect of decreasing the mutation rate becomes visible especially when decreasing it to 1%. While the genetic algorithm does not perform greatly at first compared to the other two algorithms, the average value of all the PTNs and the value of the best PTN improve greatly once the mutation rate is fixed at 1%. Moreover, at that point, the variance decreases and the average becomes closer to the minimum.

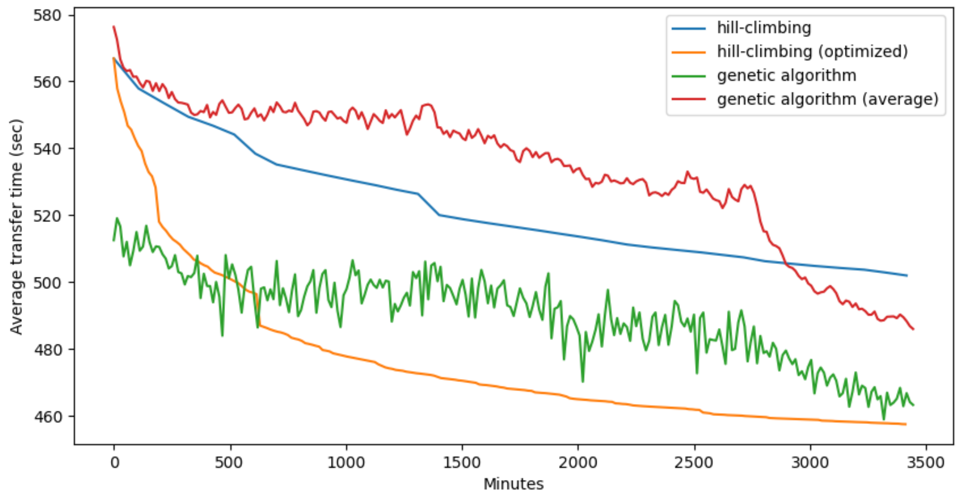

Figure 7: Improvements in the 500-stations PTN

Lastly, the graph shows why none of the algorithms has reached its finish condition. The reason for this being that the improvements of both hill-climbing algorithms and the variance of the genetic algorithm where still too high.

Figure 8 shows the values of the same time schedules from Figure 7 but in this case within the complete PTN including all the routes and stations. This graph basically depicts the same picture with the optimized hill-climbing algorithm (6753.0s, 5.7%) and the genetic algorithm (6759.7s, 5.6%) outperforming the original hill-climbing algorithm (6948.7s, 3.0%) once again.

Both hill-climbing algorithms first descend very fast and then slow down massively. The optimized hill-climbing algorithm has two phases where it completely stagnates, split by one phase where it improves only slowly. The original hill-climbing algorithm on the other hand eventually even reverts to a worse PTN and then takes a very long time to find a new overall best PTN.

Finally, with every PTN now having many more stations, the genetic algorithm is prone to a higher variance. Its best PTN can quickly jump from a really bad PTN to a really good one and vice-versa from one iteration to the next one.

7 Ethical Aspect

This study effectively tries to reduce the average transfer time between any two stations. During this procedure, the optimization algorithm might opt to increase the transfer time between two stations in order to decrease the transfer time of several other journeys. This same effect also arises when counting in the passenger demand. Counting in the passenger demand would imply that people which share the same journey have more weight on the outcome of the optimization. Granted, the general passenger satisfaction would rise, however this could lead to some travellers having to deal with longer transfer times. Lastly, travellers that used to arrive exactly on time at their destination might have to take an

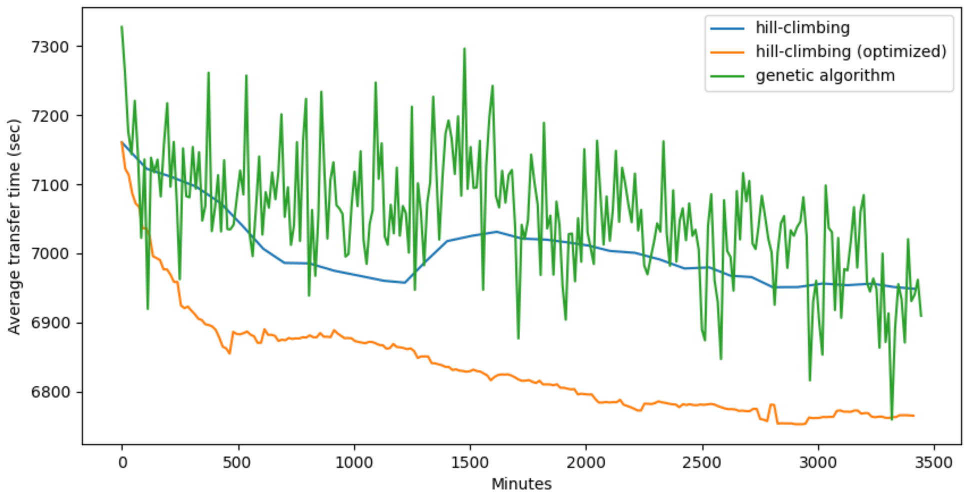

Figure 8: Improvements in the complete PTN

earlier vehicle after the optimization because the route's schedule has been pushed back and they would as a consequence not arrive on time any more. All of these three aspects can be viewed as the classical trolley problem, namely sacrificing a small group of people for a bigger group of people or simply put, the greater good. Whatever the ethical reasoning of the reader's perspective on the trolley problem is can also be applied within this context.

8 Conclusion

With an incredible 19.3% decrease of the average transfer time in the reduced PTN respectively 5.7% decrease in the original PTN, we can conclude that this study was a success.

The average transfer time in the reduced PTN was decreased from 9 minutes and 26 seconds to 7 minutes and 37 seconds. This shows that the average transfer time was reduced significantly. However, the maximum of 6 minutes transfer time preferred by 90% of public transport passengers mentioned at the end of the literature study was not met.

However, as can be seen from the results, a lot of computation time used during the experiment turned out to be wasted when switching from the reduced PTN to the original PTN. While it looked like both hill-climbing algorithms improved in the reduced PTN, they actually did not improve in the complete PTN. The main issue therefore is to decide how big the reduced PTN should be, in other words, how close it should be to the complete PTN. If the reduced PTN includes more stations, improvements are therefore more accurate in the complete PTN but the objective function takes longer to compute. On the other hand, if the reduced PTN gets too close to the complete PTN, the objective function is very likely to have to compute journeys which no passenger is ever likely to ride.

Another important decision point is when to stop the hill-climbing algorithms. The threshold in this study was 0.01 seconds but that might be too low because the algorithm could simply start a new search from a random point instead of chasing minimal improvements in the current search.

Both of these decisions are mainly dependent on the number of cores available as more cores implies that more computations can be done simultaneously. With more cores one could even speed up the calculations by having multiple cores work together on calculating the objective function instead of increasing the pool of PTNs to be calculated during every iteration.

9 Future work

One of the main goals of this study was to show that computer science can be used for sustainable improvements in ways that many people would never had thought of. However, it goes without saying that this model is heavily simplified, especially regarding the assumption of no delay within the entire PTN. This assumption is very risky as a delayed vehicle could lead to passengers not making their transfer and having to wait for a lot longer than actually planned. One way to solve this in the future would be to increase the waiting time for vehicles at the stations so that possible delays can easier be rectified. Another way would be to create an information system that keeps track of every vehicle and makes a vehicle wait for the late arrival of another vehicle. These two solutions together could ensure that vehicles are most always on time and that their passengers can still make their transfer. The downside of this proposal however is that the average travel time for the passengers will be longer.

As already mentioned in the ethical part, taking into account the passenger demand would be another optimization option. We could then put more emphasis on journeys being made by a bigger group of passengers in order to improve the general passenger satisfaction. This would also help in disregarding the journeys within our PTN evaluation that are not being travelled by any passengers.

Generally, a really thorough passenger survey would illustrate a better picture of what needs to be improved and what goals have to be achieved in order to get more citizens to favour a switch from their car to public transport. Furthermore, with Luxembourg's newly inaugurated MeluXina supercomputer, the Luxembourgish government could invest into a similar project with fewer constraints. While in this study we had 32 Intel E5-2630v3 CPUs to our disposal, each one capable of roughly 500 gigaflops, MeluXina has a total calculating power of 10 petaflops which puts it into the top 30 world ranking of the most powerful supercomputers and would significantly speed up the calculations of a similar study.

A broad passenger survey and more computation power could also eliminate two of the constraints set in this thesis. The passenger demand could actually get taken into account to adjust the PTN to suit as many travellers as possible instead of treating every single journey in the PTN equally. In addition, the delay of some vehicles can be simulated more accurately to account for vehicles getting stuck in traffic or having some technical issues along their route.

In this study a single genetic algorithm configuration was used with parameters drawn from the literature study. Only the mutation rate was adjusted as the algorithm did not converge. It would however be really interesting to compare various genetic algorithms with different parameters and find out how well they perform. Especially, comparing a fixed mutation rate to a linearly decreasing mutation rate would be an essential experiment to be conducted.

Finally, the objective function created in this study can be used to evaluate the timetable of newly added routes to an already existing PTN. By shifting the timetable of a new route forwards and backwards within the existing PTN, the optimal time schedule which results in the minimum average transfer time can be found.

References

[1] L. Athanasopoulou, H. Bikas, and P. Stavropoulos, "Comparative well-to-wheel emissions assessment of internal combustion engine and battery electric vehicles," *Procedia CIRP*, vol. 78, pp. 25–30, 2018, 6th CIRP Global Web Conference – Envisaging the future manufacturing, design, technologies and systems in innovation era (CIRPe 2018). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827118309570

[2] R. Kawamoto, H. Mochizuki, Y. Moriguchi, T. Nakano, M. Motohashi, Y. Sakai, and A. Inaba, "Estimation of co2 emissions of internal combustion engine vehicle and battery electric vehicle using lca," *Sustainability*, vol. 11, no. 9, 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/9/2690

[3] F. D. Pero, M. Delogu, and M. Pierini, "Life cycle assessment in the automotive sector: a comparative case study of internal combustion engine (ice) and electric car," *Procedia Structural Integrity*, vol. 12, pp. 521–537, 2018, aIAS 2018 international conference on stress analysis. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452321618301690

[4] Groupe CFL, "Rapport Intégré 2019 Du Groupe CFL," 2019. [Online]. Available: https://groupe.cfl.lu/getmedia/127bee10-b977-4d35-9802-6daca4855dc6/CFL-rapport-integre-2019-web.pdf.aspx

[5] G. X. et al., "Energy, transport and environment statistics," 2020. [Online]. [Online]. Available: https://ec.europa.eu/eurostat/documents/3217494/11478276/KS-DK-20-001-EN-N.pdf

[6] European Commission. "Weekly Oil Bulletin," Accessed Apr. 4, 2021. [Online]. Available: https://ec.europa.eu/energy/data-analysis/weekly-oil-bulletin

[7] STATEC. Statistics Portal of Luxembourg. Accessed Apr. 5, 2021. [Online]. Available: https://statistiques.public.lu/stat/TableViewer/tableViewHTML.aspx?ReportId=13499

[8] STATEC. Statistics Portal of Luxembourg. Accessed Apr. 5, 2021. [Online]. Available: https://statistiques.public.lu/stat/TableViewer/tableViewHTML.aspx?ReportId=13495

[9] STATEC. Statistics Portal of Luxembourg. Accessed Apr. 5, 2021. [Online]. Available: https://statistiques.public.lu/stat/TableViewer/tableViewHTML.aspx?ReportId=13520

[10] M. A. Turnquist, "Strategies for improving reliability of bus transit service," *Transportation Research Record*, no. 818, pp. 7–13, 1981. [Online]. Available: https://trid.trb.org/view/174188

[11] P. G. Furth and F. B. Day, "Transit routing and scheduling strategies for heavy demand corridors," *Transportation Research Record*, no. 1011, pp. 23–26, 1985. [Online]. Available: https://trid.trb.org/view/271167

[12] T. Liu, A. A. Ceder, J. Ma, and W. Guan, "Synchronizing public transport transfers by using intervehicle communication scheme: Case study," *Transportation*

Research Record, vol. 2417, no. 1, pp. 78–91, 2014. [Online]. Available: https://doi.org/10.3141/2417-09

[13] M. M. Nesheli, A. A. Ceder, and T. Liu, "A robust, tactic-based, real-time framework for public- transport transfer synchronization," *Transportation Research Procedia*, vol. 9, pp. 246–268, 2015, papers selected for Poster Sessions at The 21st International Symposium on Transportation and Traffic Theory Kobe, Japan, 5-7 August, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235214651500174X

[14] Z. LIU, J. SHEN, H. WANG, and W. YANG, "Regional bus timetabling model with synchronization," *Journal of Transportation Systems Engineering and Information Technology*, vol. 7, no. 2, pp. 109–112, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157066720760018X

[15] J. Yin, L. Yang, T. Tang, Z. Gao, and B. Ran, "Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches," *Transportation Research Part B: Methodological*, vol. 97, pp. 182–213, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261516305082

[16] T. Liu and A. (Avi) Ceder, "Integrated public transport timetable synchronization and vehicle scheduling with demand assignment: A bi-objective bi-level model using deficit function approach," *Transportation Research Procedia*, vol. 23, pp. 341–361, 2017, papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146517302983

[17] J. P. Fonseca, E. van der Hurk, R. Roberti, and A. Larsen, "A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling," *Transportation Research Part B: Methodological*, vol. 109, pp. 128–149, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261517304915

[18] R. C. W. Wong, T. W. Y. Yuen, K. W. Fung, and J. M. Y. Leung, "Optimizing timetable synchronization for rail mass transit," *Transportation Science*, vol. 42, no. 1, pp. 57–69, 2008. [Online]. Available: https://doi.org/10.1287/trsc.1070.0200

[19] M. Abdolmaleki, N. Masoud, and Y. Yin, "Transit timetable synchronization for transfer time minimization," *Transportation Research Part B Methodological*, vol. 131, pp. 143–159, 12 2019. [Online]. Available: https://doi.org/10.1016/j.trb.2019.12.002

[20] B. Selman and C. P. Gomes, "Hill-climbing search," *Encyclopedia of cognitive science*, vol. 81, p. 82, 2006.

[21] E. Aarts, J. Korst, and W. v. Michiels, "Simulated annealing," in *Search methodologies*. Springer, 2005, pp. 265–282.

[22] E. Aarts, J. Korst, and W. v. Michiels, "Tabu search," in *Search methodologies*. Springer, 2005, pp. 243–259.

[23] E. Aarts, J. Korst, and W. v. Michiels, "Genetic algorithms," in *Search methodologies*. Springer, 2005, pp. 93–109.

[24] K. Gkiotsalitis and N. Maslekar, "Towards transfer synchronization of regularity-based bus operations with sequential hill-climbing," *Public Transport*, vol. 10, pp. 335–361, 2018. [Online]. Available: https://link.springer.com/article/10.1007/s12469-018-0178-2

[25] V. Poorjafari, W. Yue, and N. Holyoak, "A comparison between genetic algorithms and simulated annealing for minimizing transfer waiting time in transit systems," *International Journal of Engineering and Technology*, vol. 8, pp. 216–221, 03 2016. [Online]. Available: https://doi.org/10.7763/IJET.2016.V6.888

[26] L. Jansen, M. Pedersen, and O. Nielsen, "Minimizing passenger transfer times in public transport timetables," *Proceedings of the 7th Conference of the Hong Kong Society for Transportation Studies: Transportation in the Information Age*, 01 2002.

[27] J. Parbo, O. A. Nielsen, and C. G. Prato, "User perspectives in public transport timetable optimisation," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 269–284, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X14002526

[28] Y. Shafahi and A. Khani, "A practical model for transfer optimization in a transit network: Model formulations and solutions," *Transportation Research Part A: Policy and Practice*, vol. 44, no. 6, pp. 377–389, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856410000467

[29] X. Feng, X. Zhu, X. Qian, Y. Jie, F. Ma, and X. Niu, "A new transit network design study in consideration of transfer time composition," *Transportation Research Part D: Transport and Environment*, vol. 66, pp. 85–94, 2019, special Issue on Electromobility for Green Transportation Systems and Sustainable Environment. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920917306107

[30] F. Cevallos and F. Zhao, "Minimizing transfer times in public transit network with genetic algorithm," *Transportation Research Record*, vol. 1971, no. 1, pp. 74–79, 2006. [Online]. Available: https://doi.org/10.1177/0361198106197100109

[31] F. Zhao, "Large-scale transit network optimization by minimizing user cost and transfers," *Journal of Public Transportation*, vol. 9, no. 2, pp. 107–129, 2006. [Online]. Available: https://scholarcommons.usf.edu/jpt/vol9/iss2/6/

[32] Z. Guo and N. H. Wilson, "Assessing the cost of transfer inconvenience in public transport systems: A case study of the london underground," *Transportation Research Part A: Policy and Practice*, vol. 45, no. 2, pp. 91–104, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856410001564

[33] M. R. Amin-Naseri and V. Baradaran, "Accurate estimation of average waiting time in public transportation systems," *Transportation Science*, vol. 49, no. 2, pp. 213–222, 2015. [Online]. Available: https://doi.org/10.1287/trsc.2013.0514

[34] S. Chowdhury and A. A. Ceder, "The effect of interchange attributes on public-transport users'intention to use routes involving transfers," *Psychology and Behavioral Sciences*, vol. 2, no. 1, pp. 5–13, 2013. [Online]. Available: https://doi.org/10.11648/j.pbs.20130201.12

[35] I. Olivková, "Evaluation of quality public transport criteria in terms of passenger satisfaction," *Transport and Telecommunication Journal*, vol. 17, no. 1, pp. 18–27, 2016. [Online]. Available: https://doi.org/10.1515/ttj-2016-0003

[36] Public Transport Administration. Ministry of Mobility. Accessed Apr. 19, 2021. [Online]. Available: https://data.public.lu/en/datasets/horaires-et-arrets-des-transport-publics-gtfs/

[37] Google Developers. Google Inc. Accessed May 14, 2021. [Online]. Available: https://developers.google.com/transit/gtfs/

[38] OpenStreetMap contributors. OpenStreetMap. Accessed May 14, 2021. [Online]. Available: https://www.openstreetmap.org

[39] Open Source Routing Machine. OSRM. Accessed Jun 1, 2021. [Online]. Available: http://project-osrm.org/

[40] Vrije Universiteit Amsterdam. Accessed Jun 24, 2021. [Online]. Available: https://www.cs.vu.nl/das5/

[41] Software in the Public Interest. Accessed Jun 24, 2021. [Online]. Available: https://www.open-mpi.org/

[42] T. Nieuwenhuis. GitHub. Accessed Jul 5, 2021. [Online]. Available: https://github.com/tns410/BachelorProject