

A Method for the Optimized Placement of Bus Stops Based on Voronoi Diagrams

Timo Stadler
OTH Regensburg
timo.stadler@oth-regensburg.de

Simon Hofmeister
OTH Regensburg
simon1.hofmeister@st.oth-regensburg.de

Jan Dünnweber
OTH Regensburg
jan.duennweber@oth-regensburg.de

Abstract

In this paper, a new method for placing bus stops is presented. The method is suitable for permanently installed new bus stops and temporarily chosen collection points for call busses as well. Moreover, our implementation of the Voronoi algorithm chooses new locations for bus stops in such a way that more bus stops are set in densely populated areas and less in less populated areas. To achieve this goal, a corresponding weighting is applied to each possible placement point, based on the number of inhabitants around this point and the points of interest, such as medical centers and department stores around this point. Using the area of Roding, a small town in Bavaria, for a case study, we show that our method is especially suitable for rural areas, where there are few multi-family houses or apartment blocks and the area is not densely populated.

1. Introduction

When we think of the future of transportation and, especially, sustainable transport, public transport is one of the most important means of transporting people. It facilitates the transportation of a large number of people at the same time and thus prevents traffic congestion and reduces the carbon footprint. However, this type of transport also requires an appropriate infrastructure in the form of suitably placed bus stops. The location of the bus stops is a crucial factor for an acceptance of public transport. Problems can arise if, for example, a bus stop is often overcrowded or only rarely used. Choosing locations for new bus stops is a challenging task, especially, if the placement should be automatized and the dynamic relocation of bus stops should be supported for call busses. There are many individual questions that need to be answered when a new location is chosen. These questions are: Can a new bus stop be integrated into an existing route? Is a bus stop used by enough people? Are bus stops also available for people

who live far away from the city center?

It may happen that individual neighborhoods are not taken into account when planning bus routes and thus people have no or only very difficult connections to the public transport system. This occurs more and more often in the rural areas we look at in our paper, especially due to the rapid urbanization of rural areas.

The recognition of spatial clusters can lead to biased results, since these models are oriented e.g. on the road network and thus deviate from what a set of freely located points can represent. Since we want to consider people in the very rural Roding who live away from the road network or along dirt roads, which are not shown in the map material, we did not use a network Voronoi diagram [1]. In our model, we consider the surrounding area as a planar surface with Euclidean distances. This is also possible, because the city of Roding has relatively low elevation differences, and the inclusion of Inclusion of slopes and subsidences as in a network Voronoi diagram is not necessary. We want to improve the situation using an automatized bus stop location recommendation algorithm, based on computational geometry. The underlying principle is the Voronoi Diagram, also known as Dirichlet Tesselation, which a computer can find, such that a set of given locations (in our case, the existing bus stops) is divided into polygons, each containing one of the locations and all the points nearer to that location than to any other [2]. The concept of Voronoi diagrams was introduced as early as 1908 and is named after its inventor M.G. Voronoi [3]. It is known, that the vertices of the Voronoi polygons for a set of locations are good candidates for new locations since one of them is the center of the largest empty circle [4]. However, a computer program that is only processing geometric features and does not take the distribution of the population into account, will not improve the situation of the people in rural areas. In the countryside, the largest area without a bus stop may be an agrarian field, while, in a densely populated village that already has one bus stop, a second one might be urgently required. For taking the geometric

distribution of bus stops and, as well, the population density into consideration, the software we present in this work uses open source geographic information system (GIS) data. The 1224 bus stops of a bus company from Roding are used for a case study in this paper as a test data set. For each stop, the longitude, latitude, as well as the name and the route affiliation are already recorded in the data set.

Roding is a small town with an area of 113 km^2 and about 11,500 inhabitants (as of 2008) in Bavaria, Germany. However, the area with all stops also includes the neighboring counties and is with 674 km^2 almost 6 times as large as the town itself. Therefore, Roding is a representative example of a rural region with many widely distributed settlement cores. For the city of Roding, we have already carried out a passenger demand prediction showing that some busses are only very lightly utilized and could accommodate significantly more passengers [5]. However, the infrastructure with the appropriate stops must first be created.

In topics working with GIS data, such as the extraction of a topology Voronoi diagrams have already been used [6]. Therefore, it seemed to make sense to use Voronoi diagrams for our task, which is concerned with keeping the distances between bus stops as regular as possible.

To extend the original location set, we divide the polygons into sub polygons comprising new locations, which are candidates for the new stops. Before we extend the location set by separating a polygon in the Voronoi diagram into sub-polygons, we compute the corresponding weight of the polygon using the OpenStreetMap Service. We use this data together with the Overpass API to set the new starting point directly at a road [7].

In Section 2, we discuss related work to our method and important points of reference for public transport in rural areas. Section 3 explains the problems that occur and have to be considered when placing bus stops. Our method for placing bus stops is described in Section 4. Section 5 deals with the evaluation of the method, compares it to other methods and Section 6 gives an outlook on planned improvements of public transport in rural areas based on our method.

2. Related work

We divide our discussion of related work into an overview of technical implementations as well as other methods for setting bus stops. The second section deals with the special conditions and developments in rural areas, which are considered in our paper.

2.1. Voronoi diagrams in traffic planning

The advantage of Voronoi diagrams over other methods such as a visibility graph with a runtime of $\mathcal{O}(n^2)$ is the shorter runtime of $\mathcal{O}(n \log n)$ [8]. Therefore, the algorithm achieves a much shorter runtime than others for large data sets. For example, Bhattacharya et al. employ an algorithm based on Voronoi diagrams to find the shortest path between two points in a 2D obstacle course [8]. Lebedeva et al. applied the standard Voronoi diagrams to optimize freight delivery from logistics centers. Here, the focus of optimization is on econometric factors such as gasoline consumption and the average distance a package travels, thus achieving lower utilization of the road system [9]. In contrast to this approach, our method aims to achieve the highest possible coverage of the inhabited area by weighting the factors from the GIS data, and thus to be able to persuade people to switch to public transport.

Two more approaches are listed here, which use computer-based methods to set bus stops. Ghasedi et al. model the population of the country through the heuristic optimization technique of particle swarm optimization (PSO) and thereby infer the bus stops that need to be set. [10]. Another approach for optimizing the placement of bus stops is presented by Bargegol et al. They use a genetic algorithm to model a corridor along a bus route with bus stops [11]. However, both of these methods are more focused on finding the shortest travel time, with the least amount of time lost during boarding and alighting. Our goal is rather to cover the complete traffic space as efficiently as possible with a fixed number of stops. We expect that this will lead to more people choosing public transport as a means of transport and reduce emissions, even though the buses cover proportionately longer distances because there are fewer vehicles on the roads. One approach to using Voronoi diagrams to characterize landscapes is by Lan Mu. He uses multiplicatively weighted Voronoi diagrams to reflect both the underlying process and the shape of the landscape polygons. This creates a metric that can measure a generator point's region of influence based on its weight [12]. Lu's goal was better visualization of results, while in this paper weighting is done to optimize placement of points.

2.2. Rural transportation

In their paper, Brovarone and Cotella suggest that improving public transportation is highly dependent on a wide variety of socio-economic and cultural factors. They created a layer model, wherein each deeper layer more concrete solutions are addressed. With

our approach, we contribute to a solution in the top layer, which contains tools for planning public transport [13]. Based on the top-layer knowledge, more concrete solutions can be generated in the deeper layers, such as a dynamic bus schedule.

Bosworth et al. suggested in their survey that public transportation expansion is the most desired need among rural residents. (Far ahead of broadband or mobile network expansion). They also address Mobility as a Service (MaaS) solutions in their paper, pointing out that most of these approaches are designed for urban areas. In rural areas, the lack of infrastructure means that proven technologies, such as buses, must be used. Another interesting approach to optimizing transportation in rural areas is presented by Zhan et al. They assume that the demand for buses varies greatly throughout the day and present a method that dynamically lengthens or shortens the route of a bus to optimize it [14]. This method can be combined with our approach to optimize the routes to the much larger number of stops. The fact that all stops are served by the buses only at certain times can lead to a better utilization of the core area and lower emissions.

3. The problem of bus stop placement in rural areas

The main reason for the use of public transport by the population is commuting from rural areas to the city center and back again. Our goal is to solve the so-called first-mile problem by cleverly placing bus stops, as this significantly contributes to better acceptance of public transport [15]. That is, to reduce the distance individuals have to travel to reach the bus stop. Also, the total travel time of all passengers shall be reduced. We're trying to solve this problem by putting more stops closer to residents' homes.

The problem with setting bus stops in rural areas is that there are two completely different types of villages. One is large urban cores that are already equipped with enough bus stops. The second is individual villages with very few, or no, bus stops.

The generation of Voronoi diagrams and the dual Delaunay triangulation is known to be a possible way for solving the so-called largest empty circle problem. Here the largest empty circle is always located at a vertex of the Voronoi diagram or at an intersection between a Voronoi edge and the convex hull of the chosen set of points [16]. Based on the found largest circle it can be determined, which point in the whole diagram is farthest away from all previous starting points that are chosen for the creation of the Voronoi

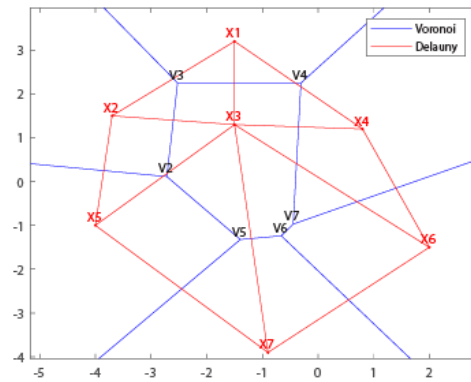


Figure 1. Representation of start points and vertices of Voronoi diagrams and Delaunay triangulation

diagram. As starting points for the generation of the Voronoi diagram the already existing stops shall be used as starting points. All points, which sit at a corner point of the polygon generated from the starting point, are the starting points of the Delaunay triangulation due to the duality. This context can be seen in figure 1 where the points marked with v are the edges of the Voronoi polygons and the points marked with x are the starting points.

The method can be used especially well in two-dimensional space and location constraints in spatial databases for finding a point with the largest empty area around it [17]. However, the purely geometric method shows that when setting new points, location centers or places of interest were never hit, although it would logically make sense to have a bus stop there. When applying the method we realized that for rural areas not only the distance from other stops is decisive for optimal placement, but also other important factors such as settlement density or important facilities in the vicinity must also be taken into account when creating new stops. Therefore, we developed a procedure that, based on weighting divides the Voronoi diagram into sub-polygons in order to find optimal locations for new bus stops.

4. A new method for setting bus stops

The implementation of our method in the program that we implemented for our Roding case study is divided into several steps:

1. In the first step, the coordinate points of the existing stops are read in and supplemented by the information from the Overpass API.
2. In the next step, a weighting of each point is

done based on the information collected about that point.

3. After that, a Voronoi diagram is generated and a division of the polygons with the highest weight into sub-polygons is performed.
4. In the last step, the newly obtained coordinates are adapted to the road system and the points are checked for accessibility.

For the implementation of the method, the programming language Python was used, because it allows easy access to the Overpass API through appropriate pre-built libraries. For accessing the data from the OpenStreetMap dataset we also installed the Overpass-Turbo API on the computer that performs the method [7]. This is a data mining tool for OpenStreetMap that allows us to retrieve data from the online OSM database about all streets and houses in our test area of Roding.

As input, the program receives all bus stops in comma-separated values (CSV) format with the longitude and latitude of the coordinate point as attributes. In our case, we use the data of all existing bus stops in Roding, which we extracted by hand from the timetables as a test data set. If the method is to be performed for another area, such a CSV file has to be created anew. For each of these points, the number of houses at a distance of 100m and the number of points of interest at a distance of 500m are first retrieved using the Overpass API. POIs include, e.g., medical centers, department stores, however, our program does not distinguish types of POI during the extraction step. The values for the distance can be adjusted at any time in a config file of the Python script for calibration and adjustment of the algorithm. From these values the weight is determined with the following formula:

$$\begin{aligned} weight = nr_{houses} / (houses_{area} / 10)^2 * 10 \\ + nr_{attractions} / (attractions_{area} / 10)^2 * 100 \end{aligned} \quad (1)$$

The size of the area was included to make sure a very large search area will not automatically lead to a very high weighting. A Point of Interest (PoI) or attraction is more important for our weighting and is therefore multiplied by 100. The area was squared because the algorithm for generating the Voronoi diagram was overwhelmed by searching such a large area. The value 100 was fixed, because an even larger value would not have been sufficiently performant in terms of scaling. These PoI's are particularly useful for our representation because they represent large facilities to which a large number of people prefer to travel. We perform filtering,

which PoIs are also attractive for passengers. Examples of such PoIs are large company buildings where very many people work, hospitals, or retirement homes. The number of attractions is not divided by the underlying area, because it can happen that such an attraction takes a larger area and therefore cannot be completely recorded if it is divided by the area. So it is rather the presence and not the density of PoIs that matters here. In addition to the values for weighting, a minimum and maximum distance between bus stops can also be given to the algorithm, so that new optimized bus stops can be generated.

If all polygons were divided into smaller sub-polygons according to our chosen method, the number of stops in the centers would increase many times over, while only a few new bus stops would be placed in the rural regions [18]. In addition, for the division in sub-polygons, it is impossible to predict whether the new stops in the rural areas are really useful, or whether they are located in completely uninhabited areas and thus cannot be reached by any person. Thus, without weighting, a check would have to be made after each subdivision to determine whether the subdivision was reasonable, and this would greatly increase the runtime of the method.

Another problem in the case study of our selected small town Roding is that it is located close to the Czech border and the buses on the routes may not cross any national borders. Thus, all new points of the Voronoi diagram that lie outside Germany have to be discarded. In order to avoid the listed problems in the creation of new stops and to generate a redundancy-free public transport system, we modified the original Voronoi diagram algorithm and built our new method, such that Voronoi diagrams are created according to GIS data. The next step starts with the generation of the Voronoi diagrams.

As described earlier, Voronoi diagrams are characterized primarily by the fact that each point within a polygon is farther away from the starting point of another polygon than from that of its own. Let P be a set of n points and R a set of n points in R^2 , $n \geq 3$. All of these points are nonlinear.

For all these points, the distance between P_i and P_j is defined as any pair $(P_i, P_j) \in R$. Let $V(P_i)$ be the Voronoi region of P_i also be defined as the set of points with minimum distance to P_i . So, for each point inside the polygon the following formula applies:

$$\begin{aligned} V(P_i) = \{P \in R^2 | D(P, P_i) < D(P, P_j)\}, \\ (i, j = 1, 2, \dots, n) \end{aligned} \quad (2)$$

Moreover, (i, j) must hold. The Voronoi diagram defined by P is thus defined as:

$$V(P)\{V(P_1), V(P_2), \dots, V(P_n) = \bigcup_{i=1}^n \{V(P_i)\} \quad (3)$$

For our Roding case study, we let the Voronoi algorithm run on a test area of $674km^2$ to create 1000 new bus stops in parallel using the Pandas library [19].

For the implementation of the algorithm to create the Voronoi diagrams, the function 'spatial.Voronoi' of the Python library scipy is used [20]. This receives all input bus stops as starting points and returns an object that contains all Voronoi vertices, finite regions, and finite ridges in 2D representation. For our subdivision into subpolygons, the vertices are of particular importance. Splitting and merging in our method is done according to the Fortunes algorithm, so there is no possibility to specify the smallest size of a polygon. of a polygon. However, due to our weighting, it is extremely unlikely that polygons will be subdivided to such an extent that a subdivision is no longer useful. For this, many houses and PoIs would have to be located at the same time in a very small space, and in this case, it also makes sense to have to place several stops at such a place. Maybe it even makes sense to place a bus station at such a place. A special case is the boundary treatment of the considered map for the creation of the Voronoi diagram. Here, due to the chosen map section, the convex hull of all stops in R is assumed to be the bounding surface. Thus, there cannot be an infinite expansion, and points that are not in the catchment area of the buses are ignored. Alternatively, it can be specified through the configuration file of the algorithm, that a polygon can reach a maximum area and should stop growing after this area. Thus the limitation by the convex hull is ignored.

After the Voronoi diagram has been created, the

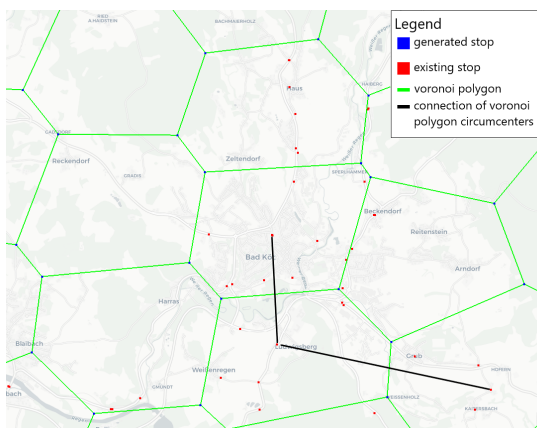


Figure 2. generated Voronoi diagram over the map

weighted setting of new bus stops can now take place.

For this purpose, the value *weight* from the Equation 1 is compared for each point and multiplied by the area of the corresponding polygon. Thus, on the one hand, very large areas are preferred for partitioning and, at the same time, polygons with a dense population or many points of interest are offered for placing new bus stops as well. Figure 2 shows the generated Voronoi diagrams superimposed on the map. The black lines drawn in represent how the subpolygons are created. These lines are not visible between all red points, because adjacent polygons with small areas can be merged. The completely generated stops are also already drawn in as red dots. Thus, it can be seen how a polygon will be divided into subpolygons. The procedure for splitting the map into subpolygons starts with the polygon that has the largest weight value. This is defined by an edge between two randomly chosen opposite points into two partial polygons. Each of these candidate polygons is assigned a new starting point, which is the geometric center of the polygon. Whether the candidate polygon is ultimately used for a new stop depends on the adaptation to the street system in the next step. Figure 3 shows a graphical representation of the weighting process for a potential new stop counts all stops from the adjacent polygons (orange circles) and the newly created stop (red circle) within a radius of 100m each and takes the number of houses in the intersection as a parameter for the weighting. This representation is independent of figure 2 and visualizes the decision making of the algorithm when creating a new stop Whenever a division takes place and the set of polygons grows, a new query is made for the number of houses within the polygon and the weighting is performed again. The algorithm ends when the specified number of iterations is reached. The number of specified iterations corresponds to the number of bus stops to be created. So for each iteration, a new stop is added. Besides weighting the locations, a classification into reachable and unreachable locations is necessary. This is performed in an additional step in our implementation. For each new candidate point, we check if it is located near a road. For this purpose, a nearest-neighbor search for the closest node is started using the Overpass API. A node is one of the core elements of the OpenStreetMap data model and contains a longitude and latitude value in addition to its ID. If the node should belong to a non-provided way type (highway or PoI) the search will continue until a suitable candidate is found. At the moment only roads cars can drive on are accepted, but no highways, since no boarding or alighting is allowed there. In the future, the restrictions on accepted nodes are more restrictive, such that a dead-end street where a bus cannot turn will not be accepted either. If no street is found within a specified

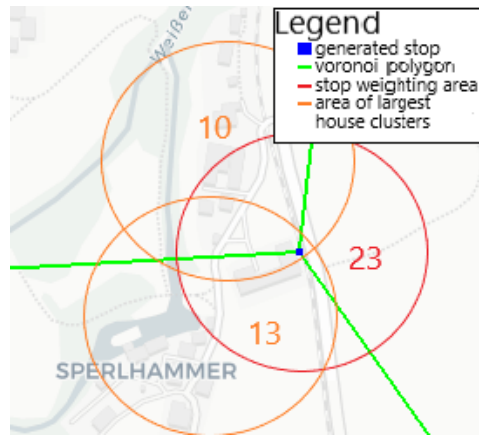


Figure 3. graphical representation of the weighting process

distance of 200m, the algorithm aborts, and the potential new point is discarded. This search can also cross the boundaries of the previously created polygons.

To get an overview of the generated stops as a person, GeoPlotLib is used [21] to render them on a map. With the help of this library, our rendering method receives as input our newly generated candidate points for bus stops and the corresponding map tiles from OpenStreetMap. Depending on the specified control parameters, the input bus stops can also be passed as additional parameters, so that a better comparison is possible. The output is done in an interactive UI or exported in *png* format depending on the user's requirements. In addition to the graphical representation two files, that help with debugging and can be processed directly by GIS Systems like the OpenRouteService [22], are also created.

Figure 4 shows all original stops as blue dots, and the newly placed ones as red dots. The image section covers the whole area of the small town of Roding and all neighboring towns and villages. The map has a scale of 50km. In our experiment, a value of 2200 new bus stops was passed to the algorithm, which should be set additionally in the area. In the examined point set, for 117 new points no street could be found in the vicinity and thus, no new stop.

5. Evaluation

In this chapter, an evaluation of the performance is done and some possible improvements and possibilities for parallelization are suggested. In addition, a comparison with other methods is made and the application area of our method is clearly delimited to these.

5.1. Runtime and memory evaluation

The tests we ran for generating the map with the newly defined stops were run on a computer with a Ryzen 9 5900HS CPU, 32GB RAM, and an RTX 3080 GPU. On this system, the average runtime of the algorithm was about 3.2 minutes. Our tests showed that API queries consumed most of that time. Since the OpenStreetMap data generally works with roads and POIs, there can be no prior exclusion of unusable areas to shorten the duration.

The procedure for retrieving the information of the individual points via the Overpass API is already parallelized with 16 jobs. Nevertheless, this process takes the most total time and is largely dependent on the speed of the Internet connection as well as the speed of the queried API. Saving the information locally the first time it is run, or loading it from a file, would completely avoid the long runtime, but it would also limit the flexibility of the algorithm. The storage of parameters can serve as a middle way. This way, the queries are only needed for the first run. If you want to run the algorithm again with a different number of output stops, the runtime is reduced significantly. A test in which all information was previously saved in a JSON representation showed that the step of data retrieval took a total of only about 2 seconds instead of the previous 2 minutes and 20 seconds. The JSON file had a file size of about 600 kilobytes.

The depiction in Figure 5 shows the shares of the runtime in the total time of the algorithm. From this diagram, it can be seen that the retrieval of the information for the stops takes most of the time. Only then follows the actual computationally intensive task of the Voronoi diagram.

As mentioned in Section 2 the algorithm has a runtime complexity of $\mathcal{O}(n \log(n))$ and a space complexity of $\mathcal{O}(n)$. Nevertheless, the RAM of the test computer is filled up due to a large amount of generated data and the 32GB were only just sufficient for the chosen use case with 2083 new stops.

After the method has been carried out, there are still some open tasks that could further improve the system. By recording the traffic volume and the main traffic direction, even better conclusions could be drawn about how useful a stop is. Work is already underway to integrate recorded traffic data into the system. Also, the impact of stops that are served by buses on different routes has not yet been taken into account. However, it can be assumed that there are significantly more passengers at these stops. We are currently conducting an investigation on these questions, using passenger counters in the busses.

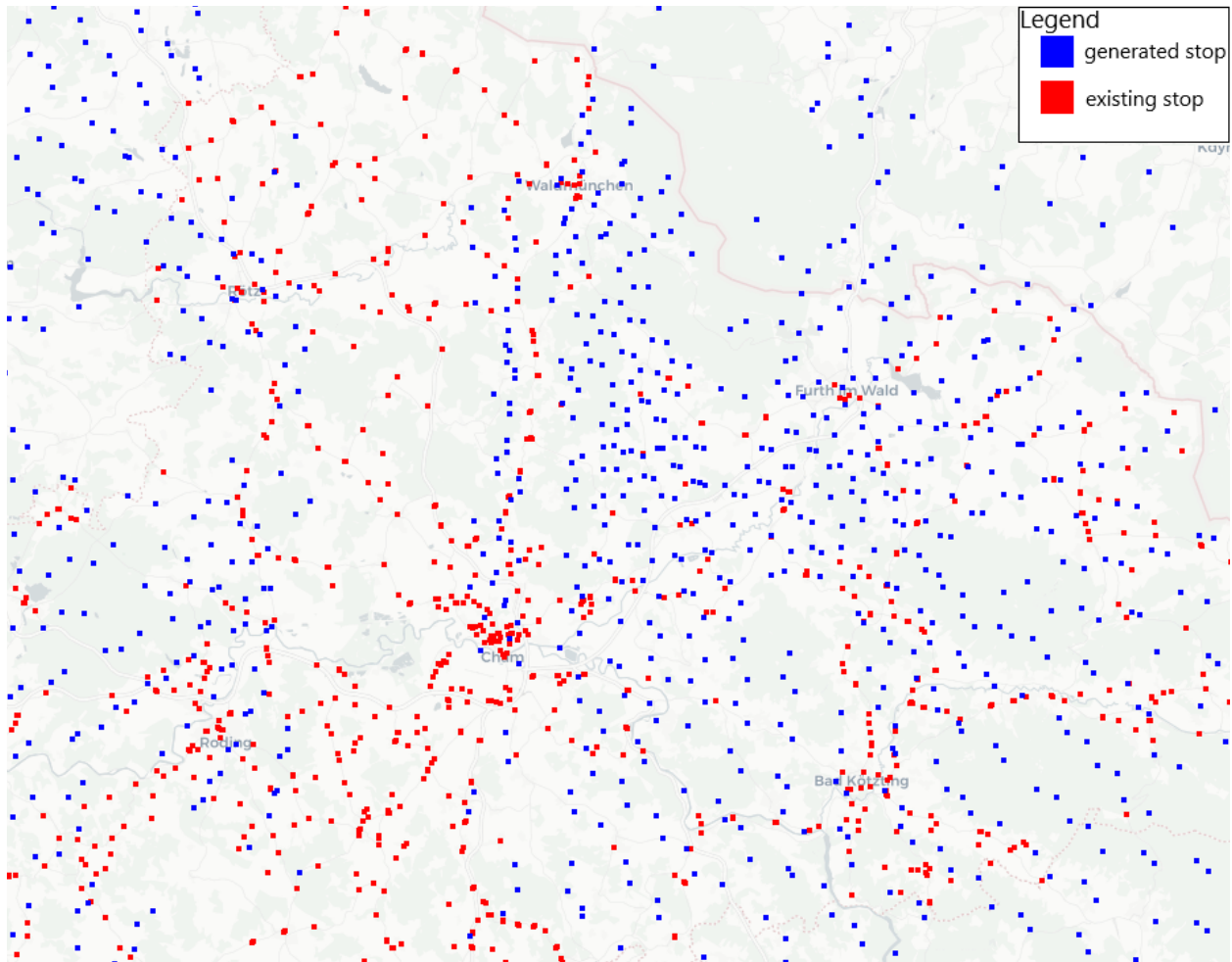


Figure 4. 2083 new and the already existing bus stops for the catchment area of Roding

An initial test in the form of integration of our results into the SUMO tool showed a shorter mean travel time for individual pedestrians [23], confirming the good positioning of new stops. For example, currently, the population uniform is estimated and does not fully correspond to the distribution in the real city of Roding. The system we developed can be extended by further attributes and parameters at any time due to its modular structure. However, city planners and bus operators in Roding are already experimenting with the tool to track where a new bus stop is needed, or which stops have long been unused and, therefore, unnecessary.

5.2. Comparison with other models

In this section, we compare our new model with existing other models. The p -median problem describes how to localize p facilities to reduce the demand weighted average distance between demand nodes and the closest facility. On general graphs, this is a

\mathcal{NP} hard problem that can be solved by a tree in polynomial time [24]. Unlike this problem, our method is not based on pre-existing points and finding the shortest average distance between them, but rather new facilities are dynamically added to an existing graph to achieve the largest possible spatial coverage. However, the p -median problem also occurs in our use case. E.g. when a passenger searches for a bus stop in the generated model. This can be solved for example by a Neighborhood Search Algorithm. The p -center problem is a subproblem of the p -median problem. It is a min-max solution that reduces the maximum distance between a demand point and a closest point in a set of p points [24]. The p -center problem has more similarities than the p -center with our method, since no discrimination by clustering occurs here. Thus, with the p -center problem, special attention is paid to individuals who have very poor transportation access. We created a middle ground by weighting

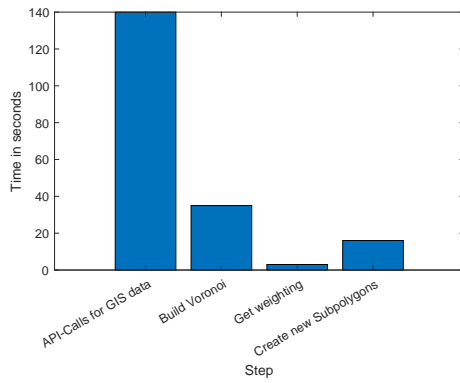


Figure 5. Runtime for the individual steps of the method for 2200 new bus stops

in our approach between minimizing total or average distance and minimizing the maximum distance of a single road user. By shifting the parameters in our weighting, the focus between these two extremes can also be changed dynamically. Two other methods, namely MLCP and the location-allocation method describe both possibilities, to find the optimal location for a company. A wide variety of factors can be considered in the analysis. Some of these include maximizing accessibility, minimizing the number of facilities, or maximizing site coverage. The facilities of competitors are also recorded and the market share can be determined. This determination of the market share based on the competition is counterproductive in our application example. Bus stops are usually served by several companies at the same time, and there is a rather and there is a rather cooperative behavior in local public transport. Therefore, no detailed analysis as in location-allocation modeling is necessary and our model with fewer parameters that benefit all bus companies in the area can be used. The maintenance location choice problem exhibits an interesting approach

that has not been considered in our approach so far. Some solutions to this problem deal not only with the current best location of a facility but also the future best [25]. Due to increasing urbanization and an increase in population, housing can be spread out very quickly and the weighting of our model has been shifted so that the positions of stops in the future are no longer optimal.

6. Conclusion

In this paper, a new method was presented that we developed to position bus stops based on the subdivision of Voronoi diagrams. This method works with a weighting based on GIS data. Especially, the surrounding points of interest and houses are used as attributes.

We have shown how to use the method in various ways in a case study. In an area with already existing bus stops the method can e.g. uncover places where a new bus stop would make sense. The position of already existing bus stops can be changed for an optimization of the distances between bus stops and better accessibility. In addition, it is possible for an area in which there are no bus stops to set as many stops as one wants. Creating a large number of bus stops makes sense especially for a demand bus solution where stops are not "buildings" but temporally used collection points with changing locations that are broadcasted via a smartphone application. As soon as enough people in a rural area request a bus, it could be called to one of the numerous new bus stops. Furthermore, the goal of setting new bus stops achieved in this paper can be combined with our previous work. By being able to predict passenger demand for existing ones and knowing where a new stop would be useful, a better-utilized system of bus routes can be generated [5]. The method has also no problems to calculate stops in the four-digit range. These many bus stops are not necessary for the area of the small town Roding. Even in megacities, such a large network of stops is unnecessary. However, if call buses are to be supported in the future and requests come from many people at the same time, it will make sense to make these calculations often. That many stops are necessary when road regulations do not allow a passenger to get on or off the bus if no bus stop is declared at this location, as it is the case in European countries and many others. To circumvent these regulations, it may make sense to declare the numerous stops generated by our method as real bus stops every time it passes by. This does not mean that a bus has to stop there.

In the next step, the newly determined stops can be combined with other tools, to create optimized bus routes. Integration into traffic simulation environments

such as Simulation of Urban MObility (SUMO) [23] to test the new stops directly is also possible immediately. This gives city planners and officials a very quick insight on which to base decisions about how to expand public transportation.

References

- [1] B. She, X. Zhu, X. Ye, W. Guo, K. Su, and J. Lee, "Weighted network voronoi diagrams for local spatial analysis," *Computers, Environment and Urban Systems*, vol. 52, pp. 70–80, jul 2015.
- [2] P. J. Green and R. Sibson, "Computing dirichlet tessellations in the plane," *The Computer Journal*, vol. 21, pp. 168–173, may 1978.
- [3] I. Heywood, S. Cornelius, and S. Carver, *An Introduction to Geographical Information Systems*. Prentice Hall, June 2011.
- [4] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, Oct. 1998.
- [5] T. Stadler, A. Sarkar, and J. Dünneweber, "Bus demand forecasting for rural areas using xgboost and random forest algorithm," *20th International Conference on Computer Information Systems and Industrial Management Applications*, p. to appear, 2021.
- [6] C. Gold, "Crust and anti-crust," in *Proceedings of the fifteenth annual symposium on Computational geometry - SCG '99*, ACM Press, 1999.
- [7] O. Wiki, "Overpass api," *URL https://wiki.openstreetmap.org/wiki/Overpass_API*, 2019.
- [8] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning," in *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, IEEE, jul 2007.
- [9] O. Lebedeva, M. Kripak, and V. Gozbenko, "Increasing effectiveness of the transportation network by using the automation of a voronoi diagram," *Transportation Research Procedia*, vol. 36, pp. 427–433, 2018.
- [10] M. Ghasedi, M. Ghorbanzadeh, and I. Bargegol, "Robust optimization of bus stop placement based on dynamic demand using meta heuristic approaches: A case study in a developing country," *Transport and Telecommunication Journal*, vol. 22, pp. 39–52, jan 2021.
- [11] I. Bargegol, M. Ghorbanzadeh, M. Ghasedi, and M. Rastbod, "Evaluation of effective factors on travel time in optimization of bus stops placement using genetic algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 245, p. 042002, oct 2017.
- [12] L. Mu, "Polygon characterization with the multiplicatively weighted voronoi diagram," *The Professional Geographer*, vol. 56, no. 2, pp. 223–239, 2004.
- [13] E. V. Brovarone and G. Cotella, "Improving rural accessibility: A multilayer approach," *Sustainability*, vol. 12, p. 2876, apr 2020.
- [14] H. Zhang, S. Zhao, H. Liu, and S. Liang, "Design of limited-stop service based on the degree of unbalance of passenger demand," *PLOS ONE*, vol. 13, p. e0193855, mar 2018.
- [15] J. Gallagher, "The last mile problem: Activists, advocates, and the struggle for justice in domestic courts," *Comparative Political Studies*, vol. 50, pp. 1666–1698, mar 2017.
- [16] M. Schuster, "The largest empty circle problem," in *Proceedings of the Class of 2008 Senior Conference, Computer Science Department, Swarthmore College*, pp. 28–37, 2008.
- [17] G. Gutiérrez, J. R. López, J. R. Paramá, and M. R. Penabad, "The largest empty circle with location constraints in spatial databases," *Knowledge and Information Systems*, vol. 55, pp. 141–169, jun 2017.
- [18] E. M. Delmelle, S. Li, and A. T. Murray, "Identifying bus stop redundancy: A gis-based spatial optimization approach," *Computers, Environment and Urban Systems*, vol. 36, pp. 445–455, sep 2012.
- [19] W. McKinney *et al.*, "pandas: a foundational python library for data analysis and statistics," *Python for High Performance and Scientific Computing*, vol. 14, no. 9, pp. 1–9, 2011.
- [20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt, "SciPy 1.0: fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, pp. 261–272, feb 2020.
- [21] A. Cuttone, S. Lehmann, and J. E. Larsen, "Geoplotlib: a python toolbox for visualizing geographical data," Aug. 2016.
- [22] P. Neis and A. Zipf, "Openrouteservice. org is three times "open": Combining opensource, opens and openstreetmaps," *GIS Research UK (GISRUK 08). Manchester*, 2008.
- [23] M. A. D. Khumara, L. Fauziyyah, and P. Kristalina, "Estimation of urban traffic state using simulation of urban mobility(SUMO) to optimize intelligent transport system in smart city," in *2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, IEEE, oct 2018.
- [24] M. S. Daskin and K. L. Maass, "The p-median problem," in *Location Science*, pp. 21–45, Springer International Publishing, 2015.
- [25] D. Kik, M. G. Wichmann, and T. S. Spengler, "Decision support framework for the regional facility location and development planning problem," *Journal of Business Economics*, jun 2021.