A Multi-objective Approach to the Transit Network Design Problem

Vasco Silva
vasco98dias@gmail.com
Instituto Superior Técnico and INESC-ID
Lisbon, Portugal

Abstract

Mobility is a key part of modern civilization. The COVID-19 pandemic made apparent the necessity to adapt our public transport systems to the ever-changing users' needs. In this work we propose a solution to improve the bus network in the city of Lisbon, Portugal. The system takes individual trip data given by smartcard validations at CARRIS buses and METRO stations and uses them to estimate the origin-destination demand in the city and then design a network that better fits that demand. Route scheduling is optimized separately from the route topology. To these ends, Genetic Algorithms are used considering both single and multi objective approaches. The single objective formulation is based on the human rating of the networks in the approximated Pareto Front. A linear regression is used to infer the weights for a weighted sum of the different objectives. The single objective optimization processes proved to improve on the multi objective optimization results with reductions in objective functions up to 28.3%. The system manages to reduce the number of routes in the network from 309 to 200 which then corresponds to a reduction in 59.8% of distance covered by buses daily when frequencies are optimized. All the passenger related objectives, including travel time and transfers per trip are improved with only the unsatisfied demand lightly increasing from 0.7% to 1.3%.

CCS Concepts: • **Applied computing** → **Transportation**; • **Theory of computation** → **Evolutionary algorithms**.

Keywords: Public Transport Network; Optimization; Route Planning; Multimodality; Multi Objective Optimization

1 Introduction

Mobility is a central aspect in modern societies, allowing people to take part in a multitude of activities that are the identity of our days. With the worldwide population increasing, and more people using the roads, problems such as congestion and air pollution arise. One way to address these, is to provide the community with an efficient public transport network.

The COVID-19 pandemic radically changed people's habits and even during periods of normal activity, over time, demand will change and efforts should be made to make sure the Public Transport Networks meet individuals' needs.

The goal of this work is to propose an improved design to CARRIS bus network, one that uses the available buses in a more efficient way and serves the clients as best as possible with the available resources, while trying to reduce the environmental impact of the network. Here, serving the client well means i) providing a service with good city coverage, ii) low travel times, iii) low transfer rates, iv) low waiting times and, v) satisfying safety guidelines such as maximum occupancy and recommended individual distances.

This work also approaches the Network Design Problem from the Multi Objective Optimization perspective. Previous works, when faced with multiple objectives, used a weighted sum of all the objectives as an objective to a Single Objective Optimization process. In this work, weights are inferred from a human rating of the networks in the Pareto Front approximation. By approaching the problem in this way, we hope to provide decision makers with better options and tools to better decide changes to be made in the public transport systems.

2 Optimization and Genetic Algorithms

Finding the best elements, $\mathbf{x}^*$, from a set of alternatives, $\mathbb{X}$, according to a set of criteria, $F = \{f_1, f_2, ..., f_m\}$ is the essence of optimization. $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is a set of what is called design or decision variables. $f_i : \mathbb{X} \mapsto Y, \quad (i = 1, 2, ...m)$, with $Y \subseteq \mathbb{R}$, are the criteria or objective functions. These problems can have constraints that limit the values that the design variables can take. We can formalize optimization problems as follows:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{X}}{\text{minimize}} \quad & f_i(\mathbf{x}), \quad (i = 1, 2, ..., m) \\ \text{subject to} \quad & h_j(\mathbf{x}) = 0, \quad (j = 1, 2, ..., o) \\ & g_k(\mathbf{x}) \leq 0, \quad (k = 1, 2, ..., p) \quad . \end{aligned} \tag{1}$$

In Single Objective Problems ($m = 1$), it is easier to distinguish the quality of solutions since we have only one criteria, so the lower the value of that objective, the better the solution. However, it becomes harder to make this distinction when we have more than one objective. For example, if we have two solutions for a bi-objective problem, $\mathbf{x}^1$ and $\mathbf{x}^2$, with $f_1(\mathbf{x}^1) < f_1(\mathbf{x}^2)$ and $f_2(\mathbf{x}^1) > f_2(\mathbf{x}^2)$, we cannot say with absolute certainty which solution is best. We can convert a multi objective problem into a single objective one by creating a new objective function defined as:

$$f'(\mathbf{x}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x}) \quad , \tag{2}$$

and then increase the objective value when constraints are violated. This formulation will, however, limit the variety of solutions we can find since they do not reveal to the user where are the compromises between objectives. Pareto optimality becomes a relevant definition when dealing with multi objective problems as it provides us with a definition of optimality that considers multiple objectives. A solution is Pareto optimal if we cannot improve one objective without damaging the quality of the remaining objectives. A Pareto optimal solution compromises the different objectives in an optimal way. Different Pareto optimal solutions represent different balances between the objectives. The set of all Pareto optimal solutions is called the Pareto Frontier. Pareto optimality is tightly coupled to the definition of domination. A solution $\mathbf{x^1}$ dominates another solution $\mathbf{x^2}$, in which case, we write $\mathbf{x^1} \prec \mathbf{x^2}$ if, and only if:

$$\begin{aligned} \forall i \in \{0, ..., m\} : f_i(\mathbf{x^1}) \leq f_i(\mathbf{x^2}) \quad \wedge \\ \exists j \in \{0, ..., m\} : f_j(\mathbf{x^1}) < f_j(\mathbf{x^2}) \quad . \end{aligned} \tag{3}$$

We can define the Pareto Frontier, $\mathbf{X}^*$ as the set of points that are not dominated by any other solution. In other words:

$$\mathbf{x}^* \in \mathbf{X}^* \Leftrightarrow \nexists \mathbf{x} \in \mathbb{X} : \mathbf{x} \prec \mathbf{x}^* \quad . \tag{4}$$

Single Objective Algorithms try to find the best solution according to a single criteria while Multi Objective Algorithms try to give us an approximation of the Pareto Front. Having the Pareto Front allows us to look at different compromises and assess their quality.

Genetic Algorithms (GA) are population-based optimization algorithms that try to mimic nature's evolutionary process. Solutions are individuals in a population represented as a string of symbols that encode a solution for the problem we are trying to solve. Individuals reproduce and mutate to give place to new individuals. Reproducing means combining the genetic code of two individuals to make new solutions that are based on their parents and mutation means randomly changing the genetic code of an individual to introduce variety and try to escape local minimums. The quality of the populations increases as the algorithms finishes more and more iterations.

The Classic Genetic Algorithm was one of the first to use these ideas in optimization and was introduced by John H. Holland [8] in 1976. It is a single objective optimization algorithm and since then, many ideas have been proposed to build on the original idea. The Non-dominated Sorting Genetic Algorithm II is a multi objective optimization algorithm proposed by Deb et al. [5]. These are the algorithms used in the present work.

3 Related Work

In this line of work, some works try to change the configuration of routes in a bus network. That is called the Transit Network Design Problem (TNDP). Others try to optimize both the route configuration and their respective working frequencies. This is called the Transit Network Design and Frequencies Setting Problem (TNDFSP). If we are trying to attribute frequencies to an already existing route set, we are solving the Transit Network Frequencies Setting Problem (TNFSP).

Newell [10] points out the non-convexity of TNDP which is illustrated by the fact that increasing the frequency of buses in some routes, i.e. increasing operation costs may lead to an increase in user costs. Baaj and Mahamassi [2] also point out the combinatorial explosion of the problem caused by its discrete nature. Additionally, the problem is usually formulated with multiple objectives and, while bus frequencies are usually depicted as decision variables in problem formulations, the network itself, in terms of routes and the stops in each route, is not depicted in problem formulations at all. This makes Mixed Integer formulations like the one proposed by Wan and Hong [14] rather cumbersome and ineffective to solve TNDP. In Wan and Hong's formulation, the presence of each node in each route is modeled as a binary variable. This makes it so that a small network with 10 nodes and 19 links, ends up having 363 binary decision variables, 30 integer decision variables and 303 continuous decision variables, when we try to solve TNDP for it. Since the network under study is much larger than the one in Wan and Hong, we will only analyze review works that use metaheuristics.

Many works have taken different approaches to solve these problems. The common denominator between them all is that, when confronted with multiple objectives, all the works reduce the problem to a single objective one as we did with Equation 2 and then solve the single objective variant [7], reducing the variety of solutions found by these works.

Jia et al. [9] tackles TNDP for the bus network in Xi'an, a city in Northwest China. This study tries to improve network sustainability and robustness by using Complex Network Theory. No efforts are put forth to make optimizations in the perspective of the users nor in the perspective of the service provider. They do not use any conventional optimization algorithm, instead they develop their own method which takes as input the current network topology. The method did improve Xi'an's network sustainability and cut down its average path length. The main takeaway is that we can use some concepts of Network Theory to evaluate how robust and sustainable our network is without looking at OD demand.

Yu et al. [15] used Ant Colony Optimization to solve TNDP. Their goal was to maximize passenger flow per unit length and minimize transfers. In their model, OD pairs work as the nest and the food source for several sub-colonies and the pheromone trail is based on passenger density. They tested their work on an existing network in the city of Dalian with 3,200 links and 2,300 nodes with 89 lines. The algorithm was able to reduce the number of lines to 61 and did so without compromising satisfied demand. Additionally, the optimized bus lines are shorter than the existing ones and the amount of demand satisfied with no transfers went from 41% to 51%.

Pattnaik et al. [11] used GA to solve TNDFSP. The work focus on minimizing traveling times and minimizing operating time of buses, given some feasibility constraints. Their approach worked in two phases, first a Candidate Route Set Generation Algorithm (CRGA) generates a set of routes. Then a GA computes the optimal set of routes and their frequencies. The GA is used in two different ways. In Fixed String Length Configuration (FSLC), every network in the population has the same number of routes. In Variable String Length Configuration (VSLC), networks have a variable number of routes. VSLC benefits from insertion and deletion operators. In this work the method developed by Baaj and Mahamassi [1] is used to evaluate the networks. The methods are tested in a subset of a network in Madras, South India, but there is no comparison to the network operating there. The objective was to compare FSLC and VSLC. FSLC produced significantly better results leaving no demand unmet, but it needed more computing time as the GA has to run for different network sizes.

Chien et al. [4] also used GA in the context of TNDFSP but in a different way. Their goal was to determine optimal feeder bus routes, not an optimal network. Feeder routes bring people from (to) several points to (from) a central hub. Their objective function also minimizes user and operator costs. The supplier cost is a function of the round trip time of the buses and the headway. The user cost involves user access cost, user wait cost and user in-vehicle cost. They tested their approach in small networks and found that, when the parameters are properly tuned, the GA can find the optimal route.

Bielli et al. [3] used GA to solve TNDFSP. Their GA modeling is quite different from that of [11]. Bielli et al. use a fixed length representation in which each chromosome is a network and each gene represents a pre-generated line with a frequency and an on/off switch that indicates whether the route is active in the network or not. This means that every pre-generated route will exist in every network but with different frequencies and it can be active or not. Their *ff* is a weighted sum of efficacy, efficiency and quality of service metrics. The work was tested on a city in middle-north Italy called Parma and the best result provides an improvement of 90%. This result was obtained after 66 iterations with a mutation probability of 0.1 and a crossover probability of 0.8. The routes used are taken from previous projects on the matter and make a total of 80 candidate routes.

Fan and Machemehl [6] also used GA to solve TNDFSP. Their objective function considers, user costs, operator costs and unsatisfied total demand costs. The relative importance of the costs can be tweaked. The constraints considered focus on controlling the load factor on any given route. Similarly to Pattnaik et al., their approach starts with a CRGA which uses Dijkstra's Algorithm and Yen's k-shortest paths algorithm to generate routes between OD pairs. Then, a GA is used, modeling the problem in similar ways to Pattnaik et al. and Chien et al.. The network analysis algorithm assigns demand to routes and sets the frequency for each route. The algorithm was tested on an example network and no comparison with existing networks was done. They did compare, however, different methods on the same network and found that the GA outperforms other population based methods. They found the optimal crossover probability to be 0.8 and the optimal mutation probability to be 0.1, just like Bielli et al. [3]. The optimal population size was 60.

4 Solution and Problem Formulation

4.1 Preprocessing

To run an optimization process on a bus network that operates on top of a road network, we first have to incorporate the available bus stops, as nodes, on the road network. That way, we can then define a bus route as a sequence of stops and assume that the buses travel between the stops through the optimal time path. The road network was retrieved from the Open Street Map (OSM) database through the OSMnx python package which gives us the network as a NetworkX graph. The stop locations were taken from the CARRIS General Transit Feed Specification (GTFS). Since the stops and networks are built on inaccurate GPS readings, fixing the stops on the network is a complex task. For that end, we used the work of Vuurstaek et al. [13].

There are stops that exist in close proximity to others and whose only purpose is to provide people with multiple boarding points so that stops do not become overcrowded. These usually serve different routes, but from an optimization point of view, they are equivalent because they provide entry and alight points in the same general areas. Because we have routes that are generated randomly, we want to have all these stops clustered under the same stop so we do not generate redundant routes. To that end, two stops, u and v are clustered into one if the edge (u, v) exists in the road network and has a cost of less than $100m$ and if $deg_{in}(v) = 1$. We force the edge (u, v) to exist because, if a road node exists between the two stops, then that means that a route can go from u to one other stop, w, without necessarily going through v, making it so that, if stops u and v are clustered under the location of v, that route would be way longer than it should as it would go from v to w. We force $deg_{in}(v) = 1$ because if there is a route that goes from a third stop w to v and we cluster v and u under the location of u then that route would have to go from w to u, potentially making it much longer than what it, in fact, is. The $100m$ requirement is completely adjustable and it is there only so we do not cluster stops on different ends of the same avenue. If stops u and v can be joined and stop v can be joined with a third stop w, then all the stops can be joined under the same cluster. We started with 2193 stops and managed to reduce this number to 1786 using this criteria to join stops.

The origin-destination estimation is also an essential step since buses require smartcard validation only upon entering. In this work, the estimators of Cerqueira et al. [12] were applied to produce OD matrices which then serve as input to our optimization processes.

4.2 Relevant Networks and Interactions

In this work, we use several different graphs to represent everything we need from the domain. It is then important to understand the different types of graphs, what is their purpose, and how they interact.

The road network, G_r, has edges representing road segments and nodes representing road junctions, road ends or bus stops.

The bus network, G_b, is the only one with several instances. This is the object of optimization and, as such, a genetic algorithm population is filled with networks of this type, i.e. at each time step t, we will have a population $P_t = \{G_b^1, G_b^2, ..., G_b^n\}$. These networks are built from a set of routes. A route is just a sequence of stops. All the tram routes are included in the bus networks but they are never interfered with during the optimization process. The edges in this network are identified by an origin stop, a destiny stop and a route making the connection.

The metro network, G_m, also has edges identified by an origin station, a destiny station and a line color connecting the two. We want to assess what the use of the whole transportation network would be like with the bus network under evaluation, so the metro network must be present. It will not be changed and it is a singleton network. The metro trains are assumed to travel at $60Km/h$.

Finally, the walking network, G_w, connects the bus network with the metro network and close bus stops. It is assumed that people are willing to walk up to $300m$ to transfer. The nodes are bus stops or metro stations and the edges represent possible walks between them. The walks are assumed to be a straight line from the origin to the destiny. People are assumed to walk at $5Km/h$. This network is never changed.

During optimization, we integrate the current bus network, under evaluation, with the walking network already connected to the metro network, creating a complete multimodal network, G_c, in which trips that involve walking, bus and metro can be planned.

4.3 Network Evaluation

In order to assess transfer needs, travel times, waiting times and all the other objectives we are striving to improve, we need to know how the public would use the transportation network under evaluation. In this work, passengers are grouped according to their origin-destination (OD) pair. To reduce computational complexity, OD units are not individual bus stops or metro stations, instead Lisbon is divided into a 30×30 grid and the squares are the OD units. Everyone moving between the same pair of squares is assumed to do so through the same trip. The trips are a result of a computation of time optimal paths in the complete network, G_c, between all OD pairs. When a bus network is able to connect an origin and a destination, it does so through a trip, which is a path in the complete network, G_c. For further analysis of the trips, we can divide the trip into stages. There are three different kinds of stages, a bus stage, a walking stage and a metro stage. Practically speaking, a bus stage is a path, along a single route, in the bus network, a metro stage is a path, along a single line, in the metro network and a walking stage is a single edge path that connects two bus stages or a metro stage and a bus stage. This distinction is important because, every time there is a stage change, we apply a transfer penalty and the path cost increases. This penalty intends to simulate user's preference for trips with less transfers if that doesn't incur a big increase in the overall trip time.

Now, we establish the following notation regarding trips and bus networks:

- W^O - the origin set of geographies;
- W^D - the destiny set of geographies;
- W - the set of all origin-destination pairs;
- $q(s,t)$ - the amount of passengers traveling between s and t;
- R - the set of all routes in a bus network;
- $\mathcal{T}$ - the set of all trips between all OD pairs. $\mathcal{T} = \{T(s,t) \mid s \in W^O, t \in W^D\}$;
- $T(s,t)$ - trip from s to t, a sequence of stages $T(s,t) = (T^0(s,t), \ldots, T^n(s,t))$, where $n \in \mathbb{N}_0^+$;
- $T^k(s,t)$ - the k^{th} stage on the trip from s to t. It is defined as sequence of triplets of the form (u,v,r) where u and v are adjacent stops or stations and r is the route/line connecting them or a special marker indicating that the path from u to v was made by foot;
- $T_{bus}(s,t)$ - bus stages in the trip from s to t;
- $T_{metro}(s,t)$ - metro stages in the trip from s to t;
- $T_{walk}(s,t)$ - walking stages in the trip from s to t;
- $t(s,t)$ - travel time between nodes s and t, i.e. $t(s,t) = t_{inv}(s,t) + t_{wai}(s,t) + t_{wal}(s,t)$;
- $t_{inv}(s,t)$ - in-vehicle time between nodes s and t;
- $t_{wai}(s,t)$ - waiting time between nodes s and t;
- $t_{wal}(s,t)$ - walking time between nodes s and t;
- f_r - frequency of service in route r (in buses/hour);

- t_r - time it takes for a bus to go from the starting station to the terminal station in a given route;
- l_r - length of route r;
- h - the number of hours the network is active per day.

Now we can define some quality metrics regarding a bus network. The total length, TL, of a bus network is computed as:

$$TL(G_b) = \sum_{r \in R} l_r \quad . \tag{5}$$

The Unsatisfied Demand, UD, of a bus network is computed as follows:

$$UD(G_b) = 1 - \frac{\sum_{(s,t) \in W} q(s,t) CO(G_b, s, t)}{\sum_{(s,t) \in W} q(s,t)} \quad , \tag{6}$$

where $CO(G_b, s, t)$, the cover function, is one if the bus network G_b provides, along with the rest of the transportation network, a connection between the origin s and destiny t within a number of transfers bellow the maximum allowed and zero otherwise.

The Required Fleet, which is the number of buses required to be in simultaneous circulation to assure the normal functioning of the network is given by:

$$RF(G_b) = \sum_{r \in R} t_r f_r \quad . \tag{7}$$

We use the distance that will be covered by all the buses in operating hours as a proxy for what will be spent to keep the network working. The Operator Costs, OC, are given by:

$$OC(G_b) = \sum_{r \in R} f_r l_r h \quad . \tag{8}$$

In vehicle time (IVT) is computed as follows:

$$IVT(G_b) = \sum_{(s,t) \in W} t_{inv}(s,t) \cdot q(s,t) \quad . \tag{9}$$

The Average Number of Transfers per passenger, ANT, is computed as follows:

$$ANT(G_b) = \frac{\sum_{(s,t) \in W} \Big(|T(s,t)| - |T_{walk}(s,t)| - 1 \Big) q(s,t)}{\sum_{(s,t) \in W} q(s,t)}. \tag{10}$$

The load factor is computed for connections between two adjacent bus stops for each route in which the stops are connected. This happens because different routes are served by different buses, therefore, the load is independent. We cannot compute the load factor by stages because there can exist several trips that use the same connection within the same route but the stage does not necessarily coincide in its entirety. The Load Factor, LF, between stop u and v, serving route r then becomes:

$$LF(G_b, u, v, r) = \frac{pass(u,v,r)}{f_r \cdot h \cdot BUS\_CAP}, \tag{11}$$

where BUS_CAP is the capacity of each bus and $pass(u,v,r)$ is the number of passengers who traveled between stops u and v through route r during the operation hours. This load factor calculation is assuming a uniform fleet and a uniform distribution of passenger flow over time. To formalize, $pass(u,v,r)$ can be defined as:

$$pass(u,v,r) = \sum_{(s,t) \in W} q(s,t) \cdot SIT(s,t,u,v,r), \tag{12}$$

where $SIT(s,t,u,v,r)$ is the Segment in Trip function which is one if $\exists k$: $(u,v,r) \in T^k(s,t)$ and zero otherwise.

4.4 Problem Formulation and Modeling

Now, we can define the optimization problems to solve. The topology optimization or the Transit Network Design Problem (TNDP) is modeled as:

$$\begin{aligned} \underset{G_b}{\text{minimize}} \quad & TL(G_b), \quad UD(G_b), \quad IVT(G_b), \quad ANT(G_b), \\ \text{subject to} \quad & l_r \leq MAX\_ROUTE\_LEN \quad \forall r \in R, \\ & l_r \geq MIN\_ROUTE\_LEN \quad \forall r \in R, \\ & |R| \leq MAX\_NUMBER\_ROUTES, \\ & |R| \geq MIN\_NUMBER\_ROUTES. \end{aligned} \tag{13}$$

In terms of genetic modeling, our proposal uses principles similar to the ones proposed by Pattnaik et al. [11] VSLC. We have a route pool which has the original routes found in the CARRIS network and a set of generated routes. Every network has all the tram routes that are in the original CARRIS network. Apart from that, the initial population is an array of randomly sized networks with routes randomly selected from the route pool. We never commit to a predefined size. Route insertion and deletion operators are introduced in the mutation process. Mutating consists of swapping a random route in the network with a random route in the route pool.

We have two types of generated routes, hub connectors and traversal routes. Hub connectors are routes connecting the busiest stops through the shortest paths and traversal routes are longer routes whose purpose is to enable easier connections between opposite sides of the city.

The Transit Network Frequencies Setting Problem (TNFSP) is modeled as:

$$\begin{aligned} \underset{G_b}{\text{minimize}} \quad & WT(G_b), \quad OC(G_b) \\ \text{subject to} \quad & RF(G_b) \leq AVAILABLE\_FLEET, \\ & LF(u,v,r) \leq MAX\_LF \quad \forall (u,v,r) \in E_{G_b}, \\ & f_r \leq MAX\_FREQUENCY \quad \forall r \in R, \\ & f_r \geq MIN\_FREQUENCY \quad \forall r \in R. \end{aligned} \tag{14}$$

This optimization process is more conventional in the sense that it can easily be represented by a set of variables. We have a variable per route in the network whose frequencies we want to optimize. In this case, we also attempt to optimize the frequencies of all the tram routes. Mutating a frequency set is just choosing a random route and give it a random frequency.

4.5 Single Objective Formulations

After we assess the quality of the solutions that are given by the multi objective formulations, we can have a sense of what we are looking for in a network. After we know that, we can try to get as close as possible to a global optimum that represents the compromise we are looking for in a network. To this end, we will be rating networks generated by NSGA-II. The rating will then serve as reference for a linear regression that will allow us to infer the weights for a weighted single objective formulation (Equation 2) that stand for what we are looking for. We are trying to estimate:

$$\mathbf{w} = (w_0, w_1, w_2, \ldots, w_m)^T, \tag{15}$$

so that we can have a finely tuned objective function, aligned with the perceived needs,

$$f(G_b) = w_0 + \sum_{i=1}^{m} w_i f_i(G_b) \quad . \tag{16}$$

To this end, we find the best weight vector $\mathbf{w}$ such that a given error function, $E(\mathbf{w})$, is minimized. The error functions measure the difference between the target value in the rated records and the estimates:

$$y(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} \quad . \tag{17}$$

Minimizing the error is also an optimization problem, but in this case, we need only solve $\nabla E(\mathbf{w}) = 0$. For the Squared Error function, the optimal weights are given by:

$$\mathbf{w} = (X^T \cdot X)^{-1} \cdot X^T \cdot \mathbf{t}, \tag{18}$$

where X is called the design matrix, defined as:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}. \tag{19}$$

Assuming that we have n bus networks from the NSGA-II algorithm, we have, for our particular TNDP problem:

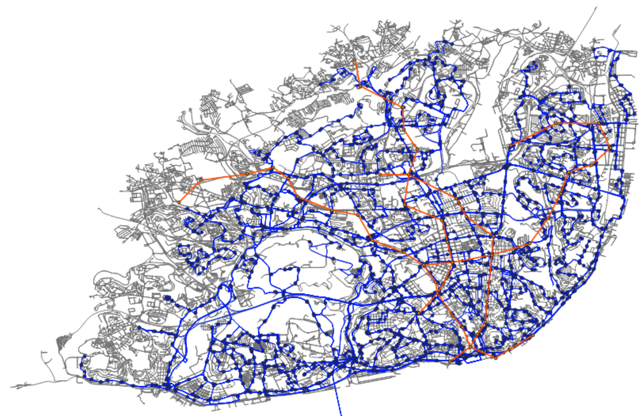

Figure 1. CARRIS network (blue) in October 2019 over the road network and Lisbon METRO network (orange).

$$X = \begin{bmatrix} 1 & TL(G_b^{(1)}) & UD(G_b^{(1)}) & IVT(G_b^{(1)}) & ANT(G_b^{(1)}) \\ 1 & TL(G_b^{(2)}) & UD(G_b^{(2)}) & IVT(G_b^{(2)}) & ANT(G_b^{(2)}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & TL(G_b^{(n)}) & UD(G_b^{(n)}) & IVT(G_b^{(n)}) & ANT(G_b^{(n)}) \end{bmatrix}, \tag{20}$$

and,

$$\mathbf{t} = \left(MAX - r\left(G_b^{(1)}\right), MAX - r\left(G_b^{(2)}\right), \ldots, MAX - r\left(G_b^{(n)}\right)\right)^T, \tag{21}$$

where $r\left(G_b^{(i)}\right)$ is the average rating given to the i^{th} network and MAX is the maximum rating a network can be given. The target for each network is MAX minus its rating because we want the objective function to be minimized and so, the better the rating, the lower the objective function value should be.

5 Results

In this work, we use traffic data from October 2019. The CARRIS bus network deployed at that time can be seen in Figure 1. The network has 309 routes and 2,193 stops. Each route has, on average, 26.2 stops and each station, on average serves 3.7 routes. For the month of October 2019, we have 6.2 million smartcard validations at the bus entrances, spanning 4 days. These validations, as well as validations on the Metro network, are used as the input for the work of Cerqueira et al. [12] for inferring the OD demand.

The results presented in this section were obtained with genetic algorithm experiments ran during 300 iterations, with a population size of 200, a mutation probability of 0.1 and, when applicable, a crossover probability of 0.8. These parameters were decided by doing sensibility analysis on a smaller data sample.

5.1 TNDP

The networks that we discuss in this section where obtained with the NSGA-II algorithm. The maximum allowable number of routes is 400 and the minimum is 200. A sample of 9 networks was chosen for a careful examination and their objective function values can be seen side by side in Figure 2. These were subjected to a rating on a scale of 1 to 10 so that we can infer the weights for a single objective optimization process. The networks are identified by their order in the crowd distance sorting. The number of routes in each network can be seen in Table 1.

network	lisbon	$n0$	$n24$	$n49$	$n74$
route count	309	201	200	200	209
network	$n99$	$n124$	$n149$	$n174$	$n199$
route count	200	207	232	200	200

Table 1. Number of routes of each network obtained with the NSGA-II algorithm.

The big majority of the networks has a number of routes very close to the minimum allowable number of routes. The network with the maximum number of routes is composed of 256 routes. We were expecting a more diverse set of networks when it comes to the number of routes, however the number of routes is not an objective, instead we opted to use the total length of the network as a measure of route efficiency so the diversity lies in the average length of the routes and not in the number of routes. The average route length varies from $9.5Km$ to $13.3Km$ and presents a bimodal distribution with peaks around the $10.5Km$ and $12.5Km$ marks.

All the networks present a total length bellow the original network but no network is capable of satisfying demand better. In terms of in-vehicle time and average transfers, there are networks in the Pareto Front capable of more direct and shorter trips but there are also networks that demand longer trips and a higher transfer rate.

Four people were asked to rate the networks selected from 1 to 10, yielding the following average rating for each network:

network	$n0$	$n24$	$n49$	$n74$	$n99$
rating	7.5	3.5	5	5.75	3.25
network	$n124$	$n149$	$n174$	$n199$	
rating	6	7.5	5.5	4.5	

Table 2. Average rating of each network obtained with the NSGA-II algorithm.

After the linear regression we get the following weights:

$w0$	$w1$	$w2$	$w3$	$w4$
-15.85	1.04×10^{-06}	53.13	7.75×10^{-09}	72.38

Table 3. Objective function weights after Linear Regression.

We ran a Classic GA with the weights in Table 3. We ran the Classic GA again, but with the individual objectives

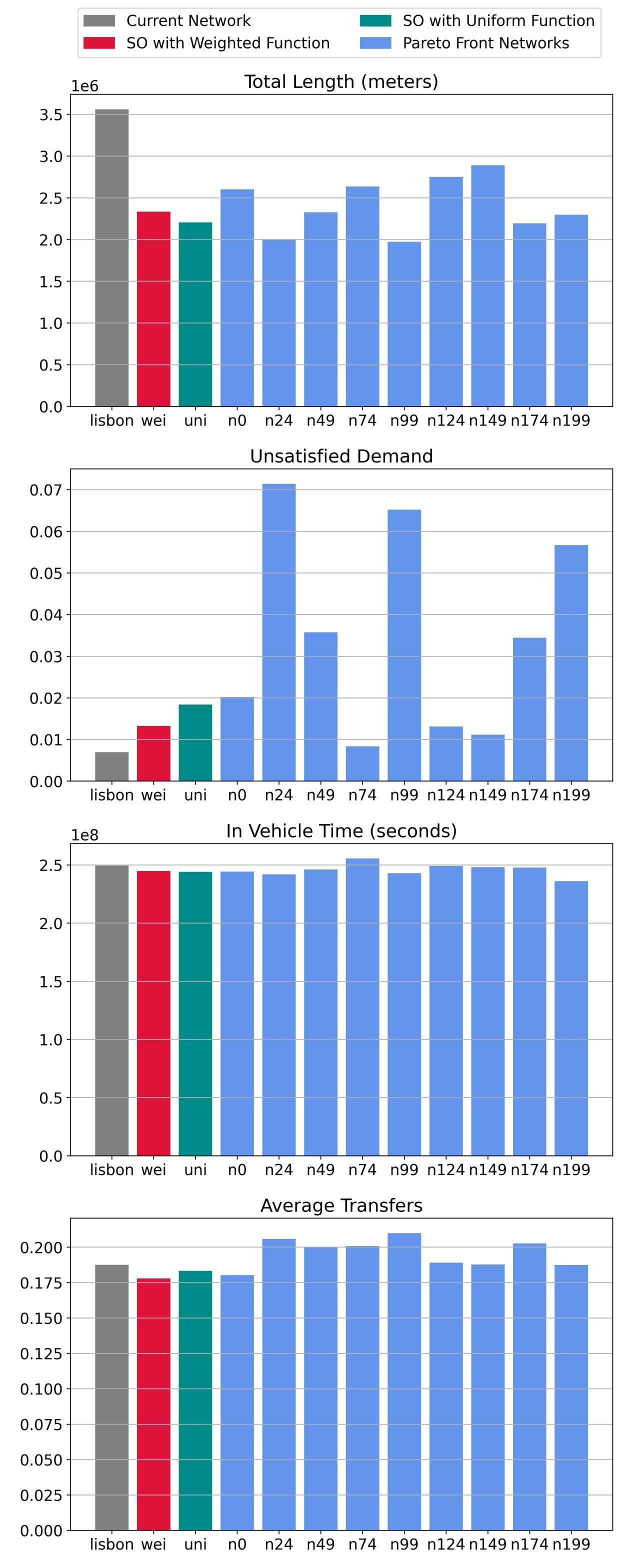

Figure 2. Comparing individual objectives of networks from Multi Objective Optimization and Single Objective Optimization. "lisbon" is the original bus network. "wei" was obtained in a single objective optimization considering the weighted function. "uni" was obtained in a single objective optimization considering the uniform function. The remaining networks resulted from NSGA-II.

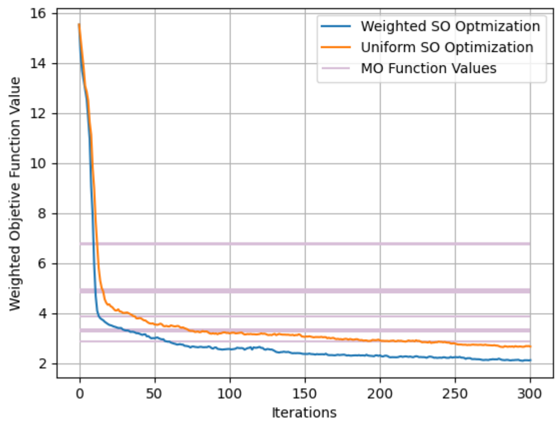

Figure 3. Average objective value along the iterations for the Single Objective GA considering Weighted function in blue and the Uniform function in orange.

summed and normalized with a min-max normalization so we could see the differences between the quality of the solutions when we emphasize different objectives against uniform weights. The objective function in this case becomes:

$$f(G_b) = \sum_{i=1}^{m} \frac{f_i(G_b) - \min(f_i)}{\max(f_i) - \min(f_i)} \tag{22}$$

The average weighted objective function value over the algorithm iterations can be seen in Figure 3 for both experiments ran. According to the criteria we set when rating the networks, when we use the weighted objective function, a Single Objective GA is able to produce networks that are better than the best network we got from the Multi Objective Optimization after about 60 iterations. When we use a uniform weight distribution with normalized objectives we can reach a level of quality superior to the Multi Objective Optimization, yet only after about 200 iterations.

In Figure 2 we can see the individual objectives of the networks obtained through Single Objective Optimization side by side with the ones selected from the Multi Objective Optimization and the original CARRIS network. Both networks obtained with Single Objective Optimization have 200 routes. The highest rated network was network 0 and, as we can see, the network obtained with the weights from the linear regression trumps that network in every objective. The network obtained with uniform weights and normalized objectives does not achieve the same success in the number of transfers but it manages to have smaller total length than the network obtained with a weighted function.

The best network obtained through optimization has 200 routes, 109 less than the original network which translates in a difference of around 1300Km that no longer have to be traveled by the buses. In Figure 5 we can see the differences

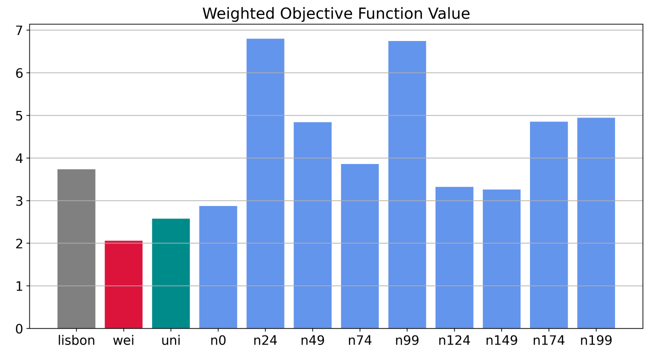

Figure 4. Weighted Objective Function Value for the original network and the networks obtained through optimization.

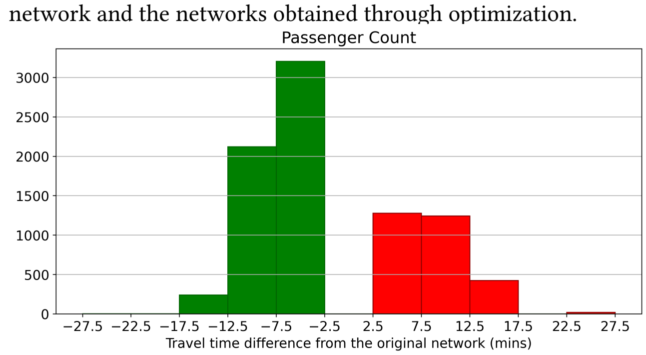

Figure 5. Differences in travel time from the original network to the best network.

in travel times imposed by the new network. Most of the passengers has their traveling time unaffected (differences of up to 2.5 are not considered) but the majority of trips in which there are significant changes were positively affected. The network also enabled more direct trips as seen in Figure 6. No passenger had their trips increased in more than one stage and the passengers that had stages cut off their trips are more than double of the ones that have to transfer one additional time now. Some even cut two transfers off their trips. In these histograms, the central bar is omitted because the total amount of passengers is 566K, which would make the differences between the smaller bars imperceptible. For the big majority of passengers, travel times and transfers remained the same. This is due to the big majority of traffic happening in the center of Lisbon in which most trips are already direct with the existing network. The new network is not able to satisfy as many trips as the original network but the difference is marginal. In general, the new network is able to provide a similar service to the one provided in the original network, but does so in a much more efficient way, using less routes.

5.2 TNFSP

When it comes to frequency setting, we want to do the same transition from multi objective optimization to single objective optimization and assess the existence of any improvements.

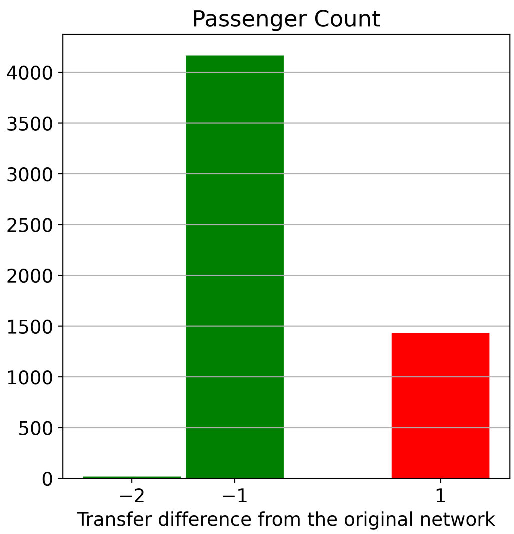

Figure 6. Differences in transfers from the original network to the best network.

We start by doing a multi objective experiment with the original CARRIS network because it provides us with a baseline to compare our results. The problem formulation is the one presented in Section 4.3. We assume a uniform fleet of 750 buses with capacity for 80 people, with a maximum load factor of 1.0, a maximum frequency of 20 buses per hour and a minimum frequency of 1 bus per hour. It would be more accurate to have a non uniform fleet with different buses that can be attributed to different routes. That falls in the Transit Network Scheduling problem which is outside the scope of this work. The results for the experiment can be seen in Figure 7, once again, with the values of the CARRIS network presented for comparison. In general, all frequency sets performed better than the original, with some presenting an improvement of about 30 seconds in the average waiting time and all reducing the total distance traveled by the buses during working hours in almost half.

When it comes to frequency setting, we were not able to get as many people to rate the frequency sets as we previously did with the networks. However, we still want to try to get the best network with set frequencies for a given quality criteria. To that end, we will use a single objective function that is the sum of both objectives normalized, similar to what is presented in Equation 5.1 but with frequency sets as the object of optimization.

To see how much improvement in terms of the TNFSP objectives a refined topology can enable, Figure 8 presents the original network and original frequencies, the original network with optimized frequencies and our optimized topology from solving TNDP with optimized frequencies. In terms of waiting time, the differences from the first case to the second is of 20 seconds and from the second to the third, 20 seconds again, which is not very significant. However from the first case to the second, the total distance traveled is reduced in

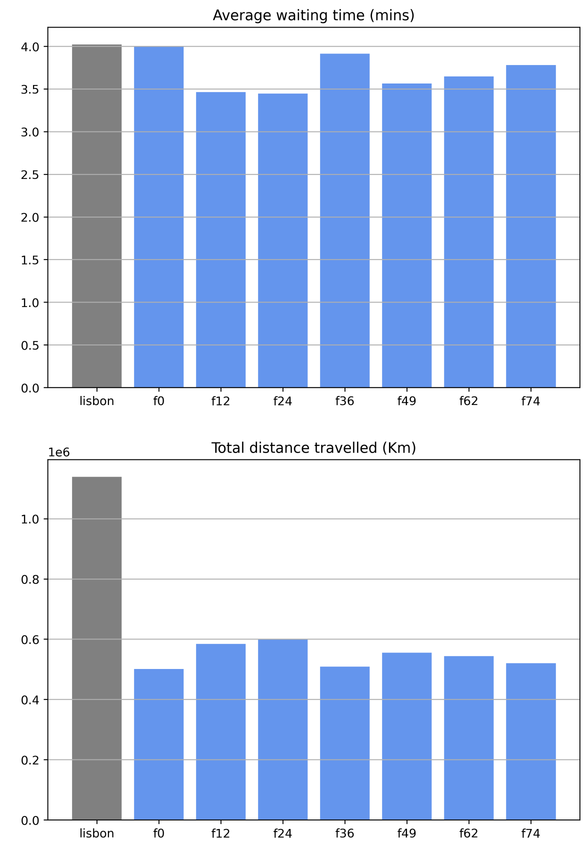

Figure 7. Frequency sets and respective objectives on the CARRIS original route plan, resulting from 300 iterations of NSGA-II with a population size of 200 and a mutation probability of 0.1.

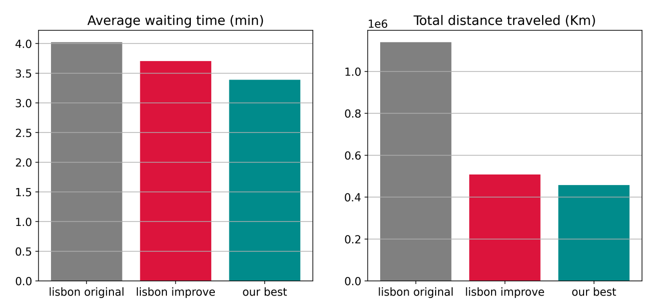

Figure 8. TNFSP objectives for the original CARRIS topology and frequencies (*original* lisbon), original CARRIS topology and improved frequencies (*improve lisbon*) and our best topology with optimized frequencies (our best).

half and from the second to the third cases the distance is reduced in 50 000 Km.

6 Conclusion

In this work, we propose a system to redesign a bus network given the OD demand in the city Of Lisbon, Portugal. Two sub-problems are tackled, first the TNDP and then the TNFSP.

In the first problem, the system decides the set of routes that will define the bus network. In the second, bus frequencies are decided for each route in the route set. Both problems were modeled as an evolutionary problem and, firstly solved with a multi objective algorithm. The results were then rated and weights for a single objective approach were inferred via a linear regression. Finally the problems were solved once more using a Classic Single Objective Genetic Algorithm. Overall, both problems managed to moderately improve the passenger related objectives while massively reducing operator related objectives.

In the TNDP, the four targeted objectives were the total length of the network, the unsatisfied demand, the in-vehicle time and the average transfers. According to our criteria, the best network in the Pareto Front approximation reduces the original network objective function in 23.1% and the network built through single objective optimization provides a 44.0% reduction relative to the original network. This last reduction translates to a 34.6% reduction in total network length (from 309 routes 200), a 5.1% reduction in average transfers, a slight reduction of in-vehicle time and an increase of unsatisfied demand from 0.7% to 1.3%. The transition from Multi Objective to Single Objective optimization proved effective. With a Single Objective optimization considering the weights inferred through our rating process being able to produce better networks than the best one in the Pareto Front after 60 iterations, while a Single Objective optimization considering a uniform weighted sum of all objectives normalized achieved the same only after 200 iterations.

In the TNFSP, the two targeted objectives were the average waiting times and the total distance covered by buses during a whole day as a proxy for operator costs with constraints on the number of buses that can be in simultaneous circulation (fleet), the load factor of the buses and the frequencies. In the multi objective optimization, when trying to optimize frequencies for the original CARRIS topology, all the frequency sets improved on both objectives but once again, passenger costs were mildly reduced while operator costs were greatly reduced. In single objective optimization, we tried optimizing frequencies for the original CARRIS topology and for the best topology from the TNDP. The frequencies in the original topology provide an average decrease of 20 seconds in waiting time and a decrease of 55.4% in total distance traveled while optimized frequencies in the optimized topology provide an average decrease of 40 seconds in waiting time and a decrease of 59.8% in total distance traveled.

6.1 Future Work

The optimization of frequencies can be separated for parts of the day in which passenger flow differ significantly. It would also be relevant to test the use of exclusive routes in certain parts of the day to try to enable shorter trips.

It would be interesting to include more transportation systems, thus allowing us to widen the area under study and highlight the need to add some routes outside the current area of actuation.

Running experiments with discrepant weights or particular constraints could help us study particular network features that help improve certain aspects of the network which could provide useful input for network design.

Finally, there should be reliable ways to go from these idealized models to the real world. Research on how to apply changes on existing networks from the knowledge attained from works similar to the present would be a step in the right direction.

References

[1] M Hadi Baaj and Hani S Mahmassani. 1990. TRUST: A LISP program for the analysis of transit route configurations. *Transportation Research Record* 1283 (1990), 125–135.

[2] M Hadi Baaj and Hani S Mahmassani. 1991. An AI-based approach for transit route system planning and design. *Journal of advanced transportation* 25, 2 (1991), 187–209.

[3] Maurizio Bielli, Massimiliano Caramia, and Pasquale Carotenuto. 2002. Genetic algorithms in bus network optimization. *Transportation Research Part C: Emerging Technologies* 10, 1 (2002), 19–34.

[4] Steven Chien, Zhaowei Yang, and Edwin Hou. 2001. Genetic algorithm approach for transit route planning and design. *Journal of transportation engineering* 127, 3 (2001), 200–207.

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[6] Wei Fan and Randy B. Machemehl. 2006. Optimal Transit Route Network Design Problem with Variable Transit Demand: Genetic Algorithm Approach. *Journal of Transportation Engineering-asce* 132, 1 (2006), 40–51.

[7] Reza Zanjirani Farahani, Elnaz Miandoabchi, W.Y. Szeto, and Hannaneh Rashidi. 2013. A review of urban transportation network design problems. *European Journal of Operational Research* 229, 2 (2013), 281–302.

[8] John Henry Holland. 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence* (1 ed.). MIT press.

[9] Guo-Ling Jia, Rong-Guo Ma, and Zhi-Hua Hu. 2019. Urban transit network properties evaluation and optimization based on complex network theory. *Sustainability* 11, 7 (2019), 2007.

[10] Gordon F Newell. 1979. Some issues relating to the optimal design of bus routes. *Transportation Science* 13, 1 (1979), 20–35.

[11] SB Pattnaik, S Mohan, and VM Tom. 1998. Urban bus transit route network design using genetic algorithm. *Journal of transportation engineering* 124, 4 (1998), 368–375.

[12] Rui Henriques Sofia Cerqueira, Elisabete Arsénio. 2020. Integrative analysis of traffic and situational context data to support urban mobility planning. *European Transport Conference (ETC), Milan, Italy* (2020).

[13] Jan Vuurstaek, Glenn Cich, Luk Knapen, Wim Ectors, Tom Bellemans, Davy Janssens, et al. 2018. GTFS bus stop mapping to the OSM network. *Future Generation Computer Systems* 110 (2018), 393–406.

[14] Quentin K Wan and Hong K Lo. 2003. A mixed integer formulation for multiple-route transit network design. *Journal of Mathematical Modelling and Algorithms* 2, 4 (2003), 299–308.

[15] Bin Yu, Zhongzhen Yang, Chuntian Cheng, and Chong Liu. 2005. Optimizing bus transit network with parallel ant colony algorithm. *Proceedings of the Eastern Asia Society for Transportation Studies* 5, 1 (2005), 374–389.