Optimization of administrative divisions as a computational graph theory problem

Peteris Daugulis

Daugavpils University, Latvia

September 7-14, 2025, BGU, Beer Sheva, Israel

Outline

Abstract

Main MSC 90C27: Combinatorial optimization.

Additional MCS 05C85: Graph algorithms.

This talk proposes a novel data-driven method for territorial division based on the Voronoi partition of edge-weighted road graphs and the vertex k-center problem. We show implementations of this approach in the context of Latvia.

Abstract

Main MSC 90C27: Combinatorial optimization.

Additional MCS 05C85: Graph algorithms.

This talk proposes a novel data-driven method for territorial division based on the Voronoi partition of edge-weighted road graphs and the vertex k-center problem. We show implementations of this approach in the context of Latvia.

Introduction

We assume that a country must be partitioned into a set of *territorial units (TU)* each containing a center.

For an administrative division to be well defined it must be based on a small set of quantitative parameters.

Time

We have chosen *time* as the single unifying human-centered metric.

We consider the time spent traveling to and from the TU center by motor vehicle to be the only quantitative parameter.

Introduction

We assume that a country must be partitioned into a set of *territorial units (TU)* each containing a center.

For an administrative division to be well defined it must be based on a small set of quantitative parameters.

Time

We have chosen *time* as the single unifying human-centered metric.

We consider the time spent traveling to and from the TU center by motor vehicle to be the only quantitative parameter.

Introduction

We assume that a country must be partitioned into a set of *territorial units (TU)* each containing a center.

For an administrative division to be well defined it must be based on a small set of quantitative parameters.

Time

We have chosen *time* as the single unifying human-centered metric.

We consider the time spent traveling to and from the TU center by motor vehicle to be the only quantitative parameter.

Introduction

We assume that a country must be partitioned into a set of *territorial units (TU)* each containing a center.

For an administrative division to be well defined it must be based on a small set of quantitative parameters.

Time

We have chosen *time* as the single unifying human-centered metric.

We consider the time spent traveling to and from the TU center by motor vehicle to be the only quantitative parameter.

Introduction

Time as tax

We view the time citizens spend to comply with their obligations or to receive crucial state or municipal services as a form of taxation.

Consequently, we seek to distribute this time burden as uniformly as possible among the population.

This approach intentionally departs from conventional methods encoded in the current legislature that typically prioritize certain regions, local interests, population size, security, defense, historical boundaries.

Introduction

Time as tax

We view the time citizens spend to comply with their obligations or to receive crucial state or municipal services as a form of taxation.

Consequently, we seek to distribute this time burden as uniformly as possible among the population.

This approach intentionally departs from conventional methods encoded in the current legislature that typically prioritize certain regions, local interests, population size, security, defense, historical boundaries.

Introduction

Time as tax

We view the time citizens spend to comply with their obligations or to receive crucial state or municipal services as a form of taxation.

Consequently, we seek to distribute this time burden as uniformly as possible among the population.

This approach intentionally departs from conventional methods encoded in the current legislature that typically prioritize certain regions, local interests, population size, security, defense, historical boundaries.

The main mathematical model — the road graph

We model the country as an undirected edge-weighted *road graph*, where vertices (nodes) represent towns/settlements/intersections and edges represent roads connecting them.

Edge weights are the minimal travel times necessary to travel the road between the two endpoint vertices, determined by speed limits and typical traffic conditions.

This representation captures the connectivity and travel times of the road network.

We are interested in designing an administrative division of a country in terms of the road graph - finding centers of TUs and subsets of vertices forming each TU.

The main mathematical model — the road graph

We model the country as an undirected edge-weighted *road graph*, where vertices (nodes) represent towns/settlements/intersections and edges represent roads connecting them.

Edge weights are the minimal travel times necessary to travel the road between the two endpoint vertices, determined by speed limits and typical traffic conditions.

This representation captures the connectivity and travel times of the road network.

We are interested in designing an administrative division of a country in terms of the road graph - finding centers of TUs and subsets of vertices forming each TU.

The main mathematical model — the road graph

We model the country as an undirected edge-weighted *road graph*, where vertices (nodes) represent towns/settlements/intersections and edges represent roads connecting them.

Edge weights are the minimal travel times necessary to travel the road between the two endpoint vertices, determined by speed limits and typical traffic conditions.

This representation captures the connectivity and travel times of the road network.

We are interested in designing an administrative division of a country in terms of the road graph - finding centers of TUs and subsets of vertices forming each TU.

The main mathematical model — the road graph

We model the country as an undirected edge-weighted *road graph*, where vertices (nodes) represent towns/settlements/intersections and edges represent roads connecting them.

Edge weights are the minimal travel times necessary to travel the road between the two endpoint vertices, determined by speed limits and typical traffic conditions.

This representation captures the connectivity and travel times of the road network.

We are interested in designing an administrative division of a country in terms of the road graph - finding centers of TUs and subsets of vertices forming each TU.

Basic definitions from graph theory

We need basic relevant notions about undirected edge-weighted graphs. Let $G = (V, E, w)$ be an undirected edge-weighted graph:

- the weight of a path in undirected edge-weighted graph is the sum of the weights of all edges in that path — in our case, time to travel the route between two towns/vertices;
- the distance between two vertices $u \in V$ and $v \in V$, $d(u, v)$, is defined as the weight of a (u, v)-path of minimal weight — in our case, the minimal time to travel between two towns. d is a metric in V.

Basic definitions from graph theory

We need basic relevant notions about undirected edge-weighted graphs. Let $G = (V, E, w)$ be an undirected edge-weighted graph:

- the weight of a path in undirected edge-weighted graph is the sum of the weights of all edges in that path — in our case, time to travel the route between two towns/vertices;
- the distance between two vertices $u \in V$ and $v \in V$, $d(u, v)$, is defined as the weight of a (u, v)-path of minimal weight — in our case, the minimal time to travel between two towns. d is a metric in V.

Basic definitions from graph theory

We need basic relevant notions about undirected edge-weighted graphs. Let $G = (V, E, w)$ be an undirected edge-weighted graph:

- the weight of a path in undirected edge-weighted graph is the sum of the weights of all edges in that path — in our case, time to travel the route between two towns/vertices;
- the distance between two vertices $u \in V$ and $v \in V$, $d(u, v)$, is defined as the weight of a (u, v)-path of minimal weight — in our case, the minimal time to travel between two towns. d is a metric in V.

Basic definitions from graph theory

- eccentricity of $v \in V$, $e(v)$, is defined as

$$\max_{x \in V} d(v,x)$$

— in our case, the time to reach the farthest point from v;

- radius of G is

$$r(G) := \min_{v \in V} e(v),$$

- center of G is $Z(G) := G[\mathcal{Z}]$, where $\mathcal{Z} := \{x \in V | e(x) = r(G)\}$ — vertices with minimal eccentricity ("centered").

Basic definitions from graph theory

- eccentricity of $v \in V$, $e(v)$, is defined as

$$\max_{x \in V} d(v, x)$$

— in our case, the time to reach the farthest point from v;

- radius of G is

$$r(G) := \min_{v \in V} e(v),$$

- center of G is $Z(G) := G[\mathcal{Z}]$, where $\mathcal{Z} := \{x \in V | e(x) = r(G)\}$ — vertices with minimal eccentricity (”centered”).

Basic definitions from graph theory

- eccentricity of $v \in V$, $e(v)$, is defined as

$$\max_{x \in V} d(v, x)$$

— in our case, the time to reach the farthest point from v;

- radius of G is

$$r(G) := \min_{v \in V} e(v),$$

- center of G is $Z(G) := G[\mathcal{Z}]$, where $\mathcal{Z} := \{x \in V | e(x) = r(G)\}$ — vertices with minimal eccentricity ("centered").

Road graph of Latvia

Our data for demonstrating an implementation of our approach is an edge-weighted road graph of Latvia $\Gamma = (V, E, t)$.

Using Google Maps, an undirected edge-weighted graph having 1067 vertices and 1753 edges has been constructed.

The edge-weight function $t : E \longrightarrow \mathbb{R}^+$ is the travel time by motor vehicle in minutes between the end vertices as recorded by Google Maps in October-November 2023.

Road graph of Latvia

Our data for demonstrating an implementation of our approach is an edge-weighted road graph of Latvia $\Gamma = (V, E, t)$.

Using Google Maps, an undirected edge-weighted graph having 1067 vertices and 1753 edges has been constructed.

The edge-weight function $t : E \longrightarrow \mathbb{R}^+$ is the travel time by motor vehicle in minutes between the end vertices as recorded by Google Maps in October-November 2023.

Road graph of Latvia

Our data for demonstrating an implementation of our approach is an edge-weighted road graph of Latvia $\Gamma = (V, E, t)$.

Using Google Maps, an undirected edge-weighted graph having 1067 vertices and 1753 edges has been constructed.

The edge-weight function $t : E \longrightarrow \mathbb{R}^+$ is the travel time by motor vehicle in minutes between the end vertices as recorded by Google Maps in October-November 2023.

Errors

1. A typical error of vertex coordinates is the geographical radius of the dot denoting the town in Google Maps. The implied travel time errors are less than 1 min - the minimal Google Maps time unit.
2. The real non-stop travel time given by Google Maps between two vertices having at least 1 intermediate vertex in their shortest path is less than the sum of travel times for all edges in the shortest path.

Since we model travel using edge weights we have to implicitly assume that travel includes stopping in all intermediate vertices. This error increases with the number of intermediate vertices.

Errors

1. A typical error of vertex coordinates is the geographical radius of the dot denoting the town in Google Maps. The implied travel time errors are less than 1 min - the minimal Google Maps time unit.
2. The real non-stop travel time given by Google Maps between two vertices having at least 1 intermediate vertex in their shortest path is less than the sum of travel times for all edges in the shortest path.

Since we model travel using edge weights we have to implicitly assume that travel includes stopping in all intermediate vertices. This error increases with the number of intermediate vertices.

Errors

1. A typical error of vertex coordinates is the geographical radius of the dot denoting the town in Google Maps. The implied travel time errors are less than 1 min - the minimal Google Maps time unit.
2. The real non-stop travel time given by Google Maps between two vertices having at least 1 intermediate vertex in their shortest path is less than the sum of travel times for all edges in the shortest path.

Since we model travel using edge weights we have to implicitly assume that travel includes stopping in all intermediate vertices. This error increases with the number of intermediate vertices.

Centered partitions of the vertex set

Suppose we are given a road graph - an undirected edge-weighted graph $G = (V, E, w)$.

Definition

TU is a pair (V', c), where $V' \subseteq V$ and $c \in V'$ is a distinguished element - the *TU center*.

Definition

A *centered partitions of* V:

$$\mathbf{P} = \{(V_1, c_1), ..., (V_m, c_m)\}, \;\; c_i \in V_i, \; \bigcup_{i=1}^{m} V_i = V, V_i \cap V_j = \varnothing.$$

Centered partitions of the vertex set

Suppose we are given a road graph - an undirected edge-weighted graph $G = (V, E, w)$.

Definition

TU is a pair (V', c), where $V' \subseteq V$ and $c \in V'$ is a distinguished element - the *TU center.*

Definition

A *centered partitions of* V:

$$\mathbf{P} = \{(V_1, c_1), ..., (V_m, c_m)\},\ c_i \in V_i,\ \bigcup_{i=1}^{m} V_i = V, V_i \cap V_j = \varnothing.$$

Centered partitions of the vertex set

Suppose we are given a road graph - an undirected edge-weighted graph $G = (V, E, w)$.

Definition

TU is a pair (V', c), where $V' \subseteq V$ and $c \in V'$ is a distinguished element - the *TU center.*

Definition

A *centered partitions of* V:

$$\mathbf{P} = \{(V_1, c_1), ..., (V_m, c_m)\},\ c_i \in V_i,\ \bigcup_{i=1}^{m} V_i = V, V_i \cap V_j = \varnothing.$$

From centers to TU - Voronoi partition, the first requirement

Suppose we are given a set of centers $S \subseteq V$. How do we define TUs with these centers?

For any $c \in S$, define its TU $V_S(c)$ containing all vertices for which c is reachable faster than any other center vertex c'.

Definition

For any $c \in S$ define the vertex subset $V_S(c)$ as the Voronoi cell of c as an element of S with respect to the d-metric:

$$V_S(c) = \{v \in V | d(v, c) \leq d(v, c'), \forall\ c' \in S, c' \neq c\}.$$

The centered partition $\mathbf{V}(S) = \{(V_S(c), c)\}_{c \in S}$ - *the centered Voronoi partition* for S.

From centers to TU - Voronoi partition, the first requirement

Suppose we are given a set of centers $S \subseteq V$. How do we define TUs with these centers?

For any $c \in S$, define its TU $V_S(c)$ containing all vertices for which c is reachable faster than any other center vertex c'.

Definition

For any $c \in S$ define the vertex subset $V_S(c)$ as the Voronoi cell of c as an element of S with respect to the d-metric:

$$V_S(c) = \{v \in V | d(v,c) \leq d(v,c'), \forall\, c' \in S, c' \neq c\}.$$

The centered partition $\mathbf{V}(S) = \{(V_S(c), c)\}_{c \in S}$ - *the centered Voronoi partition* for S.

From centers to TU - Voronoi partition, the first requirement

Suppose we are given a set of centers $S \subseteq V$. How do we define TUs with these centers?

For any $c \in S$, define its TU $V_S(c)$ containing all vertices for which c is reachable faster than any other center vertex c'.

Definition

For any $c \in S$ define the vertex subset $V_S(c)$ as the Voronoi cell of c as an element of S with respect to the d-metric:

$$V_S(c) = \{v \in V | d(v, c) \leq d(v, c'), \forall\ c' \in S, c' \neq c\}.$$

The centered partition $\mathbf{V}(S) = \{(V_S(c), c)\}_{c \in S}$ - *the centered Voronoi partition* for S.

From centers to TU - Voronoi partition, the first requirement

Suppose we are given a set of centers $S \subseteq V$. How do we define TUs with these centers?

For any $c \in S$, define its TU $V_S(c)$ containing all vertices for which c is reachable faster than any other center vertex c'.

Definition

For any $c \in S$ define the vertex subset $V_S(c)$ as the Voronoi cell of c as an element of S with respect to the d-metric:

$$V_S(c) = \{v \in V | d(v, c) \leq d(v, c'), \forall\ c' \in S, c' \neq c\}.$$

The centered partition $\mathbf{V}(S) = \{(V_S(c), c)\}_{c \in S}$ - *the centered Voronoi partition* for S.

From centers to TU - Voronoi partitions, the first requirement

Justification for defining territorial division as the Voronoi partition — minimization of the graph distance (travel time) to the TU center for each vertex.

Voronoi diagrams (Dirichlet, Thiessen) have been considered for use in territorial management and planning (WEB, [2025]; Ricca et al., [2008]).

From centers to TU - Voronoi partitions, the first requirement

Justification for defining territorial division as the Voronoi partition — minimization of the graph distance (travel time) to the TU center for each vertex.

Voronoi diagrams (Dirichlet, Thiessen) have been considered for use in territorial management and planning (WEB, [2025]; Ricca et al., [2008]).

Minimizing center bias within each TU, the second requirement

To minimize bias within TUs, we impose an additional requirement that the TU center c must belong to the graph center of its Voronoi subgraph.

Equivalently, the eccentricity of each TU center c in the TUs Voronoi subgraph is equal to the radius of the TU subgraph.

Justification — to ensure that each center is positioned centrally within its TU (its Voronoi cell). It reduces bias or unfairness.

Minimizing center bias within each TU, the second requirement

To minimize bias within TUs, we impose an additional requirement that the TU center c must belong to the graph center of its Voronoi subgraph.

Equivalently, the eccentricity of each TU center c in the TUs Voronoi subgraph is equal to the radius of the TU subgraph.

Justification — to ensure that each center is positioned centrally within its TU (its Voronoi cell). It reduces bias or unfairness.

Minimizing center bias within each TU, the second requirement

To minimize bias within TUs, we impose an additional requirement that the TU center c must belong to the graph center of its Voronoi subgraph.

Equivalently, the eccentricity of each TU center c in the TUs Voronoi subgraph is equal to the radius of the TU subgraph.

Justification — to ensure that each center is positioned centrally within its TU (its Voronoi cell). It reduces bias or unfairness.

The need for optimization requirements

If the set of TU centers S is chosen then we define TUs uniquely as Voronoi cells of S.

The crucial step is to set optimization conditions and find a solution set S_{opt} of TU centers. This would trivially imply a partition of V - its Voronoi partition $\mathbf{V}(S_{opt})$.

The need for optimization requirements

If the set of TU centers S is chosen then we define TUs uniquely as Voronoi cells of S.

The crucial step is to set optimization conditions and find a solution set S_{opt} of TU centers. This would trivially imply a partition of V - its Voronoi partition $\mathbf{V}(S_{opt})$.

Main parameter of TU and TU partitions

Definition

Radius of the TU (V_i, c_i):

$$r(V_i, c_i) := \max_{x \in V_i} d(x, c_i) = e(c_i)\Big|_{G[V_i]}.$$

Definition

Radius of a centered partition $\mathbf{P} = \{(V_1, c_1), ..., (V_m, c_m)\}, \ c_i \in V_i$:

$$r(\mathbf{P}) := \max_i r(V_i, c_i).$$

We think of radius of a territorial partition as its *badness*, would like to minimize it.

Main parameter of TU and TU partitions

Definition

Radius of the TU (V_i, c_i):

$$r(V_i, c_i) := \max_{x \in V_i} d(x, c_i) = e(c_i)\Big|_{G[V_i]}.$$

Definition

Radius of a centered partition $\mathbf{P} = \{(V_1, c_1), ..., (V_m, c_m)\},\ c_i \in V_i$:

$$r(\mathbf{P}) := \max_i r(V_i, c_i).$$

We think of radius of a territorial partition as its *badness*, would like to minimize it.

Main parameter of TU and TU partitions

Definition

Radius of the TU (V_i, c_i):

$$r(V_i, c_i) := \max_{x \in V_i} d(x, c_i) = e(c_i)\Big|_{G[V_i]}.$$

Definition

Radius of a centered partition $\mathbf{P} = \{(V_1, c_1), ..., (V_m, c_m)\}, \ c_i \in V_i$:

$$r(\mathbf{P}) := \max_i r(V_i, c_i).$$

We think of radius of a territorial partition as its *badness*, would like to minimize it.

Optimization condition, the third requirement

For a fixed number of TUs ($k = |S|$) we want to choose the center set S to minimize the radius of its Voronoi partition $\mathrm{r}(\mathbf{V}(\mathrm{S}))$:

$$\max_{c \in S, |S|=k} r(V_S(c), c) = r(\mathbf{V}(S)) \text{ is minimal.}$$

The motivation for this requirement is a drive to minimize the maximal "time burden" of TUs for a given number of TUs.

This would ensure that the TU radii are close (contribute to "fairness"). Minimization of the maximal TU radius automatically makes TU radius values close to each other.

Optimization condition, the third requirement

For a fixed number of TUs ($k = |S|$) we want to choose the center set S to minimize the radius of its Voronoi partition $\mathrm{r}(\mathbf{V}(\mathrm{S}))$:

$$\max_{c \in S, |S|=k} r(V_S(c), c) = r(\mathbf{V}(S)) \text{ is minimal.}$$

The motivation for this requirement is a drive to minimize the maximal "time burden" of TUs for a given number of TUs.

This would ensure that the TU radii are close (contribute to "fairness"). Minimization of the maximal TU radius automatically makes TU radius values close to each other.

Optimization condition, the third requirement

For a fixed number of TUs ($k = |S|$) we want to choose the center set S to minimize the radius of its Voronoi partition $\mathrm{r}(\mathbf{V}(\mathrm{S}))$:

$$\max_{c \in S, |S|=k} r(V_S(c), c) = r(\mathbf{V}(S)) \text{ is minimal.}$$

The motivation for this requirement is a drive to minimize the maximal "time burden" of TUs for a given number of TUs.

This would ensure that the TU radii are close (contribute to "fairness"). Minimization of the maximal TU radius automatically makes TU radius values close to each other.

Summarizing requirements

Requirements for an optimal territorial division:

1. Each TU is Voronoi cell;
2. center of each TU is its graph center;
3. partition radius - maximal TU radius, is minimal.

This is the *vertex k-center problem* with additional conditions. A special case of the *minimax facility location problem.*

Vertex k-center problem is a NP-hard problem. Number of flops $O(n^k)$, $n = |V|$, $k = |S|$. Approximation algorithms should be used.

Summarizing requirements

Requirements for an optimal territorial division:

1. Each TU is Voronoi cell;
2. center of each TU is its graph center;
3. partition radius - maximal TU radius, is minimal.

This is the *vertex k-center problem* with additional conditions. A special case of the *minimax facility location problem.*

Vertex k-center problem is a NP-hard problem. Number of flops $O(n^k)$, $n = |V|$, $k = |S|$. Approximation algorithms should be used.

Summarizing requirements

Requirements for an optimal territorial division:

1. Each TU is Voronoi cell;
2. center of each TU is its graph center;
3. partition radius - maximal TU radius, is minimal.

This is the *vertex k-center problem* with additional conditions. A special case of the *minimax facility location problem.*

Vertex k-center problem is a NP-hard problem. Number of flops $O(n^k)$, $n = |V|$, $k = |S|$. Approximation algorithms should be used.

Approximation algorithms

Steps of approximation algorithm:

- the greedy step — "farthest-first" search — start with a random vertex, choose next vertices maximizing minimal distance to the previously chosen vertices;
- to implement the second requirement - iteratively moving each center to the graph center of its Voronoi cell, recomputing Voronoi cells (these iterations significantly reduce the partition radius; converges after ≤ 5 iterations, can be proved);
- exhaustive search in graph neighbourhoods of centers (decreases the radius by only $5-10\%$, but takes a lot of time).

In our case, $n \sim 1000$ vertices, $k \leq 40$ centers, it is realistic to solve (get convergence, ~ 1 hour) the vertex k-center problem with the additional requirement.

Approximation algorithms

Steps of approximation algorithm:

- the greedy step — "farthest-first" search — start with a random vertex, choose next vertices maximizing minimal distance to the previously chosen vertices;
- to implement the second requirement - iteratively moving each center to the graph center of its Voronoi cell, recomputing Voronoi cells (these iterations significantly reduce the partition radius; converges after ≤ 5 iterations, can be proved);
- exhaustive search in graph neighbourhoods of centers (decreases the radius by only $5-10\%$, but takes a lot of time).

In our case, $n \sim 1000$ vertices, $k \leq 40$ centers, it is realistic to solve (get convergence, ~ 1 hour) the vertex k-center problem with the additional requirement.

Approximation algorithms

Steps of approximation algorithm:

- the greedy step — "farthest-first" search — start with a random vertex, choose next vertices maximizing minimal distance to the previously chosen vertices;
- to implement the second requirement - iteratively moving each center to the graph center of its Voronoi cell, recomputing Voronoi cells (these iterations significantly reduce the partition radius; converges after ≤ 5 iterations, can be proved);
- exhaustive search in graph neighbourhoods of centers (decreases the radius by only $5-10\%$, but takes a lot of time).

In our case, $n \sim 1000$ vertices, $k \leq 40$ centers, it is realistic to solve (get convergence, ~ 1 hour) the vertex k-center problem with the additional requirement.

Approximation algorithms

Steps of approximation algorithm:

- the greedy step — "farthest-first" search — start with a random vertex, choose next vertices maximizing minimal distance to the previously chosen vertices;
- to implement the second requirement - iteratively moving each center to the graph center of its Voronoi cell, recomputing Voronoi cells (these iterations significantly reduce the partition radius; converges after ≤ 5 iterations, can be proved);
- exhaustive search in graph neighbourhoods of centers (decreases the radius by only $5 - 10\%$, but takes a lot of time).

In our case, $n \sim 1000$ vertices, $k \leq 40$ centers, it is realistic to solve (get convergence, ~ 1 hour) the vertex k-center problem with the additional requirement.

Borders

The stated goal is to find TUs as sets of vertices. Eventually we have to draw borders.

The road graph is not enough to define borders as lines, therefore it is out of scope.

Nevertheless, for visualization we offer 2 methods –

- alpha shape method (classic, generalizes convex hull);
- border-edge midpoints method (may be innovative).

Borders

The stated goal is to find TUs as sets of vertices. Eventually we have to draw borders.

The road graph is not enough to define borders as lines, therefore it is out of scope.

Nevertheless, for visualization we offer 2 methods –

- alpha shape method (classic, generalizes convex hull);
- border-edge midpoints method (may be innovative).

Borders

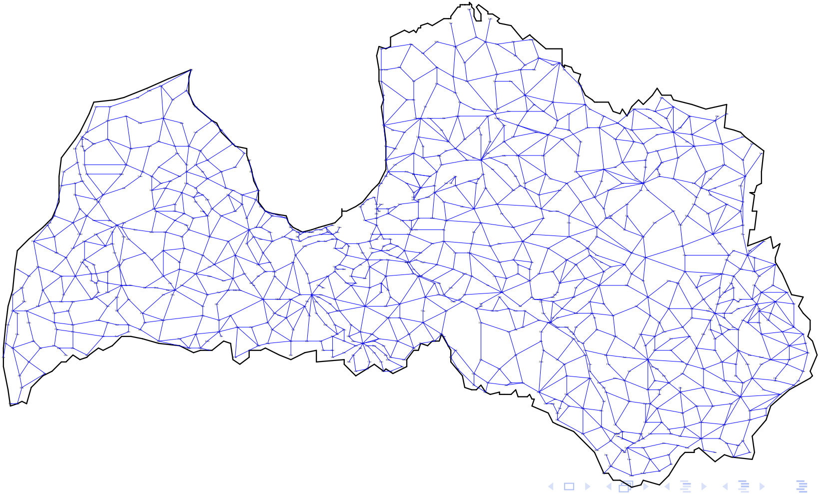
The stated goal is to find TUs as sets of vertices. Eventually we have to draw borders.

The road graph is not enough to define borders as lines, therefore it is out of scope.

Nevertheless, for visualization we offer 2 methods –

- alpha shape method (classic, generalizes convex hull);
- border-edge midpoints method (may be innovative).

The road graph of Latvia



$k = 1$

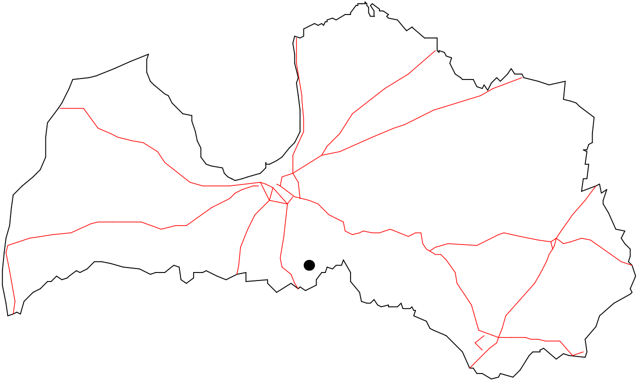
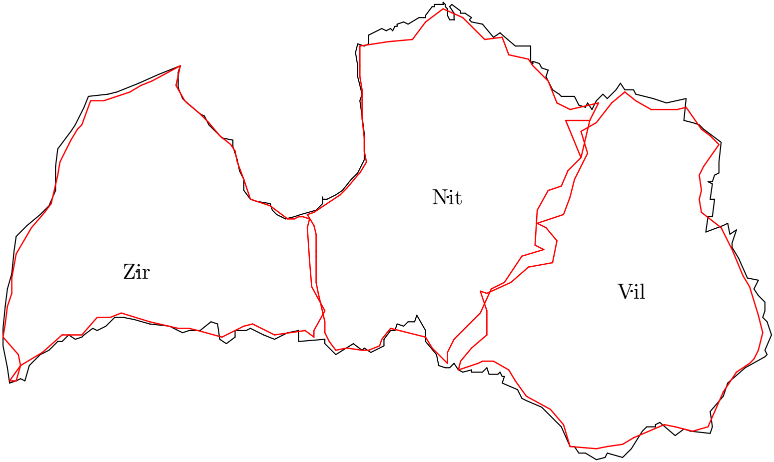
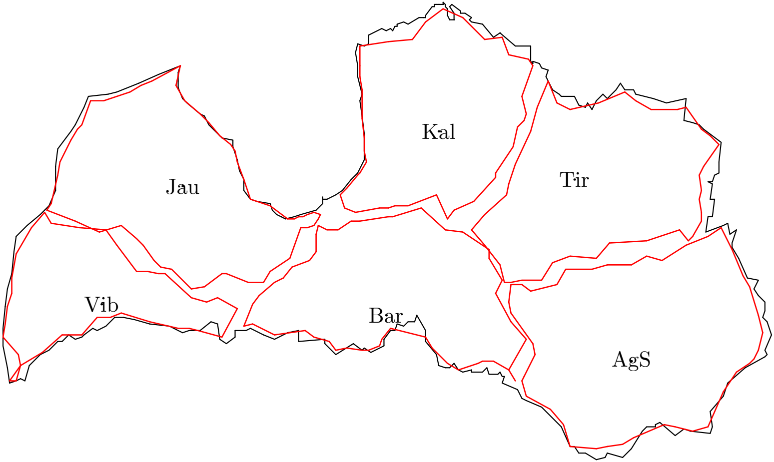


Figure: The case $k = 1$. The black dot is in the geographic position of $Z(\Gamma)$. Red lines are state main roads.

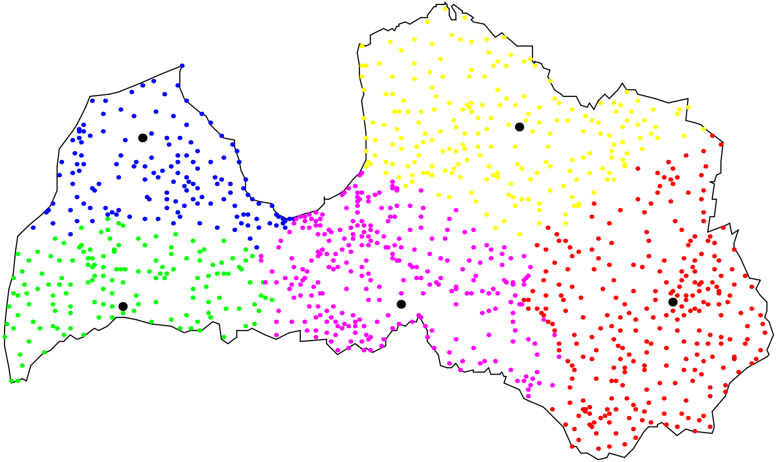
$k = 3$



$k = 6$



$k = 5$



$k = 15$

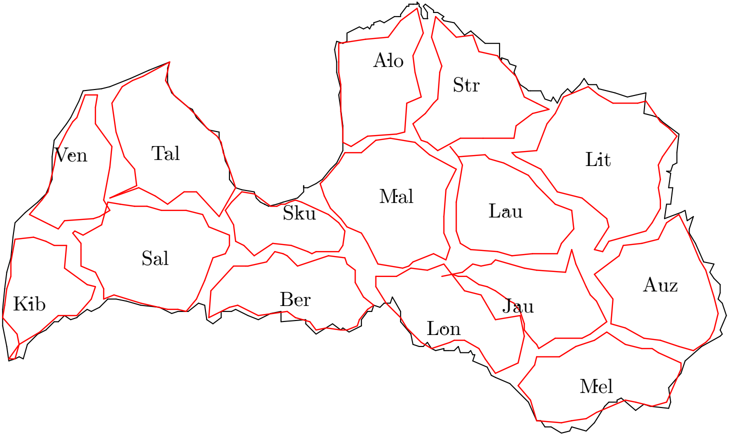


Figure: A case $k = 15$. The alpha shape view.

Conclusion

The proposed optimization approach has been shown to have the potential to improve the efficiency of the administrative structure in Latvia by reducing the number of TUs by 58% while preserving the maximal travel time to the TU center.

In future developments, additional edge weights and vertex weights can be added to the model to capture more road network, territorial and other features. Other characteristics such as population-weighted distance and distance weighted by socio-economic indicators, can be used in future iterations.

Conclusion

The proposed optimization approach has been shown to have the potential to improve the efficiency of the administrative structure in Latvia by reducing the number of TUs by 58% while preserving the maximal travel time to the TU center.

In future developments, additional edge weights and vertex weights can be added to the model to capture more road network, territorial and other features. Other characteristics such as population-weighted distance and distance weighted by socio-economic indicators, can be used in future iterations.

References I

- Bradshaw, M. J., & Keating, M. (2004) A conceptual framework for administrative territorial divisions. *Environment and Planning C: Government and Policy*, 22(3), 377-394.
- Church, R.L., Drezner, Z. (2022) Review of obnoxious facilities location problems. *Computers & Operations Research*, 138, 105468, ISSN 0305-0548, https://doi.org/10.1016/j.cor.2021.105468.
- Daugulis, P. (2020) Optimizing Administrative Divisions: A Vertex k-Center Approach for Edge-Weighted Road Graphs. *Baltic Journal of Modern Computing*, 12(2), 177–187.
- Erwig, M. (2000). The graph Voronoi diagram with applications. *Networks: An International Journal.* **36**(3), 156–163,

References II

Gawrychowski, P., Kaplan, H., Mozes, S., Sharir, M. and Weimann, O. (2018). Voronoi diagrams on planar graphs, and computing the diameter in deterministic $O(n^{5/3})$ time. *In* Proceedings of the twenty-ninth annual Symposium on Discrete Algorithms (SODA), 495–514. SIAM,

Hakimi, S.L., Labbe, M. (1988). A Locational Problem Associated with the Voronoi partition of a network. *In* ORSA/TIMS National Meeting (April). ORSA/TIMS,

Mehlhorn, K. (1988). A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters* **27**(3), 125–128,

References III

Ricca, F., Scozzari, A., Simeone, B, (2008). Weighted Voronoi region algorithms for political districting, *Mathematical and Computer Modelling*, 48 (9–10), 1468-1477, ISSN 0895-7177, https://doi.org/10.1016/j.mcm.2008.05.041

WEB Find my School, Victorian Government Department of Education, available from https://www.findmyschool.vic.gov.au/.