Optimization of Bus Stops, New Pick-up and Drop-off Locations for Public Transportation

Cristiano Martins Monteiro[1], Flávio Vinícius Cruzeiro Martins[2] and Clodoveu Augusto Davis Jr[1]

[1] Federal University of Minas Gerais, Belo Horizonte, Brazil
cristianomartinsm@ufmg.br, clodoveu@dcc.ufmg.br
[2] Federal Center for Technological Education of Minas Gerais, Brazil
flaviocruzeiro@decom.cefetmg.br

Abstract. The increase in urban population, together with the expansion of cities, has motivated the study of improvements for dynamics aspects of daily urban life. An important portion of these dynamic aspects is related to the population's routine activities, like commuting using public transportation. This work proposes two meta-heuristics and one integer programming modeling to analyze the location of bus stops and propose new pick-up and drop-off locations in order to avoid long walks to take the bus. A real dataset of the road network was integrated to the location of bus stops in the city of Belo Horizonte. Computing approaches were proposed to optimize the location of bus stops in a scalable way. Experimental results show that many new bus stops are required to improve the quality of the service rendered to the population.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Spatial databases and GIS; G.1.6 [**Optimization**]: Integer programming

Keywords: Bus Stops Optimization, Integer Programming, GIS Application

1. INTRODUCTION

Commuting to school, work, shops and other points of interest is a common routine for inhabitants of metropolitan areas [Logiodice et al. 2015]. The constant increase in urban population[1] and the greater difference between the urban and rural population[2] motivate the study of improvements for urban transportation services. Such improvements are important to achieve efficiency and accessibility, mainly for residents in peripheral regions. Regarding this problem, recent works use public data sources to diagnose or to propose solutions regarding the city's transit system [Santos et al. 2017; Jerônimo et al. 2017].

Commuting in urban areas is commonly performed by means of the public transit system. In the city of Belo Horizonte, Brazil, public transportation comprises buses, taxis and a metro rail system. The city's metro rail runs on the surface and has only one operating line, whose extension is 28 kilometers. This extension is relatively small, given the city has an area of 331 km^2 [Garrides et al. 2016] and maintains a road network of 9,047 kilometers. Thus, in Belo Horizonte, buses and taxicabs (including shared ride services, such as *Uber*) stand out in relation to metro rail because the former means of transportation have much greater coverage than the latter.

[1] http://data.worldbank.org/indicator/SP.URB.TOTL
[2] http://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS

This article is an extended version of Monteiro et al. [017a] presented in XVIII Brazilian Symposium on GeoInformatics (GEOINFO 2017). Authors acknowledge the support of CNPq, CAPES and FAPEMIG, Brazilian agencies in charge of fostering research and development.

Public transport services could provide greater accessibility and attract new passengers if there were more embarkation and disembarkation points (also known as pick-up and drop-off points) spatially distributed throughout the city [Loader and Stanley 2009]. This work aims to determine optimal locations for these new points in the city of Belo Horizonte. This article is an extended version of Monteiro et al. [017a] presented in XVIII Brazilian Symposium on GeoInformatics (GEOINFO 2017). Along with the results from Monteiro et al. [017a], three optimization algorithms are proposed to suggest new points, aiming to benefit as many of the city's neighborhoods as possible.

A dataset of Belo Horizonte's complete road network was used to create a graph, on which the optimization processes for new pick-up and drop-off points have been executed. Given the size of the dataset and the high computational cost needed to perform an optimization process over such a large graph, our implementation must use efficient data storage and retrieval methods. The optimization method proposed in this work is compared to an adapted version of the algorithm presented by Takakura et al. [2015]. Experimental results show the effectiveness of the proposed method in relation to the compared work, and the significant improvements that optimization techniques can propose to the public transportation in Belo Horizonte. A broader distribution of pick-up and drop-off points throughout the city would especially benefit residents of neighborhoods far away from the downtown area [Veras et al. 2016]. The suggestion of these new pick-up and drop-off points can work as a basis for additional bus lines, or to propose the location of new taxi stands.

This article is organized in six sections. Next section discusses related work. Section 3 presents the datasets used in this work and the methodology. Sections 4 and 5 explain the proposed optimization methods. Section 6 shows the results, and Section 7 concludes this article.

2. RELATED WORK

Recent works use open data to analyze the public transport operation or to propose improvements in the location of bus stops and taxis. Some of these works apply meta-heuristics or stochastic processes to achieve their goals. Other works aim to explain city dynamics through public transportation data.

Logiodice et al. [2015] propose an inaccessibility index to measure distance and time spent commuting between zones of a city. The authors applied this index to the São Paulo Metropolitan Area and contextualized results with the amount of trips made in each zone and with the average income of its population. Low income and peripheral residents were found to live in areas with high inaccessibility.

An exploratory analysis about Belo Horizonte was presented by Santos et al. [2017]. The authors analyze the spatial and temporal pattern of traffic accidents, integrating an official dataset and an unofficial dataset collected from Waze[3]. Results indicate that only seven percent of the accidents were reported by both sources.

Spatial and temporal patterns from public transport were also analyzed by Monteiro et al. [2016]. The authors have studied the San Francisco (USA) and Rome (Italy) regions with greater flow of taxicabs, presented the variations of taxi operation along the week, and determined the ten most common places for pick-up and drop-off in San Francisco. Although the downtown area in both cities show higher supply and demand of taxi services, taxi trips were also performed in the peripheral regions.

Silva Júnior et al. [2016] presented a spatial and temporal analysis of the taxi service in Belo Horizonte.Among other results, the authors found that 77% of the routes were serviced by taxi drivers who were 500 meters or less away from the passenger pick-up location. Besides, only 5% of the routes were serviced by taxi drivers whose locations were more than one kilometer away from the passenger. Therefore, if a passenger calls a taxi to pick him up at some location, if there are no taxis nearby at that moment, it is unlikely that the trip will be started.

[3] https://www.waze.com/

The driving distance between the taxi driver and the passenger was the subject of research by Oliveira et al. [2015]. The authors compared optimization techniques in order to define the best driver to respond to a call from a passenger. The algorithms aimed to minimize time and driving distance needed to move from the taxi driver's location to the passenger's location.

In addition to taxis, walking distance to a bus stop in Belo Horizonte was analyzed by Veras et al. [2016]. Data from an origin-destination survey from 2012 were used to analyze the Accessibility Index of the city. The shorter the walking distance to a bus stop and the waiting time until the bus arrives, the greater (and better) is the Accessibility Index for a given region. According to the literature analyzed by Veras et al. [2016], the accessibility of a region is not considered to be bad, in respect to walking distance, if the distance to a bus stop is, on average, less than or equal to 500 meters.The analysis indicates that the 500 meters threshold is exceeded in most of Belo Horizonte, and there are discrepancies on this distance even for neighboring regions.

A Genetic Algorithm was proposed by Takakura et al. [2015] to define the location of ten new bus stops for the city of Nonoichi, in Japan. The authors' goal was to minimize the walking distance between students dormitories and the nearest bus stop with connection to the Kanazawa Institute of Technology. The best solution reduced walking distance by up to 702 meters.

Nalawade et al. [2016] proposed an optimization method to define the space between bus stops in the city of Aurangabad, India. The optimization aims to maximize the covered population. The authors use a Random Walk technique to explore the nodes and edges from the city's road network.

Yao et al. [2017] proposed a bi-objective optimization method to reduce lengthy walks to take a bus, or long times waiting for a taxi, in the city of Dalian, China. The authors proposed a circular bus line in the downtown city in which terminals are set dynamically. The bus stops were chosen based in a set of possible pick-up and drop-off points generated near the passengers' destiny. The two objectives in the work were to minimize passenger and operator travel costs. The optimization applied the Non-dominated Sorting Genetic Algorithm (NSGA-II).

This article differs from previous related works by proposing an optimization method to find new locations of pick-up and drop-off points (NPDPs), and applying it to improve the accessibility in the city of Belo Horizonte. The term Pick-up or Drop-off Point (PDP) will be used in this work to encompass the concept of bus stops, taxi stands, as well as other places destined for people to get in or out of a public transport vehicle. The results of this study can be useful for public transit companies, taxi services (including taxipooling and competitors like Uber), and other private initiatives such as Buser. The dataset used and the proposed optimization method are described in the next section.

3. DATASET AND METHODOLOGY

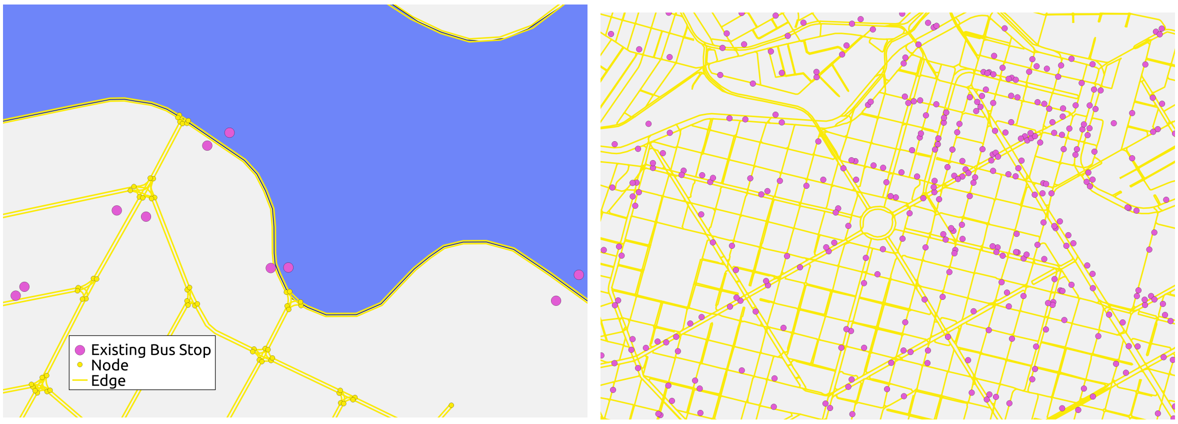
This section presents the dataset used in this work and its methodology. Subsection 3.1 describes the treatment and integration of the road network and bus stop location datasets from Belo Horizonte.Subsection 3.2 explains the selection of neighborhoods for the optimization.

3.1 Data Treatment and Integration

In this work, two datasets from Belo Horizonte were integrated: the road network[4], and the bus stops locations, maintained by the public transport company BHTrans[5]. Figure 1 illustrates two different regions from the city using these datasets. Figure 1 (a) shows the road network (points and lines in yellow) and the bus stops (purple points) near the Pampulha lake. Figure 1 (b) illustrates the edges and bus stops in the downtown region.

[4] https://geodadosbh.pbh.gov.br/

[5] http://servicosbhtrans.pbh.gov.br/bhtrans/e-servicos/S43F01-extracao.asp



(a) Pampulha region

(b) Downtown region

Fig. 1. Illustration of the datasets

Figure 1 (a) shows that the location of bus stops usually does not superimpose an edge of the road network dataset, and that the distance varies between a bus stop and the closest edge. In some cases (especially for edges at a road intersection), it is unclear which edge includes the bus stop. In many of these cases, both the street and bus stop location can be correct, because a bus stop in Belo Horizonte can vary from a simple sign fixed to a pole, to a Bus Rapid Transit (BRT) station. However, if the location is incorrect, it is impossible to automatically determine whether the error is on the road network data, or on the bus stops dataset. According to Monteiro et al. [017b], errors while matching a point to a street (provided by different datasets) can happen due to factors such as: (i) GPS system inaccuracy when calculating the coordinates of the point, street or both; (ii) errors when recording the data; (iii) missing streets or streets set with incorrect direction.

Since the georeferenced data integration is subject to different sources of errors, the optimization method proposed in this work simplifies the location of bus stops and PDPs by simply matching them to the nearest edge. Therefore, some edges had more than one bus stop. About 12% of the bus stops were located at an edge with two or more bus stops. This overlapping of bus stops is considered reasonable, because even a short street segment can have more than one bus stop. Usually, each bus stop in a street segment supports different bus lines. Therefore, all those bus stops actually work as if they were only one bus stop. Next Subsection presents the selection of neighborhoods.

3.2 Selection of Neighborhoods

Veras et al. [2016] show that the number of bus stops by region in Belo Horizonte varies significantly. Nowadays, there are neighborhoods with as few as one bus stop for every two kilometers of streets on average, and neighborhoods with one bus stop for every 200 meters on average. Neighborhoods with a high number of bus stops are not the focus of optimization in this work. Thus, neighborhoods with more than one bus stop for every 800 meters of streets were not selected to be optimized.

Having one bus stop for every 800 meters of streets implies that, for each address contained in that neighborhood, there will be, on average, a bus stop 400 meters away. That threshold was chosen because, as mentioned by Veras et al. [2016] and Nalawade et al. [2016], it is considered reasonable to walk 400 meters to get to a bus stop. Considering this threshold, 299 neighborhoods were selected to be optimized by the meta-heuristic method. Among the neighborhoods not selected are those located in the city's central region, and neighborhoods such as "Gameleira", "Campus UFMG", "São Gabriel" and "Venda Nova", which act as regional centers away from downtown. The following sections presents the optimization methods.

4. META-HEURISTIC OPTIMIZATION METHOD

This section presents the proposed meta-heuristic method to optimize the location of NPDPs. This optimization is based on the *Simulated Annealing* algorithm [Kirkpatrick et al. 1983]. This algorithm is a meta-heuristic that explores a search space looking for the optimal solution for a given problem.

Initially, Simulated Annealing generates a random solution for the problem. From this initial solution, the algorithm evaluates neighbor solutions, walking on the search space towards an optimal solution. In order to prevent the algorithm from getting stuck at a local maximum or minimum, the optimization process also allows (momentarily) solutions worse than the best one found so far. In the Simulated Annealing, the decision to allow a worse solution is given based on a probability. This probability varies along the optimization, leading to larger movements through the search space at the beginning of the optimization (to speed up the process), and smaller movements through the search space at the end of the optimization (to refine the best solution found) [Kirkpatrick et al. 1983].

Meta-heuristics are useful to optimize the location of bus stops when the optimizing area is extensive. Finding a optimal combination of nodes and edges for the transit network design problem (TNDP), for example, is NP-Hard task [Baaj and Mahmassani 1991]. The combinatorial problem applied in this work is similar to the TNDP since it needs to traverse the edges checking distances to suggest an NPDP, and it must not disturb the spacing between the already suggested NPDPs. An simplification for this problem would be to perform Breadth First Search (BFS) on Belo Horizonte's road network graph, restarting the search from every existing bus stop. Inside each BFS a NPDP would be suggested every time a distance threshold is reached without finding another bus stop or a previously suggested NPDP. This method can suggest a different set of NPDPs for every starting bus stop. Therefore, the best solution would be the set with the largest number of suggested NPDPs.

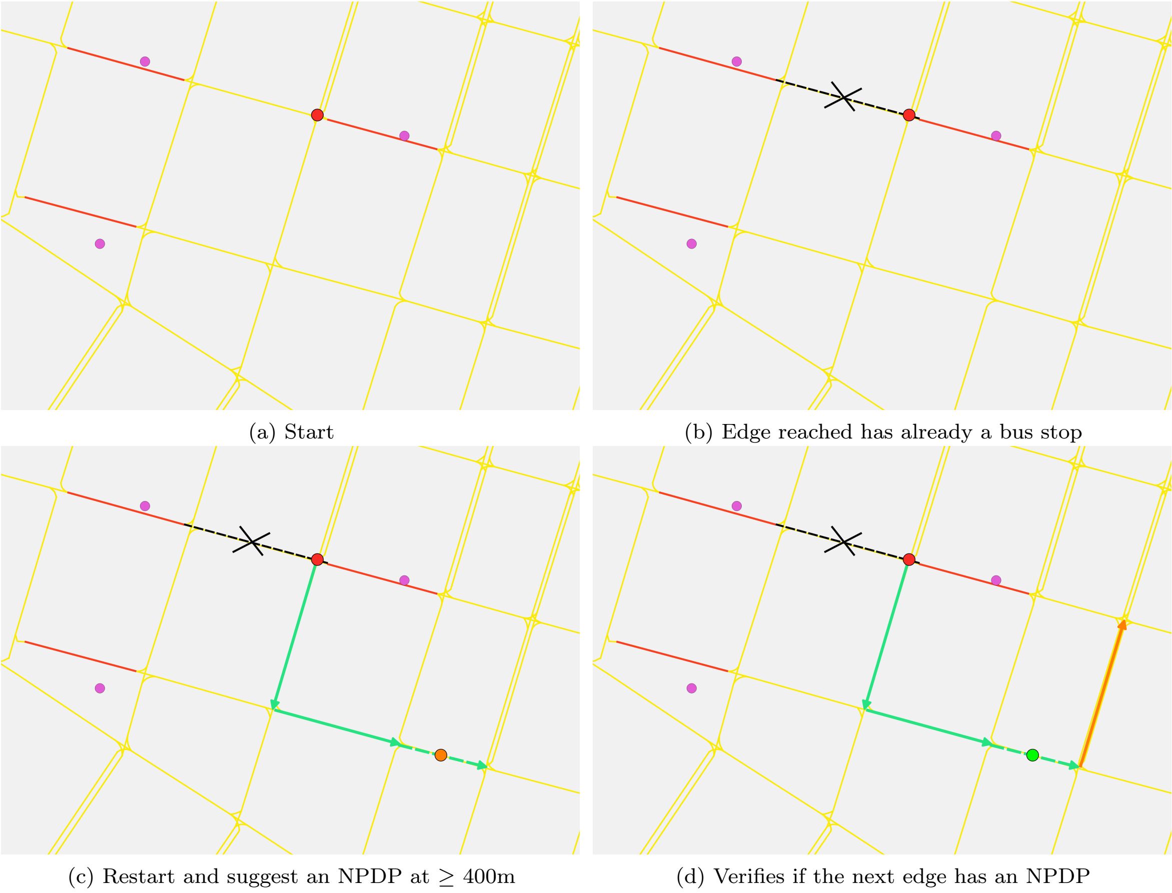
This procedure would have time complexity order of $O(|P| \times (|N| + |E|))$, where $|P|$ is the amount of existing bus stops, $|N|$ represents the number of nodes in the road network, and $|E|$ is the number of edges. Regarding the 299 selected neighborhoods to be optimized, $|P| = 9,428$, $|N| = 127,196$ and $|E| = 201,816$. Keeping one instance of this graph for each individual of a meta-heuristic like a Genetic Algorithm would make the method impracticable due to memory issues. The Simulated Annealing algorithm was chosen because it is a non-population-based meta-heuristic, well suited for this problem. The proposed optimization method that uses Simulated Annealing was implemented in two ways, described in Subsections 4.1 and 4.2

4.1 Random Walk

This optimization method aims to maximize the reach of NPDPs, i.e., the length of street segments served by each NPDP. Reach maximization was performed using Random Walk on the road network's graph, following Nalawade et al. [2016], to optimize the spacing between bus stops.

A neighbor solution for Simulated Annealing would consist in adding or removing an NPDP. However, in our approach, existing bus stops are not updated or removed. The task of adding a NPDP consists in performing a Random Walk (following the real direction of the streets and roads) starting near an existing bus stop, and defining a NPDP for every 400 meters that were walked without coming by any existing bus stop, and without crossing edges or nodes previously walked to add another NPDP. Figure 2 illustrates this process.

Figure 2 (a) displays existing bus stops using purple points, red lines indicate the street segments matched to a bus stop (as described in Section 3.1), and the red point is the starting node of the Random Walk. This starting node is chosen randomly among the nodes close to an existing bus stop. Hence, there will be at least one existing bus stop connecting to the suggested NPDP. This allows using NPDPs to create new bus stops, for example. To prevent a NPDP from being located too close to an existing bus stop or previous NPDP, the Random Walk is restarted whenever an edge with a



(a) Start

(b) Edge reached has already a bus stop

(c) Restart and suggest an NPDP at ≥ 400m

(d) Verifies if the next edge has an NPDP

Fig. 2. Random Walk suggesting an NPDP

bus stop or NPDP is crossed, as illustrated in Figure 2 (b). A NPDP is suggested after at least 400 meters were randomly walked without finding bus stops, NPDPs, or repeating edges already walked when suggesting a previous NPDP. Figure 2 (c) illustrates this procedure. The method also verifies if the NPDP is being located right before other NPDP or bus stop. Therefore, it checks whether the Random Walk's next step will have or not an existing bus stop or NPDP. If the next step is also clear, the NPDP is added, as illustrated by the green point in Figure 2 (d).

The 400 meters spacing was chosen because Nalawade et al. [2016] also used it in their optimization process, applying Random Walk. When looking for a location for the NPDP, if the Random Walk reaches a sink node (node without outgoing edges) or becomes stuck in a cycle (walking more than 100 steps without suggesting an NPDP), the NPDP addition is canceled. The objective function is to maximize the sum of walked distances without reaching an existing bus stop, or reaching a previously suggested NPDP, or visiting an edge or node already visited while suggesting a previous NPDP. The fitness function used is the same as the objective function. Next Subsection presents a version of this method based on the work of Takakura et al. [2015].

4.2 Grid-Based Random Walk

This method is based on the procedure proposed by Takakura et al. [2015] to define new bus stop locations. The method uses a grid with a predefined number of cells, where each cell represents a uniformly divided region. For each region, only one bus stop is suggested. This method ensures that the new bus stops are distributed throughout the city, avoiding bus stop concentrations in small areas.

For the city of Belo Horizonte and the problem of suggesting new pick-up and drop-off points, 8,050 NPDPs would be necessary in order to achieve the goal of 400 meters walk to get to a bus stop in the selected neighborhoods. This number of 8,050 NPDPs was calculated by dividing the total street length by the desired walking distance. Therefore, a grid with 8,050 cells (one for each NPDP) must be created. According to the grid-based approach proposed by Takakura et al. [2015], each cell must have four candidate points to become an NPDP. The optimizing process consists in defining which candidate point will be selected as the suggested NPDP.

However, cells located on lakes or buildings can be away from streets. This reduces the diversity of bus stops, because probably there will be another candidate point close to the nearest street. Figure 3 illustrates this situation near Pampulha lake. Each point indicates a candidate point, but only the green points are near a street segment (edge on the graph). The red points would be avoided along the optimization process, because there is a green point closer to a street segment.

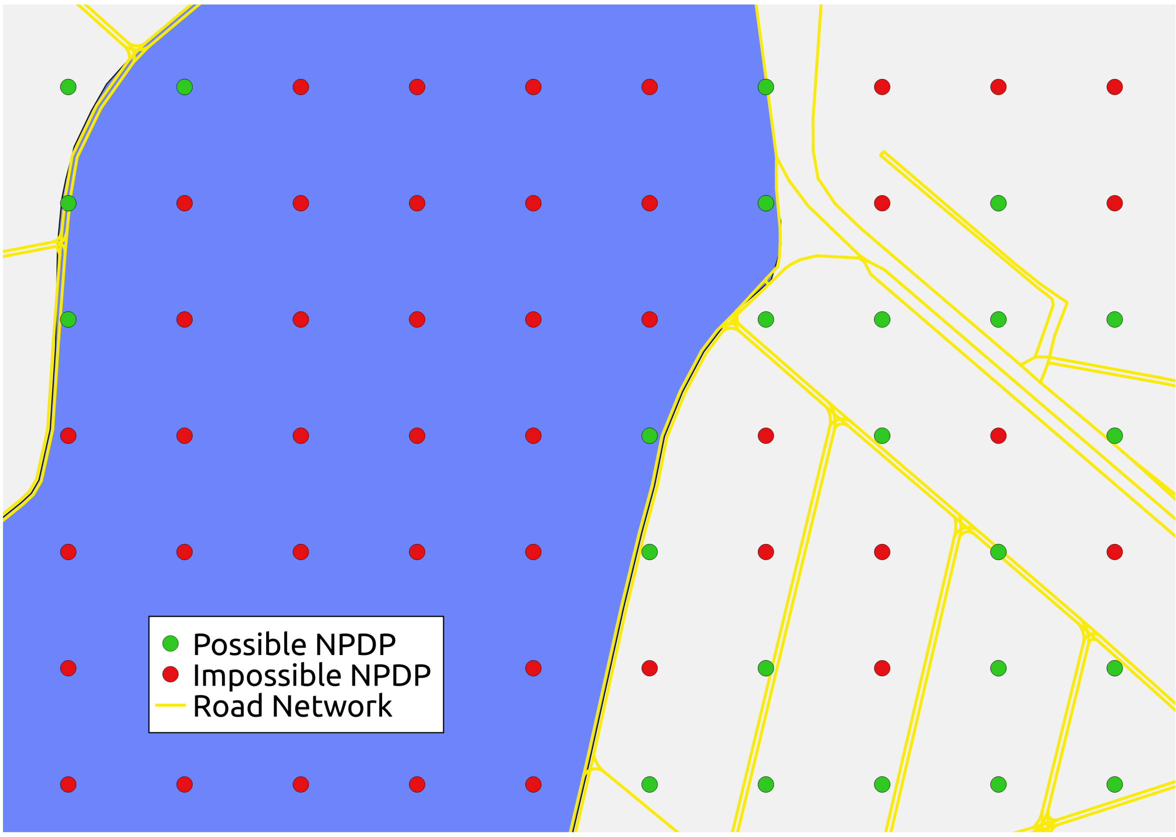


Fig. 3. Grid with possible NPDPs

This method was adapted to the Random Walk as follows: beyond the conditions of walked distance without NPDPs (mentioned in Section 4.1), the suggested NPDP must be close to a candidate point in the grid. The NPDP is considered close to a candidate point in the grid if both match the same edge, as mentioned in Section 3.1. This method does not partition the optimization search in grid cells, but filters the solutions found by Random Walk, allowing only those that match a candidate point. Therefore, although the grid-based approach ensures a more spaced distribution of NPDPs throughout the city, it also reduces the possible locations for NPDPs. Next section presents the Linear Integer Programming Method proposed and its computational challenges.

5. LINEAR OPTIMIZATION METHOD

This section presents the proposed Integer Programming Method to optimize the location of new NPDPs. Optimization was performed using the library CVXPY for convex optimization [Diamond and Boyd 2016] by the solver GLPK_MI[6] for integer-mixed optimization. For integer linear optimization, as in the NPDP location problem, the solver applies the Simplex algorithm to fix the variables to integer using the Branch-and-Cut algorithm.

According to [Kostina 2002], the Simplex algorithm finds the global optimal solution for linear problems in the general form shown in Equation 1:

$$\begin{aligned} & max\ c^T x \\ subject\ to \quad & Ax = b \\ & l \leq x \leq u \end{aligned} \tag{1}$$

where the objective is to maximize the column vector x, which is multiplied by a vector of coefficients c. The vector c is transposed to become a line vector and have the correct dimensions to multiply x. The problem's restrictions exist to avoid the maximization to raise the x so that it produces unfeasible values for the current problem. Restrictions are represented by the equality $Ax = b$, and by the inequalities $l \leq x \leq u$, where A is a matrix of components multiplying x, and b is a column vector with the expected results for that multiplication. The thresholds l and u are, respectively, the lower and upper bound limits for the optimized variable x. The following Subsection explains the proposed method to optimize the location of NPDPs using linear integer programming.

5.1 Modeling Integer Programming

This section presents how the linear integer optimization method was modeled. The optimization's objective in this method is to maximize the number of NPDPs in the city of Belo Horizonte. Following the general Simplex form and its variables from Equation 1, vector x contains the number of NPDPs suggested for each edge of the city's road network. The edges correspond to vector c, having one edge for each element in c. Each column in matrix A represents an edge, in the same order of the elements in c. Each line in matrix A holds one restriction imposed by those edges. The lower bound l was set to 0 to avoid solutions with a negative amount of NPDPs, and the upper bound u was not set. The set of restrictions defined by $Ax = b$ prevents raising the amount of NPDPs for a edge to $+\infty$.

The restrictions in $Ax = b$ ensure that the number of NPDPs surrounding a node in the city's road network will not surpass a threshold. Therefore, an iteration over all nodes from city's network was performed to define matrix A and the vector b. In each iteration, edges that are closer than a threshold d from the current node are selected to compose one restriction. In order to make the distance measure more realistic, the length of sequential edges was computed, instead of simply selecting edges inside a radius from the current node. Initially, 146,505 restrictions (one restriction for each node with outgoing edges) were defined. That process was performed for $d = 200, d = 400$ and $d = 800$ meters. The number of initial restrictions is different from the number of nodes specified in Section 4 because, for this method, the neighborhoods with less than 800 meters, on average, between bus stops were also selected. They were selected because edges near to the neighborhoods' borders must also participate in restrictions generated from nodes located in the near neighborhoods, since they are within d meters.

Figure 4 illustrates the process, using $d = 800$ meters. The points (diamonds) in orange and yellow are nodes from the road network. The edges in purple are the edges that participate in the restriction from the orange point. The edges in red are the edges from the yellow node's restriction. The edges in green represent the overlap between the edges in purple and the edges in red. The restrictions from the orange and the yellow node limit the number of NPDPs to one. Therefore, the edges in purple can

[6] https://www.gnu.org/software/glpk/

have at most one NPDP, and the edges in red can have at most a second NPDP. If a NPDP is set on a edge in green, the edges in purple and the edges in red cannot have another NPDP, since the edges in green already participate in both restrictions. The edges in gray participate in other restrictions, generated from other nodes not shown in Figure 4. Those edges also overlap with themselves, and even with the purple and red edges from Figure 4.

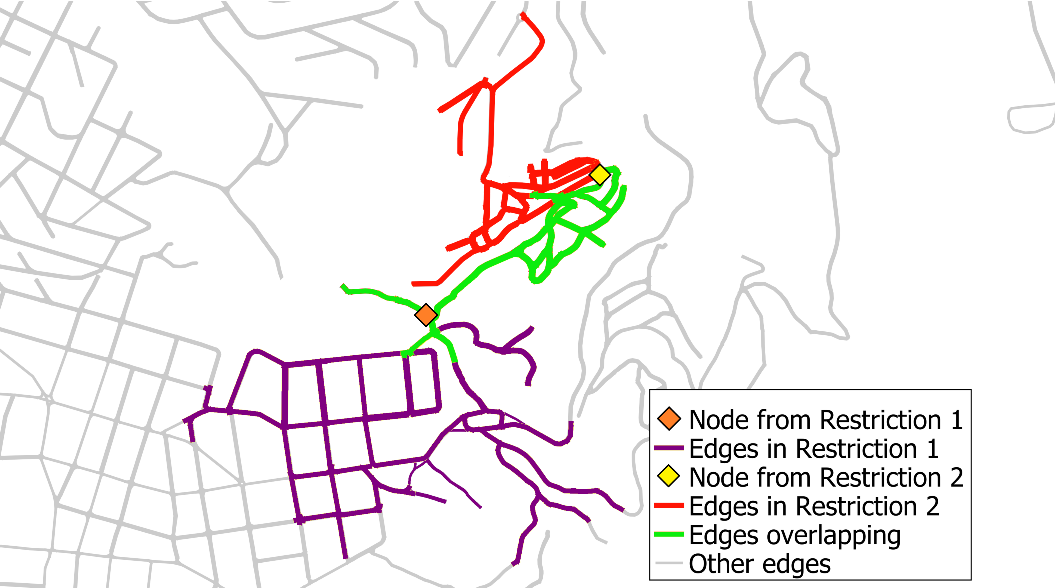


Fig. 4. Restrictions illustration from the Integer Programming Method

Vector b holds the maximum number of NPDPs allowed for each node's surrounding area. For most of the restrictions, b equals 1. There are some exceptions, because some edges are longer than the threshold d. In that case, the value of b increases proportionally to suit the long edges. For example, if the threshold is $d = 400$ meters but close to a node there is a 2,000-meters-long edge, b will increase from 1 to $2000 \div 400 = 5$. For the cases when the division has a remainder different than 0, b is set to the greatest integer below the division's result. In other words, b is the floor of the division between the edge's length and the d threshold. If a restriction has more edges longer than or equal to d, b is increased for all those edge lengths accordingly. Thus, a restriction r has the maximum number of NPDPs b_r according to the Inequality 2. The already existing bus stops are set in the model by the restrictions defined in Equation 3.

$$b_r \leq \begin{cases} max(1, \sum\limits_{e \in \mathbb{S}_r} n_e), & \text{if } l_e \leq d \\ max(\sum\limits_{e \in \mathbb{S}_r} \lfloor l_e \div d \rfloor, \sum\limits_{e \in \mathbb{S}_r} n_e), & \text{if } l_e > d \end{cases} \tag{2}$$

$$b_e = r_e \quad \forall e \in \mathbb{E} : n_e \geq 1 \tag{3}$$

Where n_e is the number of existing bus stops in e, e is an edge $\in \mathbb{E}$, $\mathbb{E}$ is the set of all edges, l_e is e's length, $\mathbb{S}$ is a subset $\subset \mathbb{E}$, and $\mathbb{S}_r$ indicates the subset of edges that participate in the restriction r. For each line in A, thus for each restriction, the selected edges have 1 in their respective columns, otherwise the value is 0. Regarding the restrictions that are inequalities on the number of NPDPs, equation $Ax = b$ works as if it were $Ax \leq b$. Those inequalities can be converted to equalities by adding one variable for each inequality, and consequently adding one more column in A and c, for each existing inequality restriction [Wagner 1958]. Next Subsection discusses the computational challenges to solve this combinatorial optimization problem.

5.2 Computational Challenges

This Subsection presents computational challenges to solve the NPDPs location using integer programming, and how these challenges were met. The challenges can be summarized into memory usage and processing time. These challenges are described, respectively, in the following Subsections.

5.2.1 *Memory Issues.* A considerable issue on optimizing the NPDP locations using the Simplex method is memory usage. For the entire city of Belo Horizonte, matrix A is composed by 146,505 lines × 231,535 columns, and thus would hold 33,921,035,175 elements. Even not considering the additional columns to convert the inequalities to equalities, and considering each element to use as little as one byte, almost 34 gigabytes of main memory would be required just to store the elements from matrix A.

To avoid this issue while building matrix A, only the nonzero elements are stored in a data structure formed by a vector of lists. That approach is well suited for this memory issue because even though the distance thresholds d considered are reasonable for walking distances, they cover only small fractions of the whole city. In other words, since an NPDP location is only influenced by some restrictions composed by the surrounding set of edges (limited by the distance threshold $d = 200$, $d = 400$ or $d = 800$ meters), edges that are farther away do not need to be stored for that restriction because they will not influence the optimization at all. Thus, instead of holding 231,535 columns, the average number of edges per restriction drops to 16, 74 and 363 edges for $d = 200, 400$ and 800 meters, respectively. The higher the number of edges used, the greater the threshold d is expected to be, since a greater distance threshold selects wider and more extensive sets of edges for the restrictions.

Another way to reduce the need for memory is to eliminate repeated restrictions. Once each restriction is defined by selecting edges that start from a node, nodes close to each other can eventually generate the same set of edges, even if in a different order. In this case, one of those restrictions can be eliminated, since both restrictions provide the same semantic for this optimization problem. There are other situations where a restriction includes all the edges from another restriction, and other edges beside those. That situation can happen depending on the city's geography, where nodes with few outgoing edges (on a corner, for example) can reach only a subset of a bigger set of edges reached by a nearby node. In this case, the restriction to be eliminated must be the one with fewer edges, since the more complete restriction already holds all semantics provided by the less complete one.

Therefore, the elimination of repeated restrictions was performed when the Relation 4 is satisfied.

$$\mathbb{S}_r \subseteq \mathbb{S}_v : \forall r, v \in \mathbb{T} \wedge \mathbb{S} \subset \mathbb{E} \tag{4}$$

Where $\mathbb{T}$ is the set of all restrictions generated. This procedure is performed only once, right before starting the optimization for each value of d. Assuming the restrictions set is sorted by the number of edges in each restriction, the elimination of repeated restrictions has a time complexity of $O(|\mathbb{T}| * |\mathbb{S}_r|)$, where $|\mathbb{T}|$ is the number of restrictions and $|\mathbb{S}_r|$ is the number of edges in the restriction r. That time complexity represents the worst case, which happens when only the last edge $\in \mathbb{S}_v$ differs from the edges $\in \mathbb{S}_r$. Thus, it is only known if $\mathbb{S}_r \subseteq \mathbb{S}_v$ after comparing all the edges $\in \mathbb{S}_r$ to the edges $\in \mathbb{S}_v$.

After the elimination of repeated restrictions, the number of restrictions fell from 146,505 to 113,482 for $d = 200$ meters, a reduction of 22.5%. Using $d = 400$ meters, the number of restrictions fell to 137,848, a reduction of 5.9%. And for $d = 800$ meters, the number of restrictions fell to 144,919, a reduction of 1.1%. The elimination of repeated restrictions has a more effective impact with the lowest threshold d. That happens because the lower the d threshold, the fewer number of edges $|\mathbb{S}_r|$ that participate in a restriction. With fewer $|\mathbb{S}_r|$, less diverse $\mathbb{S}_r$ are generated. With less diverse $\mathbb{S}_r$, and consequently more similar sets of $\mathbb{S}_r$, the rule expressed by the Relation 4 is more often satisfied.

Although the memory size would still be an issue for larger instances and greater d values, those two approaches to reduce the needed memory enabled to start the optimization for Belo Horizonte without raising memory errors. However, 8 Gigabytes of main memory were not enough to hold the optimization without paging between the main and the secondary memories. Since the optimization time becomes impracticable when the system performs paging, other approaches were applied to overcome this issue. These approaches are presented in the following Subsection.

5.2.2 *Processing Time Issues.* The approaches applied to the processing time issue comprised dividing the optimization area and including additional restrictions. The system started paging because the optimization problem became too big, while optimizing the whole city at once. The approach applied to solve that was to optimize each neighborhood separately and then put together the solutions found. By doing that, another issue appears: the number of NPDPs suggested for the same edge intersecting multiple neighborhoods can be different for each neighborhood's optimization. Therefore, this approach brings two drawbacks: i) optimal solutions are only guaranteed for each neighborhood separately, not for the set of neighborhood solutions; ii) the solution made by joining the neighborhoods can be unfeasible due to possible NPDP that are suggested close to the neighborhood borders.

Trying to overcome both drawbacks, the number of suggested NPDPs for an edge on the border of each neighborhood was the smallest one for that edge among the performed neighborhoods' optimization. Even with this approach, the time needed for one neighborhood to be optimized varied from less than a minute to some days. That variation depended on the neighborhood's road network extension and on the d value. To speed up the optimization process, mainly regarding big neighborhoods and $d = 200$ meters, other restrictions were included on the matrix A to reduce the number of possible solutions. Similar approaches are common in linear optimization, focusing on pruning the search tree in order to avoid unnecessary or semantically repeated steps [Margot 2010].

The additional restrictions are about edges with extensive length. Since the NPDPs are located on edges and d is the distance threshold between NPDPs, edges that are longer than d are more likely to have at least one NPDP. The cases when such edges have no NPDP occur when other nearby edges have NPDPs, and the optimal solution used e only as part of the spacing between those NPDPs. However, when l_e is greater than $2 \times d$, the edge e in an optimal solution will have at least one NPDP. Besides, when e is even longer, such as $l_e > 3 \times d$, there is enough space to contain segments that are d meters long within e. This pattern, which uses an integer multiplying d, can be applied to define a lower bound to the number of NPDPs for an edge. Therefore, the number of NPDPs on the edge e with $l_e \geq d$ has its lower bound defined according to its length and the distance threshold d, as shown by Inequality 5.

$$n_e \geq \lfloor l_e \div d \rfloor - 1 \quad : \forall e \in \mathbb{E} \land l_e \geq d \tag{5}$$

Those lower bounds were set as additional restrictions to speed up the optimization process. As these additional restrictions try to increase the number of NPDPs, instead of limiting them, the optimal solution was not disturbed by them. That method allowed us to optimize all neighborhoods using $d = 800$ meters in a viable time. However, for one neighborhood using $d = 400$ meters and for 40 neighborhoods using $d = 200$ meters we were unable to assure the optimality in a reasonable time.

The single neighborhood with no optimal solution for $d = 400$ meters is named "Jardim dos Comerciários". After about 2,000 iterations from the optimization method, a solution with 88 NPDPs was found for that neighborhood. Besides being the best solution found so far, the Branch-and-Cut algorithm also holds a upper bound threshold obtained from the previously explored nodes from the optimization search tree. After about 20 million iterations, the upper bound was updated to 89 NPDPs. Therefore, the optimal solution for that neighborhood could be 88 (guaranteed) or (maybe, at best) 89 NPDPs. Both the best solution found and the upper bound were maintained even after about 95 million iterations (five days and a half of computations, and the optimization was not finished yet). In a nutshell, the best solution found on the first 2,000 iterations did not change after 95 million iterations, and that one can also be the optimal solution for that neighborhood.

To finish the optimization process in a viable time, we established a limit of 100,000 iterations for the neighborhood with $d = 400$ meters and the 40 neighborhoods with $d = 200$ meters. Thus, the solutions used in this work for those cases are the best solutions found up the 100,000th iteration. Next section presents the results, including a comparison of the solutions obtained using the Random Walk and the Grid-Based implementations, and an analysis of the Integer Programming Method applied to the NPDP location problem.

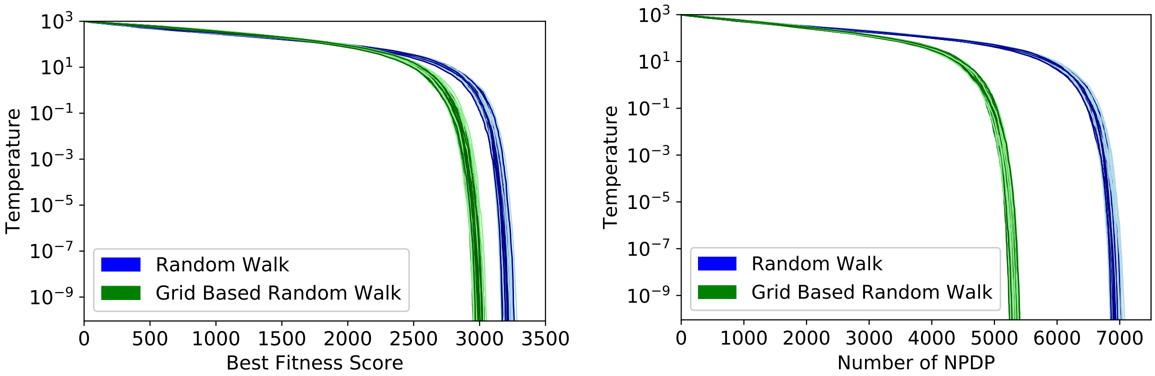
6. RESULTS

This section presents the results found by the optimization processes. The results regarding the meta-heuristic optimization methods are presented in Subsection 6.1, and the results about the Integer Programming method are presented in Subsection 6.2.

6.1 Meta-heuristic Methods

Empirical tests were used to define a set of parameters that enable the methods to converge. The defined parameters were: initial temperature = 1,000; iteration number for each temperature = 1,000; temperature decreasing rate = 0.9; and minimum temperature = 10^{-10}.

Figure 5 shows the convergence curve after 40 executions of the Random Walk and Grid-Based Random Walk optimization methods. Figure 5 (a) presents the evolution of the solution with the best fitness score in each execution of both optimization methods. Figure 5 (b) presents the number of NPDPs suggested. The "Temperature" axis of both graphs is in logarithmic scale. The curve in both graphs shows saturation when the temperature reaches a value of about 10. After this temperature, the fitness score and the number of NPDPs keep increasing, but more slowly.



(a) Best fitness score

(b) Number of NPDPs

Fig. 5. Optimization's convergence

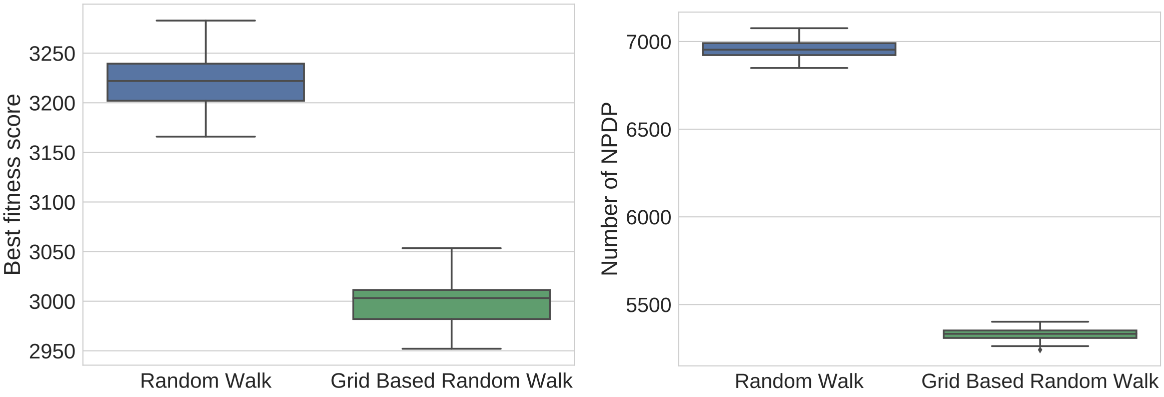
The Random Walk method presented in Section 4.1 achieved better fitness scores. The Grid-Based Random Walk method achieved fitness scores close to those from the Random Walk method (only 7% of difference between the best results from both) but using many fewer NPDPs (23.8% less). This large difference for the number of NPDPs is due to the location constraint imposed by the grid, in which the suggested NPDP must coincide with a point in the grid. This suggests that the Grid-Based method may be suitable for multi-objective versions of this optimization problem, in which the goals are to maximize the reach of NPDPs and minimize the number of NPDPs required.

Table I presents the basic statistics of both optimization methods. The Grid-Based Random Walk had a lower standard deviation for the fitness scores and number of NPDPs. This lower variation is probably due to the grid-imposed reduction of candidate locations for NPDPs. The values varied linearly from the lowest to the highest. The absence of strong variations among the 40 executions of each method indicates that the execution samples generalize well the generated solutions.

Table I. Statistics of the proposed meta-heuristic optimization methods

	Best fitness score		Number of NPDPs	
Measures	Random Walk	Grid-Based R.W.	Random Walk	Grid-Based R.W.
Lowest	3,165.92	2,952.09	6,849.00	5,242.00
1st Quartile	3,202.07	2,982.10	6,922.50	5,309.50
Average	3,221.12	2,999.33	6,951.93	5,332.03
Median	3,221.95	3,003.16	6,953.50	5,333.00
3rd Quartile	3,239.43	3,011.36	6,990.75	5,352.25
Highest	3,282.84	3,053.45	7,076.00	5,402.00
Std. deviation	25.93	23.90	53.34	36.00

Figure 6 compares the boxplots from the resultspresented in Table I. Figure 6 (a) presents the best fitness scores achieved and Figure 6 (b) shows the number of NPDPs suggested. In each figure, the boxes from the boxplots do not overlap. This indicates that there is significant statistical difference between the results [Krzywinski and Altman 2014]. Therefore, it can be asserted that the Random Walk method achieved better fitness scores than the Grid-Based Random Walk, and that the Grid-Based Random Walk generates fewer NPDPs, as compared to the Random Walk method.



(a) Best fitness scores

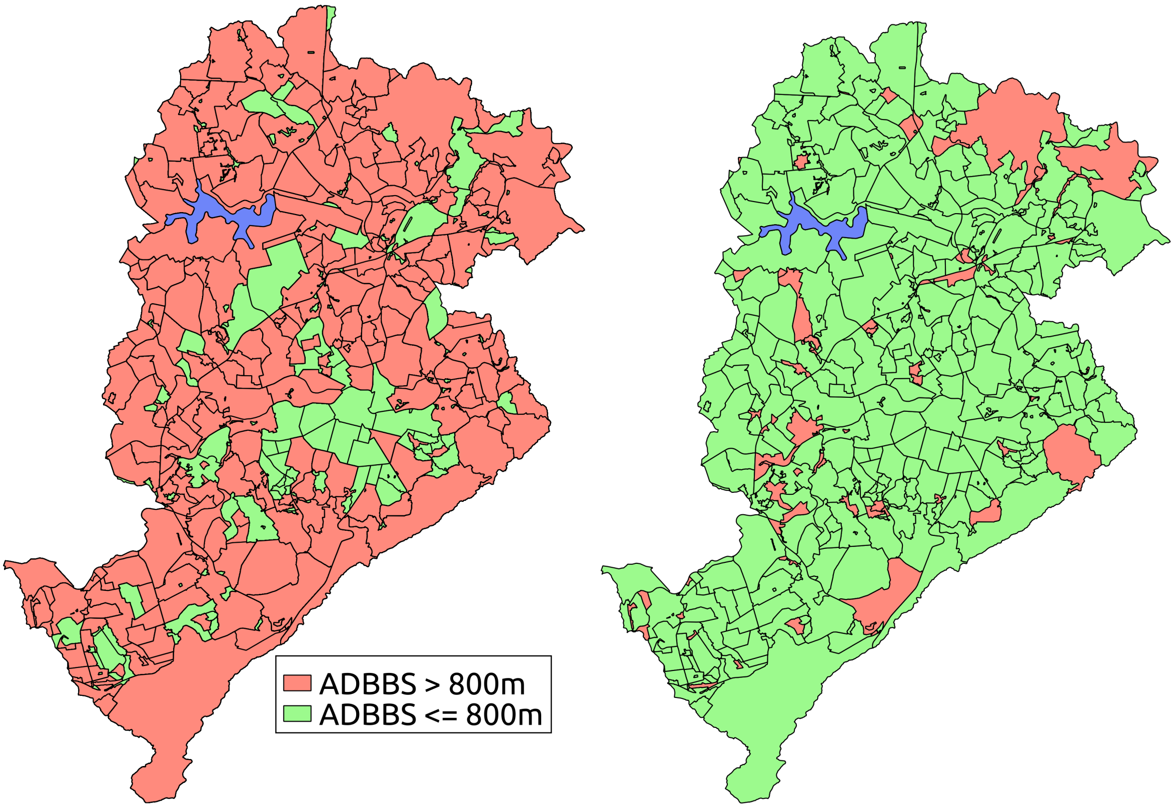
(b) Number of NPDPs

Fig. 6. Boxplots of the optimization methods

Figure 7 illustrates the optimized Belo Horizonte neighborhoods using the best solution found. This solution, which is analyzed next, is based on the solution with a fitness score of 3,282.84, found by the Random Walk method. Figure 7 (a) presents the neighborhoods selected to be optimized, as described in the Section 3.2. The red neighborhoods were used in the optimization process, and the green neighborhoods already have one bus stop for every 800 meters of streets or less. This measure is represented in the figure as Average Distance Between Bus Stops (ADBBS). After the optimization process, in 71 of the 299 neighborhoods the ADBBS stayed above 800 meters. All these 71 can have the ADBBS $\leq$ 800 meters if the parameters from the Simulated Annealing were changed. The main parameter that could be able to reduce the ADBBS even more would be the minimum temperature, but the processing time would also be greater.

Together, these 71 neighborhoods cover an area of 37 km^2, the equivalent to 11.16% from the 331.4 km^2 of Belo Horizonte. Nevertheless, the total length of streets in these 71 neighborhoods represents only 6.5% of Belo Horizonte's network. The lower street length in these 71 neighborhoods hampered the access and generation of NPDPs by the Random Walk.

In spite of this limitation, some of these regions with ADBBS > 800 meters are slums, whose typically narrow streets may even be unfit for bus traffic. The solution with the best fitness score would provide to the public transportation of Belo Horizonte the contributions listed on Table II. Therefore, considering that the NPDPs can work as bus stops (for example), the creation of 7,076 NPDPs (equivalent to 75.1% of the existent bus stops) would reduce the average walking distance to reach a bus stop, improving the accessibility of 76.3% of the neighborhoods to a regular level, as defined by Veras et al. [2016] and Nalawade et al. [2016]. In relation to the whole city, these NPDPs would reduce the ADBBS in 39%. However, in relation to the street length from neighborhoods that have ADBBS > 800 meters, there was a reduction of 92.8%. Next Subsection presents the Integer Programming results.



(a) Without NPDPs

(b) With NPDPs

Fig. 7. Accessibility impact of the NPDPs on Belo Horizonte's neighborhoods

Table II. Impact of the Random Walk Method

Measures	Original	With R.W.	Variation
Number of bus stops + NPDPs	9,428	16,504	+75.1%
Average distance for bus stop or NPDP	959.57 m	585.59 m	-39.0%
Number of neighborhoods to be improved	299	71	-76.3%
Total area neighborhoods to be improved	280.1 km^2	37 km^2	-86.8%
Street length to be improved	8,195 km	587.9 km	-92.8%

6.2 Integer Programming Method

Table III presents the results from the proposed Integer Programming method. The number of NPDPs generated and, consequently, the average distance between NPDPs were close to the original metrics when $d = 800$ meters. The larger variation for $d = 800$ meters regards the number of neighborhoods to be improved. Also, although the number of NPDPs was close to the number of bus stops, the number of neighborhoods with ADBBS ≤ 800 meters decreased by almost one third. These results indicate that the Integer Programming method tends to span regularly the NPDPs through the city, which is especially beneficial to areas far from downtown. The most similar metrics between the Integer Programming method and the Random Walk method occurred with $d = 400$ meters. The most different metrics between the two methods was about the street length to be improved. That happened because, as the Integer Programming generates NPDPs regularly spaced among all directions, neighborhoods with more streets are not more likely to have an NPDP, as the Random Walk method determines.

Table III. Impact of Integer Programming Method

		d values					
Measures	Original	200 m	Variation	400 m	Variation	800 m	Variation
Number of NPDPs	9,428	35,476	+276.28%	14,921	+58.26%	9,974	+5.79%
Avg. distance for NPDP	959.57 m	231.95 m	-75.83%	583.72 m	-39.17%	897.12 m	-6.51%
Neighb. to be improved	299	2	-99.33%	79	-73.57%	203	-32.11%
Area to be improved	280.1 km^2	0.06 km^2	-99.98%	98.95 km^2	-64.67%	232.56 km^2	-16.97%
Street length to be Impr.	8,195 km	3.1 km	-99.96%	3,623 km	-55.79%	7,623 km	-6.98%

The best metrics were obtained with Integer Programming and $d = 200$ meters. However, the number of NPDPs generated for that configuration was more than 3.5 times the number of existing bus stops. Even though a high number of NPDPs is needed in regions such as downtown, peripheral regions may not have enough demand to effectively use all the suggested NPDPs. In that scenario, a greater d value or optimization by the Grid-Based Random Walk would be better suited. The proposed optimization methods achieved their goal to optimize the locations for New Pick-up and Drop-off Points. All the three proposed methods can improve the accessibility of the public transportation by their different processes and benefits.

7. CONCLUSION

Optimization algorithms for public transportation compose a recent and promising target of research. This work presented three methods to optimize the location of NPDPs, each method with its different process and benefits. The Random Walk optimization method achieved better fitness scores than the Grid-Based Random Walk. However, the Grid-Based Random Walk reached fitness scores close to the ones obtained by the Random Walk method, but using many fewer NPDPs. This indicates that the Grid-Based Random Walk can be attractive for a multi-objective version of this problem. Although the Random Walk has been effective in suggesting NPDPs for Belo Horizonte, its dependency on following the road network hampered the access to neighborhoods with fewer registered streets. This issue was overcome by the Integer Programming method by suggesting regularly spaced NPDPs.

The three methods were developed with a concern on scalability issues. The methods based in Simulated Annealing performed the optimization without the need to keep in memory a population to evolve, as in Genetic Algorithms. The Integer Programming method used techniques to eliminate unnecessary values from memory, and to speed up the process by optimizing the neighborhoods separately, and by adding new constraints. These approaches were essential to make the optimization process viable. Although the methods were effective on defining the location of NPDPs, neither of the three methods guarantee global optimality for the whole city of Belo Horizonte.

As future work, we plan to evaluate other formulations for the Integer Programming method in order to improve the performance and scalability for the optimization process. Possible improvements include standardizing the edge lengths and decomposing the restrictions set in subproblems. Another future work consists in using a measure based on the local average of demographic density or door's address, instead of fixed thresholds such as ADBBS and d. We also propose implementing a multi-objective version with a second goal to also minimize the number of NPDPs defined.

REFERENCES

Baaj, M. H. and Mahmassani, H. S. An AI-Based Approach for Transit Route System Planning and Design. *Journal of advanced transportation* 25 (2): 187–209, 1991.

Diamond, S. and Boyd, S. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *The Journal of Machine Learning Research* 17 (1): 2909–2913, 2016.

Garrides, M. G. M., Souza, P. C., and Campos Neto, L. S. Transporte Público em Belo Horizonte: um estudo comparativo entre Metrô e Monotrilho. *Revista Petra* 2 (1): 1–16, 2016.

Jerônimo, C. L. M., Campelo, C. E., and de Souza Baptista, C. Using Open Data to Analyze Urban Mobility from Social Networks. *Journal of Information and Data Management* 8 (1): 83–99, 2017.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. Optimization by Simulated Annealing. *Science* 220 (4598): 671–680, 1983.

Kostina, E. The Long Step Rule in the Bounded-Variable Dual Simplex Method: Numerical Experiments. *Mathematical Methods of Operations Research* 55 (3): 413–429, 2002.

Krzywinski, M. and Altman, N. Points of Significance: visualizing samples with box plots. *Nature Methods* 11 (2): 119–120, 2014.

Loader, C. and Stanley, J. Growing Bus Patronage and Addressing Transport Disadvantage—The Melbourne experience. *Transport Policy* 16 (3): 106–114, 2009.

Logiodice, P., Arbex, R., Tomasiello, D., and Giannotti, M. A. Spatial Visualization of Job Inaccessibility to Identify Transport Related Social Exclusion. In *XVI Brazilian Symposium on GeoInformatics (GEOINFO)*. Campos do Jordão, Brazil, pp. 105–118, 2015.

Margot, F. Symmetry in Integer Linear Programming. In *50 Years of Integer Programming 1958-2008*. Springer, pp. 647–686, 2010.

Monteiro, C. M., Martins, F. V. C., and Davis Jr, C. A. Optimization of New Pick-up and Drop-off Points for Public Transportation. In *XVIII Brazilian Symposium on GeoInformatics (GEOINFO)*. Salvador, Brazil, pp. 222–233, 2017a.

Monteiro, C. M., Silva, F. R., and Murta, C. D. Análise de Padrões Espaciais e Temporais da Mobilidade de Táxis em San Francisco e Roma. In *43o. Seminário Integrado de Software e Hardware (SEMISH)*. Porto Alegre, Brazil, pp. 1736–1747, 2016.

Monteiro, C. M., Silva, F. R., and Murta, C. D. Pré-processamento e Análise de Dados de Táxis. In *44o. Seminário Integrado de Software e Hardware (SEMISH)*. São Paulo, Brazil, pp. 2610–2621, 2017b.

Nalawade, D. B., Nagne, A. D., Dhumal, R. K., and Kale, K. Multilevel Framework for Optimizing Bus Stop Spacing. *IJRET: International Journal of Research in Engineering and Technology* vol. 5, pp. 298–304, 2016.

Oliveira, A., Souza, M., Pereira, M. A., Reis, F. A. L., Almeida, P. E. M., Silva, E. J., and Crepalde, D. S. Optimization of Taxi Cabs Assignment in Geographical Location-based Systems. In *XVI Brazilian Symposium on GeoInformatics (GEOINFO)*. pp. 92–104, 2015.

Santos, S. R. d., Davis Jr, C. A., and Smarzaro, R. Analyzing Traffic Accidents based on the Integration of Official and Crowdsourced Data. *Journal of Information and Data Management* 8 (1): 67–82, 2017.

Silva Júnior, A. M., Sousa, M. L., Xavier, F. Z., Xavier, W. Z., Almeida, J. M., Ziviani, A., Rangel, F., Avila, C., and Marques-Neto, H. T. Caracterização do Serviço de Táxi a partir de Corridas Solicitadas por um Aplicativo de Smartphone. In *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. Salvador, Brazil, pp. 17–30, 2016.

Takakura, M., Furuta, T., and Tanaka, M. S. Urban Bus Network Design Using Genetic Algorithm and Map Information. In *Proceedings of the Eastern Asia Society for Transportation Studies*. Cebu, Philippines, pp. 1–13, 2015.

Veras, D., Pinto, G., Lobo, C., Cardoso, L., and Garcia, R. Acessibilidade Urbana em Belo Horizonte: apontamentos sobre a acessibilidade aos serviços do transporte coletivo municipal. In *7o. Congresso Luso Brasileiro para o Planejamento, Urbano, Regional, Integrado e Sustentável (Pluris)*. Maceió, Brazil, pp. 1–12, 2016.

Wagner, H. M. The Dual Simplex Algorithm for Bounded Variables. *Naval Research Logistics (NRL)* 5 (3): 257–261, 1958.

Yao, B., Cao, Q., Jin, L., Zhang, M., and Zhao, Y. Circle Line Optimization of Shuttle Bus in Central Business District without Transit Hub. *PROMET-Traffic&Transportation* 29 (1): 45–55, 2017.