

A modified multilevel k-way partitioning algorithm for trip-based road networks

C Withanage*, D Lakmal, M Hansini, K Kankanamge, Y Witharanage, and U Thayasivam

Department of Computer Science and Engineering, University of Moratuwa,
Sri Lanka

Abstract. In today's world, the traffic volume on urban road networks is multiplying rapidly due to the heavy usage of vehicles and mobility on demand services. Migration of people towards urban areas result in increasing size and complexity of urban road networks. When handling such complex traffic systems, partitioning the road network into multiple sub-regions and managing the identified sub regions is a popular approach. In this paper, we propose an algorithm to identify sub-regions of a road network that exhibit homogeneous traffic flow patterns. In a stage wise manner, we model the road network graph by using taxi-trip data obtained on the selected region. Then, we apply the proposed modified multilevel k-way partitioning algorithm to obtain optimal number of partitions from the developed road graph. An interesting feature of this algorithm is, resulting partitions are geographically connected and consists minimal inter-partition trip flow. Our results show that the proposed algorithm outperforms state-of-the-art multilevel partitioning algorithms for trip-based road networks. By this research, we demonstrate the ability of road network partitioning using trip data while preserving the partition homogeneity and connectivity.

1 Introduction

Road traffic networks are rapidly growing in size with increasing population and urbanization. The increased usage of motor vehicles worsens the problem of traffic congestion and creates a more challenging environment for the traffic management. In managing, these road traffic networks can be recognized as distinct sub-networks or partitions which exhibit homogeneous traffic flow patterns within the partitions. By identifying such traffic flow patterns, partition specific traffic flow management strategies can be implemented. To manage traffic in an area, it is important to identify the patterns of traffic. Road network partitioning helps in identifying the traffic flow patterns and partition accordingly.

With urbanization and mobility on demand systems getting popular among the community, people tend to use taxi services instead of their own vehicles. Reference [1] states that the total number of licensed taxi and private hire vehicles in England increased by 16% at the end of 2017 proving higher community involvement in using taxi services.

* Corresponding author: chanaka.14@cse.mrt.ac.lk

Many of the taxi services provide facilities like ridesharing which is cheap comparative to using a privately-owned vehicle. A taxi trip trajectory data contains information about pick up geo location, drop off geolocation, pickup time, travel time, distance travelled and many other taxi service related attributes. Therefore, data obtained from taxi trip trajectories is a key component in identifying community behaviour and traffic condition variations in an area.

Traffic flow in an urban area is governed by multiple factors. For example, traffic conditions near shopping complexes, hospitals are heavier due to dense population while traffic conditions are minimum in less populated areas. Traffic conditions also demonstrate time varying characteristics. That is, in rush hours higher congestion can be seen, same as for weekends and public holidays. Weather conditions in an area is an important factor to determine traffic conditions. Heavy rains and thick fog increases traffic due to unclear roads. Due to this dependency on multiple factors, identification of partitions which show traffic homogeneity is extremely complex.

In summary, the contributions of this paper are as follows,

- We propose an algorithm to obtain optimal number of partitions for a large urban road network where partitions are geographically connected and preserve the traffic homogeneity.
- We demonstrate that large scale taxi trip trajectory data can be used to partition road networks.

Rest of this paper is organized as follows: Section 2 contains, a literature review of the related work on graph partitioning and road network partitioning. Section 3 provides the methodology of constructing the road network graph. Section 4 describes our experiment, Section 5 shows our results and comparison with existing algorithms and finally we conclude the paper in Section 6.

2 Literature Review

2.1 Road Network Partitioning

A number of works have been done on road network partitioning. A recent work [1] of Locus D. and Paraskevas N. have introduced a graph partitioning technique namely gpMETIS which applies multilevel k-way partitioning on urban road networks to identify different heterogeneous regions distinguished based on their geospatial and traffic patterns. The paper [1] partitions the road network in two different ways. First, the network has partitioned only considering the geospatial characteristics. Then, the network has partitioned based on both geospatial and operational characteristics., the results of these two different procedures have compared to conclude that considering both geospatial and operational characteristics results in more accurate partitioning. So, in this paper, we would follow a similar approach as in [1]. Splitting up large urban road networks based on their traffic congestion patterns using k-means clustering and a k-way α -cut partitioning has been introduced in [2]. In the paper [2], both spatial importance and temporal importance have considered in identifying traffic flow patterns. Similar to them we have also considered how the temporal importance affects to the partition quality.

Most of the road network partitioning publications based on the spatial characteristics of the network. Y. Ji and N. Geroliminis [3] presents a clustering procedure of transportation networks based on the spatial features of congestion during a specific time period. Furthermore, Insights are provided on how to extend this framework in the dynamic case. Spectral theory-based graph cut (referred as α -Cut) has used in the paper "Partitioning road networks using density peak graphs: Efficiency vs. accuracy" [4] to obtain partitions of large urban road networks based on traffic measures.

2.2 Graph Partitioning Methods

Graph partitioning problem has been addressed in many times through different fields of studies. The algorithm we use here is mainly based on the multilevel k-way partitioning algorithm proposed by G. Karypis and V. Kumar [5]. Multilevel partitioning algorithm is a widely used methodology to obtain effective partitions while handling multiple constraints. Multilevel k-way partitioning follows a simple approach First, A graph is coarsened down to few hundreds of nodes. Then, using recursive bisection algorithm the graph is partitioned into different regions. Finally, the partitioned graph is projected back to the original graph. In the paper “A Fast and High-Quality Multilevel Scheme for Partitioning Irregular Graphs” [6], the authors have experimented with various parameters of multilevel algorithms and their effect on the quality of partition and ordering.

Spectral bisection [7] is the other widely used partition algorithm which is well known for producing good partitions. The spectral bisection algorithms use the eigenvector of the graph Laplacian matrix corresponding to the algebraic connectivity of the graph. S. T. Bernard and H. D. Simon [8] propose a Spectral bisection algorithm which is based on the concept of maximal independent set. A refinement of spectral bisection algorithm is presented in the paper “Spectral bisection of graphs and connectedness” [9] by J. C. Urschel and L. T. Zikatanov.

Geometric partitioning algorithms [10], [11] are another common partitioning method that uses the geometric information of the graph. The paper [12] discusses the combination of techniques that have used to improve the geometric partitioning algorithms. A slightly similar problem has been addressed by the paper [13] which is about dividing a geographic region into sub-regions. Geometric partitioning algorithms are claimed to be faster than other partitioning algorithms but in our case speed of the partitioning process is not a limitation.

According to Meyerhenke H. et. al. [14], the current state of the art graph partitioning methodology is Multilevel K-way partitioning algorithm. So, we use that to obtain partitions in our scenario. But the road network partitions have some specific requirements which are not much concerned in common graph partitioning. The road segments in each subregion of a road network should be geographically connected. Furthermore, multilevel partitioning algorithms generate partitions of a graph using a bisection approach. So, the generated number of partitions is always a power of 2 (ex: - 2, 4, 8, ...). We will be addressing these issues later in the methodology section.

3 Methodology

Our main objective is to partition a road network into homogeneous components based on the taxi trip data. The basic requirement of the partitioning process is to minimize the trip number happens between the partitions. First, we convert the road network into a road graph. Then, the constructed graph is partitioned into different sub-regions using a modified multilevel Recursive bisection algorithm.

3.1 Road Graph Formation

The first step of the partitioning process is to convert the road network of a particular area into a graph which can be denoted as vertices and edges. We have considered the linkages of the roads (junctions) as vertices and the roads as edges.

Then, we take the taxi trip data of the selected area to weight the edges. First, we get the geographical coordinates of the start and end location of a trip. Then, we calculate the nearest nodes of the graph corresponding to the start and end locations. Using the road

network, we obtain the shortest path between the start and end node. Here, we assume that a passenger always uses the shortest path between two locations. Then, the edges along the obtained path are weighted according to the number of taxi trips occurred on that road.

3.2 Multilevel Recursive Partitioning

An explanation about the Multilevel Recursive Bisection is in this section. Furthermore, the modification done to optimize the algorithm for our specific scenario are discussed.

The road graph can be denoted as $G = (V, E)$ where V denotes the set of vertices and E denotes the set of edges in G . Junctions in the road network have represented as vertices and the roads have represented as edges, $E = (u, v)$. Here, u denotes the starting vertex of the edge and v denotes the end vertex of the edge. The goal of the partitioning process is to find a set of partitions ($P = \{P_1, P_2, P_3, \dots, P_k\}$), such that edge cut of the graph is minimized. The edge cut is the sum of edges where the two vertices are in different partitions. $Edgecut = \sum_{i=0}^n (e_i)$, Where $P(e_u) \neq P(e_v)$

Multilevel recursive bisection partitioning consists of three main steps.

3.2.1 Coarsening phase

First step of multilevel partitioning is the coarsening phase. In this step, the larger graph is mapped into a graph with a few hundred (or thousand) nodes. The coarsening makes the partitioning process easier and less time-consuming. There are several different approaches [15] to perform the coarsening.

```

Until graph is small enough
    u = select random vertex(graph)
    v = select random neighbour(u)
    collapse edge from u to v
    w = form new vertex and connect neighbours of u and v
    if vertex k adjacent to both u and v:
        Edge weight(w,k)=edge weight(u,k) +edge weight(v,k)

```

Fig. 1. Random matching algorithm.

We have used a random matching algorithm (figure 1) to generate coarsen graphs. In this algorithm, vertices in the graph are visited in a randomized order. If any vertex u , has not matched with any other vertex then, we select an unmatched neighbour of u as v . then, we collapse the edge between u and v . The coarsening process is done while ensuring the edge cut of the partitioned coarser graph is equal to the edge cut of the partitioned original graph.

3.2.2 Initial Partitioning Phase

In the second phase, we partition the coarsened graph using recursive bisection algorithm [6]. The objective of road graph partitioning is to minimize the inter-partition trips which means the edge cut. Traditional bisection algorithm divides the graph into two roughly equal partitions in each level.

```

Bisect partitioning(graph)
    U = Select a random node(graph)
    Move u to partition_A
    Move all remaining nodes to partition_B
    Prepare a priority queue for neighbours of u (based on edge cut gain)
    Until partition_A.size < partition_B.size:
        Move the top priority node to partition_A
        Refresh the priority queue

```

Fig. 2. Bisection Algorithm.

The algorithm (figure 2) starts with selecting a random vertex from the graph. Then, the selected vertex is used to construct a graph using a breadth-first approach. The number of partitions the bisection algorithm produces is always a power of 2 (ex: - 2, 4, 8). We have customized the algorithm to generate a desired number of partitions. The traditional algorithm extracts a vertex from the original graph and grows a graph around that selected vertex until the initial graph is split roughly into two equal partitions. Instead of growing the graph until it becomes a half, we stop at a desired threshold as shown in figure 3.

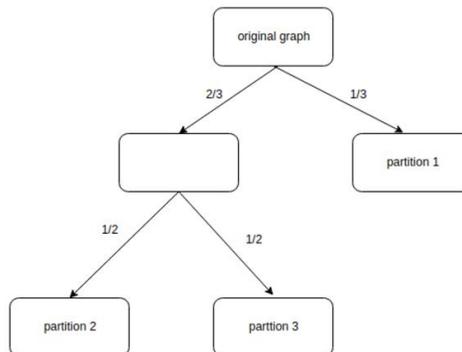


Fig. 3. Modified bisection partitioning.

Later in the result section, we can see that the modified bisection approach performs better than the traditional bisection in our case. The traditional recursive bisection starts with selecting a random vertex from the graph. The edge cut of the partitioning process depends on the starting vertex of the algorithm. Our experiments show that the different starting vertices can yield different edge cuts. Starting with a boundary vertex in the road network results in a smaller edge cut than a random start. Those boundary vertices have a degree of one. Furthermore, this produces a proper dividend of sub-regions in the road network.

3.2.3 Expanding Phase

In the final phase, we project the coarsen graph back to its original graph while preserving the partitions. Each vertex in the coarsen graph is a combination of two or more vertices in the original graph. We split these contracted vertices and connect them using the stored data. This process does not affect to the edge cut of the partitioning. Edge cut of the coarsen graph equals to the edge cut of the original graph.

The optimal partitioning of the coarsen graph may not be the optimal partitioning of the original graph because some new edges appear after the expansion of the contracted vertices. So, we refine the graph after the expanding phase to obtain an optimal partitioning. We have used the Kernighan and Lin (KL) [16] algorithm to refine the partitions.

4 Experiment

As described in the above methodology section after forming the road graph, we implemented the presented algorithm to partition the graph. The road graph has partitioned into different number of partitions (1, 2, 3, 4...) while focusing on minimizing the edge cut.

We selected the region of Colombo city to construct the road graph. The constructed graph has 33979 vertices and 100000 edges. The entire area covered by the road network is 2,654km². Taxi trips of a Local taxi service company from 2015/06/08 to 2015/10/08 have considered in the partitioning process. As described above the multilevel k-way partitioning does not guarantee a global minimum of edge cut in each partitioning. So, to get a fair result we have executed the algorithm 10 rounds for a particular partition count k and calculated the average edge cut. State of the art in graph partitioning is multilevel k-way partitioning and Metis is a software package which provides various multilevel algorithms for partitioning. In our case, we have used k way partitioning provided by Metis package to compare our k-way algorithm which has modified specially for the roadmap partitioning process

5 Results

Figure 4 shows the edge cut variation as the number of partitions k increases. Total weight of the graph is 54555355. The edge cut for each k has presented as a percentage of the total graph weight.

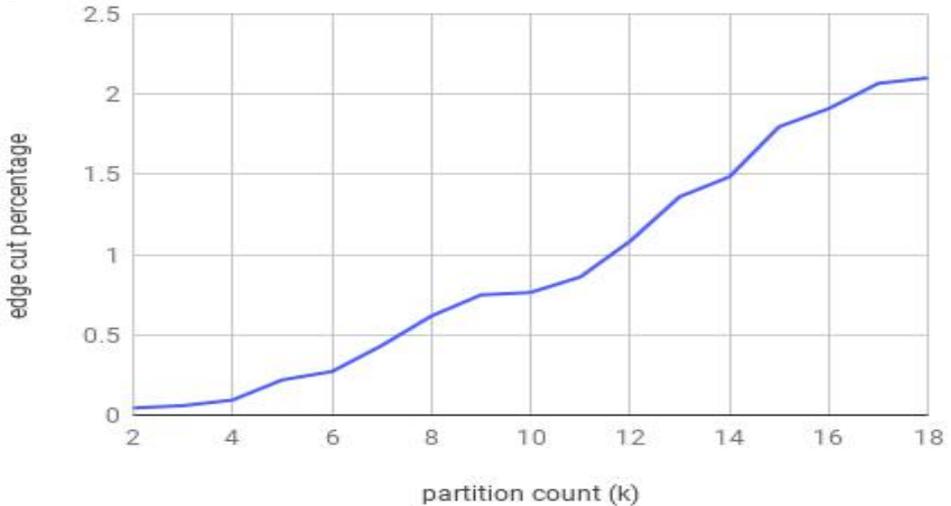


Fig. 4. Edge cut percentage vs Partition count.

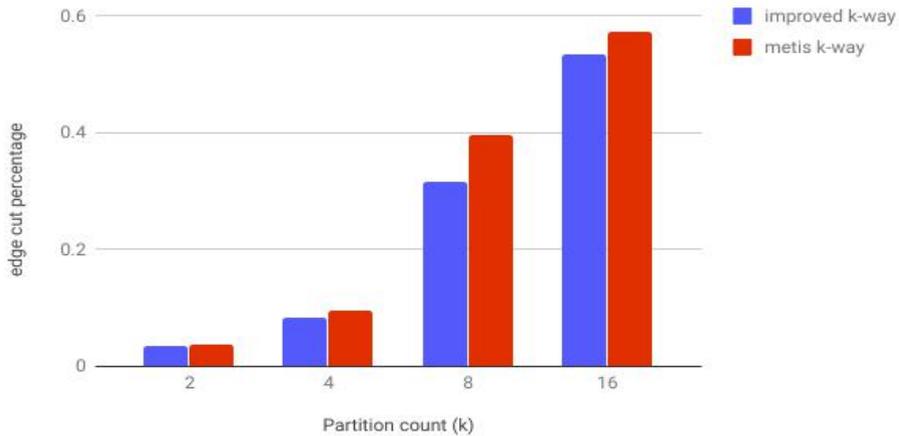


Fig. 5. Edge cut comparison between traditional and modified algorithms.

The performance of our modified multilevel k way partitioning algorithm has evaluated against the state-of-the-art metis k way partitioning algorithm (figure 5). The same road graph has partitioned using both algorithms and the results have compared using the edge cut of each partitioning process.

The resulting partitions of the selected road map have presented in the following figures (figure 6 – figure 9). The nodes belong to each partition has interpreted in different colours. Just for the ease of interpretation in the paper, we have presented the roadmap partitions from partition count $k = 2$ to 5.

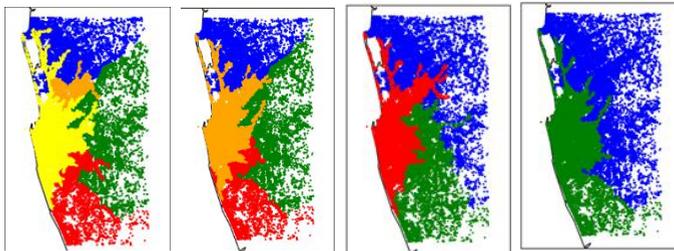


Fig. 6. $k = 2$ **Fig. 7.** $k = 3$ **Fig. 8.** $k = 4$ **Fig. 9.** $k = 5$

6 Conclusion

In this paper, we presented a modified multilevel k-way partitioning process for large urban road networks based on taxi trip data. Our goal was to find road partitions while focusing on minimizing the inter-partition trips. The road network converted into a graph with junctions as vertices and roads as edges. The road graph has weighted using the taxi trip data of the selected region.

State of the art for graph partitioning which is k-way partitioning algorithm was modified to generate partitions by minimizing the inter-partition trips flow. Despite the normal graph partitioning, road graph partitioning has some special requirements to fulfil. All the nodes of a particular partition should be geographically connected. We also customized the algorithms to give the ability to generate different number of partitions as we desired (2, 3, 4, 5...) rather than a $2n$ number of partitions.

Traditional k-way partitioning algorithm intends to produce balanced partitions. We modified the algorithm to produce partitions minimizing edge cut while allowing the partitions to be unbalanced in terms of size. The improved algorithm generated quality partition in our experiments for a wide range of k (number of partitions) values. For lower k

values like $k=2$, the edge cut stayed at 0.0243% from the total edge weight of the road graph while the traditional k -way partitioning generates a 0.038%. Even for higher k values like $k = 16$, the edge cut stayed at 1.910%.

The results suggest that our modified multilevel k -way partitioning algorithm can be used to partition road graphs effectively than traditional k -way partition algorithm. The inter partition trip flow is a very small number compared to the total trip count. So, the above proposed methodology can be used to effectively partition the road network. A partitioned road network will simplify the traffic management of the area and taxi trip operations.

Acknowledgement

This research work was partially supported by Digital Mobility Solutions PVT Ltd, Sri Lanka. We thank the company for fulfilling the necessary data and infrastructure requirements. The company does not hold any liability on whatsoever interpretations and conclusions made within the research.

References

1. <https://www.gov.uk/government/collections/taxi-statistics> [Online] Accessed On: 24-09-2018
2. L. Dimitriou and P. Nikolaou, "Dynamic partitioning of urban road networks based on their topological and operational characteristics," in 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017.
3. Tarique Anwar, Chengfei Liu, Hai L. Vu, Christopher Leckie, "Spatial Partitioning of Large Urban Road Networks," presented at the 17th International Conference on Extending Database Technology (EDBT 14), Athens, Greece, pp. 343–354.
4. Y. Ji and N. Geroliminis, "On the spatial partitioning of urban transportation networks," *Trans. Res. Part B: Methodol.*, vol. 46, no. 10, pp. 1639–1656, 2012.
5. T. Anwar, C. Liu, H. L. Vu, and C. Leckie, "Partitioning road networks using density peak graphs: Efficiency vs. accuracy," *Inf. Syst.*, vol. 64, pp. 22–40, 2017.
6. G. Karypis and V. Kumar. "Multilevel k -way partitioning scheme for irregular graphs" *Journal of Parallel and Distributed Computing*, 48(1):96–129, 1998.
7. G. Karypis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.
8. H. D. Simon, "Partitioning of unstructured problems for parallel processing," *Computing Systems in Engineering*, vol. 2, no. 2–3, pp. 135–148, 1991.
9. S. T. Barnard and H. D. Simon, "Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems," *Concurr. Comput.*, vol. 6, no. 2, pp. 101–117, 1994.
10. J. C. Urschel and L. T. Zikatanov, "Spectral bisection of graphs and connectedness," *Linear Algebra Appl.*, vol. 449, pp. 1–16, 2014.
11. H. D. Simon, "Partitioning of unstructured problems for parallel processing," *Computing Systems in Engineering*, vol. 2, no. 2–3, pp. 135–148, 1991.
12. M. T. Heath and P. Raghavan, "A Cartesian Parallel Nested Dissection Algorithm," *SIAM J. Matrix Anal. Appl.*, vol. 16, no. 1, pp. 235–253, 1995.

13. G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, “Automatic Mesh Partitioning,” in *The IMA Volumes in Mathematics and its Applications*, 1993, pp. 57–84.
14. S. Arora, S. Rao, and U. Vazirani, “Geometry, flows, and graph-partitioning algorithms,” *Commun. ACM*, vol. 51, no. 10, p. 96, 2008.
15. B. Hendrickson and R. Leland, “A multilevel algorithm for partitioning graphs,” in *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Supercomputing '95*, 1995.
16. B. W. Kernighan and S. Lin, “An Efficient Heuristic Procedure for Partitioning Graphs,” *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.