Partitioning of urban networks to optimize postal delivery routes

Mark Bogataj[a], **Jakob Maležič**[a], **and Luka Žontar**[a,1]

[a]University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, SI-1000 Ljubljana, Slovenia

The manuscript was compiled on June 21, 2021

In this article, we study the optimization of postal delivery routes in an urban network of a smaller city. Since e-shopping is becoming a trend and is changing the way customers and providers interact, postal services are facing piles of packages to deliver, with limited human resources and available work hours. However, postal delivery routes can still be improved in many ways and in this article we try to develop a novel approach to the aforementioned problem. While lots of work has been done in the area of optimizing the path from point A to point B, we will focus more on finding the optimal partitioning of the city to minimize the time spent and the distance made by a postman, while also equalizing the amount of labor over partitions. We present automatization of city infrastructure graph construction. We evaluate several graph partitioning methods based on average travel time and its standard deviation, which represents the differences in the amount of labor required to deliver mail in particular district. And lastly, we show the interactive dashboard with which users can more easily perform network analysis.

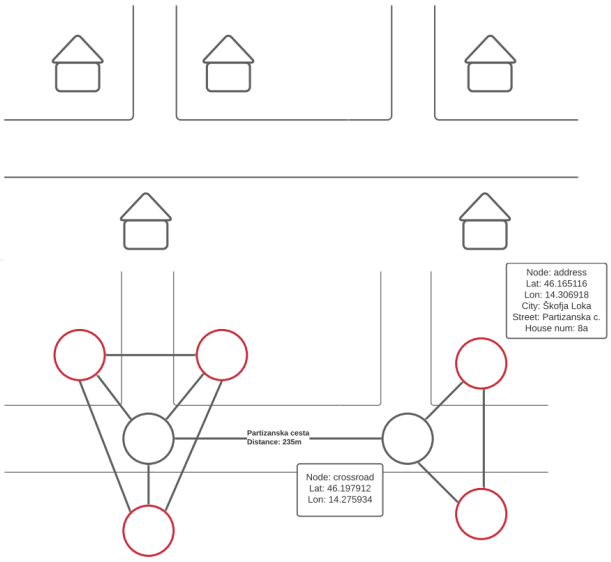

Fig. 1. Abstract illustration of graph structure

Problem definition, motivation, background and contributions. While e-shopping is becoming a trend, hand-written mails and printed invoices are rapidly getting replaced with electronic alternatives and modern society is rapidly changing the way customers and providers interact. Nevertheless, postal services are facing piles of packages to deliver with limited human resources and time. On top of that, customers are growing restless and want to receive mail as soon as possible due to technological developments and swiftly growing pace of contemporary life. This is why optimizing the delivery system of postal services is a task that was overlooked for too long.

There are several potential improvements in existing postal delivery systems. As an example, each postman can have a different perspective on to where the optimal start of delivery is. While it might be a good idea to start with the nearest possible point, we should also consider first visiting addresses in the centre of the city due to traffic jams that could occur at later hours. It is also important for a postman to find the optimal path through a given district. And finally, we should try finding the optimal partitioning of the mail delivery network, such that all the workers will be equally occupied and will finish as early as possible.

While we can contribute to improving postal delivery systems in many ways, we will focus on analyzing optimal network partitioning. The focal point of our research will be to find which network properties should be used to get the optimal network partitioning. Even though city partitioning in postal districts are already used in practice, we believe they can be further improved by exploiting various network analysis techniques. It is quite safe to assume that houses that are close to each other will most likely be in the same partition, however, we will try to optimize the position of borders between partitions.

Even minor improvements in the aforementioned processes, could vastly improve the overall postal distribution over a year and consequently decrease the waiting time of customers to get their mail. Furthermore, it would also decrease expenses of postal companies, which is why, we assume they would value greatly any contributions made in this area.

Our main contributions are the analysis of how different partitioning methods and network properties can be used to obtain the optimal postal delivery system. In this article, we work on a real-life example by analyzing the partitioning of a small Slovenian city Tolmin. Additionally, we provide scripts and instructions to extract similar networks on other cities [*] and an interactive dashboard that helps with a more thorough network examination [†].

Related work

In this section, we will present relevant literature that is related to the content of our project. Similar to what we are trying to

[*]Instructions and scripts to extract infrastructures of other Slovenian cities can be found on Git: https://github.com/lzontar/Partitioning-of-urban-networks-to-optimize-postal-delivery-routes

[†]Interactive dashboard is deployed via Heroku and can be accessed on: https://delivery-sys-opt.herokuapp.com/

All authors contributed equally to this work.

[1]To whom correspondence should be addressed. E-mail: lz3057@student.uni-lj.si.

achieve with postal delivery systems, Liu et al. (1) focused on clustering weighted graphs for community detection of large social networks. Their contribution was an algorithm that takes into account users' attractiveness and core degree to detect communities.

Network analysis can be applied to a vast area of problems. Holmberg proved that by introducing heuristics to optimize urban snow removal (2). With his colleagues, he redefined the problem to solve it as a k-Chinese Postman Problem. The latter algorithm was thoroughly explained in Osterhues's and Mariak's work (3).

However, it is not necessary that our community detection algorithm only recognizes convex communities. Gong et al. (4) proposed a non-convex optimization approach for finding appropriate communities. Another area of research that is crucial in our project is the representation of the city infrastructure. Marshall with his colleagues (5) analyzed different ways to build a street network. In our work, we somewhat replicate this by using junctions (i.e. crossroads) in our graph construction.

Similary as we will in our project, Anwar et al. (6) tried to partition a graph that represented urban road infrastructure, so they could detect traffic congestion patterns. On Amsterdam city network they compared three techniques to partition a graph, where they tried to maximize the speed of the average vehicle. They concluded that no technique stands out, with respect to metrics TVn and CCD. Finally, Yi and Geroliminis (7) used NCut and k-means algorithms on a real urban transportation network to partition a graph and detect congested areas, where mobility should be improved.

Wang and other authors (8) proposed the use of a clustering algorithm named DBSCAN, which is a density-based spatial clustering method. One of its qualities is, that it does not require the number of clusters as an input, but rather considers the minimum number of data points in each cluster and the minimum distance between clusters to find the optimal partitioning. One of the vastly used method for graph partitioning is also bisection (9), where the algorithm divides the graph into two equally sized subgraphs. This characteristic suits well to our problem, since our goal is to equally divide work between postmen.

To evaluate the partitioning, we will need to find the optimal partition traversal and compute the travel duration of such traversal. Such problem is well known and defined as traveling salesman problem. It tries to find the optimal path through the graph, while visiting all the nodes in the graph and returning to the original node. Traveling salesman problem can be solved using genetic algorithm as proposed by Mohan et al. (10). Genetic algorithms are designed to replicate the natural selection process to carry generation. In other words, they implement the concept of: survival of the fittest.

Results

As mentioned above, we decided to try and solve a more practical problem and have constructed a network of a smaller Slovenian city called Tolmin. Our network consists of two types of nodes: crossroads and housing complexes. Nodes are connected with roads, where each connection holds the information about the travel duration between the housing complexes or crossroads. We assume postmen are using scooters to deliver mail. Each crossroad is connected to the next and the previous crossroad on the corresponding roads. Each housing complex is connected to the closest crossroad and to all the housing complexes that are connected to this crossroad. Such network structure was defined to reduce time complexity and still preserve the practicality of the problem. City infrastructure network of Tolmin consists of 324 housing complexes and 79 crossroads.

Algorithms\Clusters	C1	C2	C3	C4	Standard deviation
K-means	128	27	105	64	38.7
K-means (clusters of equal sizes)	81	78	86	79	3.08
Fluid communities	139	136	33	16	56.83
AGDL	115	81	76	52	22.48
Girvan-Newman	114	101	64	45	27.72
K-cut (number of neighboring households)	81	80	81	82	0.71
K-cut (travel duration)	82	83	81	78	1.87

Table 1. Results: Number of households per cluster distribution.

Clusters \ Algorithms	C1	C2	C3	C4	Average travel duration	Standard deviation
K-means	2:04:42	0:42:20	2:01:11	1:33:12	1:35:21	0:32:57
K-means (clusters of equal sizes)	2:03:57	1:59:33	1:35:32	1:45:26	1:51:07	0:11:18
Fluid communities	3:29:52	2:56:37	0:46:52	0:34:28	1:56:57	1:17:18
AGDL	2:42:33	1:07:08	1:36:06	0:29:15	1:28:45	0:48:45
Girvan-Newman	2:38:57	2:27:07	0:37:27	0:39:12	1:35:40	0:57:30
K-cut (number of neighboring households)	2:11:03	1:29:17	1:44:59	2:08:39	1:53:29	0:17:17
K-cut (travel duration)	2:36:35	1:51:08	1:48:33	1:46:25	2:04:40	0:20:48

Table 2. Results: Travel duration per cluster distribution (each time entry is given in the hh:mm:ss format).

For simplicity, we assume that postmen need to visit all the houses in Tolmin and that crossroads are uniformly distributed over clusters and also have to be visited. These assumptions can be easily translated to our problem as postmen often visit almost all households to distribute advertisements flyers. Obviously, to deliver mail to a household they often pass a crossroad, which is why our assumptions are justified.

In Table 1, we see results of partitioning algorithms, where the distribution of the number of households in each partition is shown. This metric is crucial to ensure equality of labor in each partition. Quality of partitioning can be evaluated using the standard deviation of the number of housing complexed over partitions. Our goal was to minimize the aforementioned metric. We see that three methods stand out as the best in doing so. Both k-cut methods and k-means have very small standard deviation, while the rest do not perform so well in optimizing this metric.

The travel duration of a graph traversal is evaluated with genetic algorithm solution of the travelling salesman problem, which was implemented without the use of any third-party libraries. In a subgraph, we first find the optimal traversal over all the crossroads and then iterate over crossroads and add the travelling time of the traversal over the neighboring

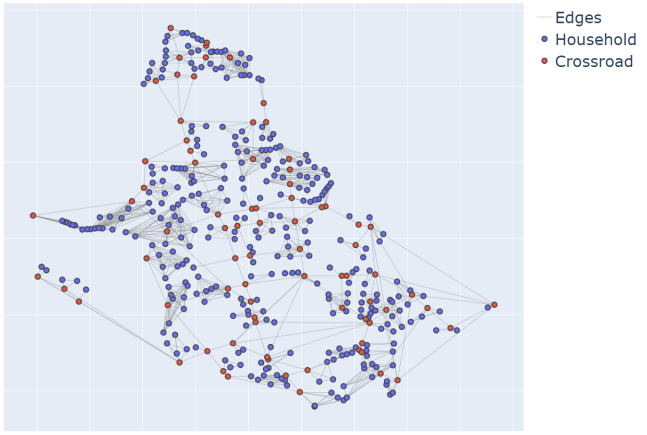

Fig. 2. Tolmin: Graph structure. In figure we see a visualization of previously presented graph structure.

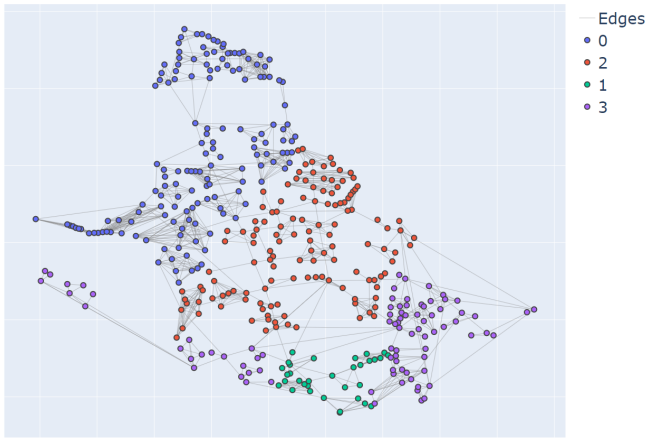

Fig. 3. Tolmin: Partitioning with AGDL. In figure we see a visualization graph partitioning with AGDL algorithm. As described in section **Results**, this is the method that finds clusters with minimum average travel duration.

housing complexes.

We show results of traveling durations of different graph partitionings in Table 2. Besides the distribution of travel durations over graph partitions, we also show the average travel durations and its standard deviation. The latter metrics are most important and will be used to determine the optimal graph partitioning. Our goal is to minimize the average travel duration over all the partitions. This way, we will decrease the time that postmen spend to deliver mail. Furthermore, we try to minimize the standard deviation to enforce the equality of labor over all partitions and consequently postmen. Note that, while k-cut algorithms are very efficient in minimizing standard deviation, they perform the worst if ranked by the average travel duration. On the other hand, we see that AGDL has much lower average travel duration, but much bigger standard deviation. We notice that fluid communities algorithm probably performs the worst. Its average travel duration is quite high, while it also fails to reduce standard deviation.

We can see, that our methods can be easily divided into two conceptually distinct groups. First, we have methods such as k-cut algorithm using either node weights, where weights are calculated using the number of neighboring housing complexes, or edge travelling duration weights. K-means that tries to produce clusters of equal sizes also belongs to this group, where the main goal is to equalize the labor over all the clusters. Either by equalizing the number of nodes in each cluster or travelling duration. On the other hand, we have methods that try to minimize the average travelling duration over clusters or fail to do both of our tasks.

Discussion

In this section, we present our main contributions and discuss relevant results. As already announced, our contributions consist of three parts. Firstly, we defined instructions on how to build the city infrastructure graph and automatized the process to an acceptable extent.

Secondly, we developed an interactive dashboard to simplify network analysis. There, you are able to choose the city and the algorithm by which you would like to partition the network. The dashboard shows graph structure and partitioning as can be seen in figures 2 and 3. At the moment, graphs need to be premade to use the tool and we only included Tolmin city infrastructure graph in the dashboard. However, in future work, we would like to both further automatize the graph creation of an arbitrary city and allow user to change more parameters.

To conclude with the most important contribution, we evaluated different graph partitioning methods that are further described in section **Methods**. As already mentioned, AGDL is the algorithm that provides partitioning with lowest mean travelling time. It exploits network analysis properties such as the number of indegrees and outdegrees. While we do not have a directed graph, it might be that the algorithm notices our graph structure, i.e. connected conjunctions and connecting housing complexes to nearest crossroad. This might lead us to conclusion that to get partitioning with optimal travelling duration, we should simply assign postmen conjunctions and each postman should visit all the housing complexes that are closest to their assigned crossroads.

On the other hand, if we were to minimize the differences in travelling duration in partitions, we might prefer to use k-cut algorithm or k-means that produces clusters of equal sizes. Both k-means and one of the two k-cut algorithms perform same task in a different manner and the task they perform is actually maximizing the equality of labor. Interestingly, k-cut algorithm that uses travel duration on edges performs almost just as good. Due to our graph structure we will be removing edges with maximal travel duration and thus cutting the graph in partitions, where we will cut connections between distant crossroads, since travel duration there is highest. This makes sense, if we were to assume that the number of housing complexes that belong to a crossroad is uniformly distributed over all the crossroads. Similar results can also be seen in distributions of the number of houses over partitions.

In contrast to the equality of labor metrics, k-cut algorithms and k-means that produces clusters of equal sizes perform quite poorly in average travelling duration. From this, we might conclude that providing all the postmen the same amount of work might not be the best possibility. Postal service providers might benefit greatly, if their postmen would not need to visit the same amount of housing complexes. A way of introducing such change would be to suggest that postmen would switch

districts on the weekly basis.

The regular k-means seems as the optimal choice by providing the equilibrium of both equality of labor and minimal travelling time. We see that clustering using only travelling durations works well. Such implementation might be viewed as partitioning of a fully connected graph, where non-existing edge costs are replaced with costs of paths in graph found by Dijkstra's algorithm.

Note that although choosing four partitions might seem irreasonable since the overall travelling time of a single postman rarely exceeds two hours, you should be aware that the majority of time that a postman spends on his job is not travelling, but rather sorting mail, waiting for recipients (if signature or payment is necessary) or similar.

In addition to already described features that will be implemented in our dashboard, in future work, we would like to develop a method that optimizes for travel duration while including equality of labor constraint, where both the number of houses and traveling time should be considered at the same time. Additionally, we might want to further automatize the retrieval of city infrastructure graphs. And lastly, we would like to derive a practical solution to our problem by developing a software or add-on that postal service providers would be able to simply integrate in their existing system to provide better and more efficient services.

Methods

The following section will focus on different methods that were used to partition our network. For each of the methods, we will explain how it works and which network characteristics it uses to split the graph.

Girvan-Newman algorithm (11). The Girvan-Newman algorithm focuses on removing the most valuable edge in each iteration, where such edge has the highest betweenness centrality. Since edges that supposedly connect clusters are removed, we eventually generate multiple connected components. The algorithm returns the desired number of connected components, which in our case, are used as partitions.

AGDL (12). AGDL is an agglomerative algorithm used for clustering graphs. Using the product of average indegrees and outdegrees, the authors of the algorithm defined the affinity measure of clusters. Essentially, it exploits indegrees and outdegrees to detect partitions in a graph.

K-Cuts (13). An alternative view on graph partitioning is iteratively cutting the graph into two disjoint sets until the desired number of partitions is achieved. METIS (14) is a set of programs to partition graphs that we used in proposed methodology. The aforementioned algorithm aims to find the maximum cut, where the aggregate weight is not smaller than the weight of any other cut. This is one of the few methods that tries to comprehend the importance of weights in our graph. We perform k cuts to get the optimal partitioning of the city infrastructure.

We evaluate two variations of k-cuts algorithm. The first uses the duration of travel between a pair of nodes in our graph. And the second tries to find partitions with equal sum of node weights. We define weights to be the number of neighboring households. Therefore, such algorithm tries to find partitioning with equal number of households in each partition.

K-Means (15). A well known clustering method that aims to partition observations, which are nodes in our case, using prototype observations. The goal of the algorithm is to calculate the position of these prototypes. All observations are then associated with the closest prototype and each prototype forms a partition.

K-means usually uses Euclidean distance or similar to calculate distances between nodes. In our case, we use the weights of edges, which can either be the duration of travel between nodes or the distance between them.

We implemented two variations of k-means. One is regular, while the other tries to find clusters of equal sizes. This is crucial since we do not want to overoccupy any of the postmen.

Fluid communities (16). This algorithm is based on the simple idea of fluids (i.e. communities) interacting in an environment (i.e. a non-complete graph), expanding and contracting. It is propagation-based algorithm and it allows to specify the number of desired communities. Furthermore, the implementation we use is asynchronous, where each vertex update is computed using the latest partial state of the graph.

1. Ruifang Liu, Shan Feng, Ruisheng Shi, and Wenbin Guo. Weighted graph clustering for community detection of large social networks. *Procedia Computer Science*, 31:85–94, 2014. ISSN 1877-0509. . 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014.
2. Kaj Holmberg. Heuristics for the weighted k-rural postman problem with applications to urban snow removal. *Journal on Vehicle Routing Algorithms*, 1, 03 2018. .
3. A. Osterhues and Frank Mariak. On variants of the k-chinese postman problem. 2005.
4. Maoguo Gong, Qing Cai, Ma Lijia, and Licheng Jiao. *Big Network Analytics Based on Non-convex Optimization*, pages 345–373. 05 2016. ISBN 978-3-319-30263-8. .
5. Stephen Marshall, Jorge Gil, Karl Kropf, Martin Tomko, and Lucas Figueiredo. Street network studies: from networks to models and their representations. *Networks and Spatial Economics*, 18:1–15, 09 2018. .
6. Tarique Anwar, Chengfei Liu, Hai Vu, and Christopher Leckie. Spatial partitioning of large urban road networks. 05 2014. .
7. Yuxuan Ji and Nikolas Geroliminis. On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, 46(10):1639–1656, 2012. ISSN 0191-2615. .
8. Tianfu Wang, Chang Ren, Yun Luo, and Jing Tian. Ns-dbscan: A density-based clustering algorithm in network space. *ISPRS International Journal of Geo-Information*, 8:218, 05 2019. .
9. Ravi B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 280–285, 1987. .
10. Senthilkumar Mohan, Nallakaruppan Kailasanathan, Chandra Segar Thirumalai, and Prasanna Santhanam. A modified and efficient genetic algorithm to address a travelling salesman problem. *International Journal of Applied Engineering Research*, 9:1279–1288, 01 2014.
11. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. ISSN 0027-8424. .
12. Wei Zhang, Xiaogang Wang, Deli Zhao, and Xiaoou Tang. Graph degree linkage: Agglomerative clustering on a directed graph. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 428–441, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33718-5.
13. M. E. J. Newman. Community detection and graph partitioning. *EPL (Europhysics Letters)*, 103(2):28003, jul 2013. . URL https://doi.org/10.1209/0295-5075/103/28003.
14. George Karypis and Vipin Kumar. Metis – unstructured graph partitioning and sparse matrix ordering system, version 2.0. 01 1995.
15. Xin Jin and Jiawei Han. *K-Means Clustering*, pages 563–564. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. . URL https://doi.org/10.1007/978-0-387-30164-8_425.
16. Ferran Parés, Dario Garcia-Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. Fluid communities: A competitive, scalable and diverse community detection algorithm. pages 229–240, 11 2018. ISBN 978-3-319-72149-1. .