UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

Problem predstavitve objektov v psevdo-evklidskem prostoru

(Capacitated Facility location problem in pseudo-euclidean space)

Ime in priimek: Marko Palangetić
Študijski program: Matematika
Mentor: prof. dr. Andrej Brodnik
Somentor: asist. dr. Rok Požar

Koper, september 2016

Ključna dokumentacijska informacija

Ime in PRIIMEK: Marko Palangetić

Naslov zaključne naloge: Problem predstavitve objektov v psevdo-evklidskem prostoru

Kraj: Koper

Leto: 2016

Število listov: 36 Število slik: 11 Število tabel: 6

Število referenc: 9

Mentor: prof. dr. Andrej Brodnik

Somentor: asist. dr. Rok Požar

Ključne besede: Algoritmi, Optimizacija, Matematično programiranje

Math. Subj. Class. (2010): 65K05, 90C26, 90C27

Izvleček:
V tisti tezi bomo pokazali metodo za reševanje posebne oblike problema postavitve objektov. Motivacija za samo delo se nahaja v projektu Evropske Unije glede recikažo starega lesa kjer se želijo zgraditi tvornice za obdelavo tega lesa. Tudi druga motovacija za reševanje je da do zdej znani algoritmi za reševanje tega problema so prepočasni za velik nabor podatkov. Osnova za reševanje tega problema se nahaja v stardandem algoritmu za problem gručanja podatkov, kateri se modificira z uporabo linearnega programa za potrebe problema. Ker je program narejen za premik skozi metričen prostor tudi smo uporabili vgrajanje metričnih prostorov za transformacijo naš začetni splet akumulacijskih centrov v ustrezeno množico točk v več dimenzijskem evklidskem prostoru kjer lahko uporabljamo modificirani algoritem za gručanje podatkov. V resnici algoritem je kombinacija randomizacijske in hevristične pretrage. Algoritem je testiran za splet akumulacijskih centrov za države Avstrijo in Slovenijo.

Key words documentation

Name and SURNAME: Marko Palangetić

Title of final project paper: Capacitated facility location problem in pseudo-euclidean space

Place: Koper

Year: 2016

Number of pages: 36 Number of figures: 11 Number of tables: 6

Number of references: 9

Mentor: Prof. Andrej Brodnik, PhD

Co-Mentor: Assist. Rok Požar, PhD

Keywords: Algorithms, Optimization, Mathematical programming

Math. Subj. Class. (2010): 65K05, 90C26, 90C27

Abstract:

In this paper we will present method for solving one special form of facility location problem. Motivation for doing that lies in European project for wood recycling where we need to build facilities for wood processing. Also second motivation for solving this problem is that most of the famous algorithms for solving this problem are inefficient for big data set. Base for solving our problem is k-means method which is modified using linear program for purposes of problem. Because that algorithm is made for moving in metric space we also used metric embedding for transforming our initial network of accumulation centers into proper set of points in high dimensional Euclidean space where we can use modified k-means algorithm. Essentially algorithm is combination of randomization and heuristic search. Algorithm was tested for countries of Austria and Slovenia.

Acknowledgment

First of all I would like to express my deep gratitude to my parents for supporting me during whole period of my education. I would also like to thank to my mentor prof. Andrej Brodnik and co-mentor assist. prof. Rok Požar for their guidance, useful help and scientific advice for completing my final project paper. At the end, I wish to thank Faculty of Mathematics, Natural Sciences and information Technologies for given support trough scholarship and financial help for attending to many international competitions and conferences.

Contents

List of Tables

List of Figures

1 Introduction

1.1 General about FLP and CLFP

The facility location problem (shorter FLP), also known as location analysis, k-center problem or warehouse location problem is branch of mathematical programming, computational geometry and operations research. There are many variations of this problem, however the common part of them is that we want to find an optimal placement of facilities to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' facilities. Capacitated facility location problem (shorter CFLP) is special form of FLP where facilities which we want to place, are limited by material which can be transported to them.

1.2 Real Life Motivation

Efficient resource use is the core concept of cascading, which is a sequential use of a certain resource for different purposes. This means that the same unit of a resource is used for multiple high-grade material applications (and therefore sequestering carbon for a greater duration) followed by a final use for energy generation and returning the stored carbon to the atmosphere. Intelligent concepts for reuse and recycling of valuable materials at the end of single product life will reduce the amount of waste to be landfilled. In order to successfully implement the cascading concept it needs to be financially beneficial. This is not trivial to asses as it depends on a number of factors ranging from operation costs, legislation, logistics, etc. Wood that is accumulated is not immediately reusable, hence it needs to be sorted to separate the reusable wood from the rest. The criteria for sorting is mostly size and legislation that limits the type of wood viable for reuse. Additionally waste wood is contaminated with chemical compounds, metals, glass, etc. Hence it needs to be decontaminated. Both the sorting and decontamination process later the usable waste wood from the unusable. The unused waste wood will most likely get burned to produce energy or if legislation forbids it, used for landfill. An implementation of the reverse logistic for waste wood, would hence have to include facilities for sorting and decontamination. The logistics of transporting waste wood in-between facilities must also be included both in terms of costs and carbon emissions. One of the key challenges will be determining the optimum facility locations considering the reverse logistic chain, costs, and constraints involved. Accumulation sites are places where waste-wood gets accumulated. Some of the waste-wood is unusable and hence gets treated as junk. The rest gets transported to facilities, namely sorting facilities. Again, after sorting the waste-wood some of it can be burned if profitable, otherwise it is transported to decontamination facilities where unwanted material gets removed and the wood is prepared for further processing into products or sold as a raw material. At each step in the reverse logistics chain waste-wood can be burned and sold as energy. In some cases burning can be more profitable than transporting it to the next facility. [1]

1.3 Scientific motivation

Capacitated facility location problem is well researched area with lot of publications [10]. Namely CLFP in many forms can be formulated as mixed integer linear program (short MILP). Known algorithms are mostly based on modifications of that MILP. However branch and bound methods for solving it can be very slow in practical implementation and they are unusable for larger set of input data but they are giving result really close to optimal solution (in some cases it can be in ϵ neighborhood of optimal solution for arbitrary positive ϵ). For example MILP made for our problem, for data of size 1830 run solution on 6-core processor for two weeks and then crush. So we are forced to find new ways for calculating optimal solution. That improvement of run time will be payed by decreasing guarantee of that how much is our solution is near to optimal. So in thesis we will present fast algorithm for one form of CLFP where we will use randomization and where expected value of our result is more far from optimal solution than already known algorithms.

1.4

For simplified problem we use that facility for sorting and decontamination is at same place and that merged construction we call s-d facility. Also all wood from one accumulation center is transported to the some s-d facility so we do not have junk and burned wood on the beginning. Also we want to build s-d facilities at same land where is some accumulation center.

Before we strictly mathematical formulate our problem we will present its background. Our main problem for optimization is how to minimize costs of production and to maximize how much money can be obtained. We assume that amount of money which

can be obtained is fixed for one year because it is directly proportional to the yearly amount of wood which is accumulated in all accumulated centers. That amount we denote by C. So we only optimize costs of production. Number of s-d facilities which we will built we will denote by k and their capacity for processing by b
We can divide all costs into a few categories:

- Transportation costs, denoted by C_T. We may write $C_T = P_T \cdot T$ where P_T is a price of transport per wood unit per distance unit and T is amount of wood transported times distance made during transporting.
- Costs of loading and unloading, denoted by C_L and C_{UL} . Since we are transporting all wood from the accumulation centers we can say that these costs are proportional to value C, so we have that $C_L = P_L \cdot C$ and $C_{UL} = P_{UL} \cdot C$ where P_L and P_{UL} are prices for loading and unloading respectively per wood unit.
- Tax for emission of CO_2 denoted by C_{CAR}. In EU transporting companies need to pay tax for emission of $C0_2$. These costs are counted per kilometer. In our problem mileage depends on how many time truck will go from accumulation center to s-d facility and since that depends on how much wood is in that accumulation center we may say that $C0_2$ costs like transportation costs depend on T. We write $C_{CAR} = P_{CAR} \cdot T$ where P_{CAR} is a price for $C0_2$ emission per wood unit per distance unit.
- Costs of building one facility denoted by C_B, which depends of capacity of that facility so we will put $C_B = C_B(b)$.
- Operation costs denoted by C_{OP}.

With described notations write our objective function for optimization.

$$f(k, b) = (P_T + P_{CAR}) \cdot T + (C_L + C_{UL}) \cdot C + k \cdot C_B(b) + C_{OP}.$$

From objective function we exclude operation cost because it is relative and in practice it will depends on other costs which need to be minimized. Since $(C_L + C_{UL}) \cdot C$ is constant we will redefine our objective function as $f(k, b) = (P_T + P_{CAR}) \cdot T_{OPT}(k, b) + k \cdot C_B(b)$. Idea is calculate $f(k, b)$ for relevant k and b and choose the smallest one.$C_B(b)$ will be given as input so we look at it as constant also. At the end we got that objective function is value T times some constant, so our main problem is to optimize value T and further that value we will call transportation cost.

1.5 Problem description

Let A be the set of accumulation centers. As an input data we will have:

- $d : A \times A \to \mathbb{R}^+$ as distance function between points of A and
- $c : A \to \mathbb{R}^+$ as capacity function of points in A. Capacity function corresponds to amount of accumulated wood in accumulation center.
- Positive integer k as number of s-d facilities.
- Positive real b as bound of every s-d facility.

Our problem is to find subset $S \subseteq A$ of size k and function $w : A \times A \to \mathbb{R}^+$ such that:

$$T(S, w) = \sum_{x \in A} \sum_{y \in S} d(x, y)w(x, y) \text{ is minimized} \tag{1.1}$$

subject to:

$$\forall x \in A : \sum_{y \in S} w(x, y) = c(x) \tag{1.2}$$

$$\forall y \in S : \sum_{x \in A} w(x, y) \leq b. \tag{1.3}$$

Here we have

- $S \subseteq A, |S| = k$, positions of s-d facilities
- $w : A \times S \to \mathbb{R}^+$ describes how much wood is transported from one accumulation center to one s-d facility.
- $T(S, w)$ is transportation cost which depends on set S and mapping w.

Because all wood from accumulation centers will be processed in some of s-d facilities, holds constraint 3.2. Further because of limitation in production in every s-d facility, holds constraint 3.3.

We will have notations:

- S_{OPT} as optimal value of S.
- w_{OPT} as optimal value of w.
- Value T_{OPT} as optimal value of function T.

1.6 Structure of thesis

This thesis consist consist from 8 chapters including this one. Second chapter is about related work on our research. In that chapter we defined known problems and algorithms for solving which are used as base for construction of our solution. In third chapter we described solutions in case when A is in Euclidean space and distances are squares of Euclidean distances. In fourth chapter we applied that result for given A with geographic coordinates. Next chapter is about problems with that kind of approach because in practice, road distances are more offer used than geographic distances. In sixth chapter we present way for solving that kind of problem using modified algorithms for metric embedding problem. Seventh chapter is analysis of obtained results of combination of using Euclidean approach of solving and metric embedding. The last chapter is about ideas and plans for future research.

2 Related Work

In this chapter we will present papers and material used in our research.

2.1 K - means problem

This problem is very well known problem in clustering analysis and we can formulate it as follows. Let k be positive integer and $I \subset \mathbb{R}^n$ finte set. Find finite $J \subset \mathbb{R}^k, |J| = k$ such that:

$$\sum_{x \in I} \min_{y \in J} \|x - y\|_2^2 \text{ is minimized}$$

We want to determine k points in $\mathbb{R}^n$, which are centers of our clusters, such that we minimize sum of square of distances between each input point to the nearest cluster center. Cluster of points with respect to its center is defined such that for centers $y \in J$ we have set $CL_y\{x \in I : \min_{z \in J} \|x - z\|_2 = \|x - y\|_2\}$, that is the set of nearest points to cluster center y. The following theorem describes the complexity of k-means problem [8].

Theorem 2.1. *k-means problem in $\mathbb{R}^2$ is NP-hard.*

Remark 2.2. k-means problem in $\mathbb{R}^2$ is sometimes called Planar k-means problem.

Proof of this theorem exceeds level of this thesis. Basic idea is to prove that 3-SAT problem is polinomialy reducible to planar k-means problem. Still there is no known research results for arbitrary Euclidean space.

Since we have that k-means problem is NP hard, we do not know is there algorithm which can solve it in polynomial time. Because of that we introduce some heuristic methods for find approximative solution of it.

2.2 Standard k-means algorithm

This is well known heuristic approach for solving K-means problem and it is used mostly in data analysis. However because of its simplicity and elegance we will use it for as basic method for our optimization [2]. Algorithm is using iterations for improvement temporary result until it can.

Let $y_1^{(1)}, y_2^{(1)}, \ldots, y_k^{(1)}$ be some initial set of points. Exponent in brackets denotes time step in which we are working. Two main steps of algorithm are:

- Assignment step: In each time t and for each i we determine $CL_{y_i}^{(t)}$ with respect to centers $y_i^{(t)}$.
- Updating step: In this step we are denoting new centers of clusters as geometric center of every cluster writing that as:

$$\forall i \in \{1, 2...k\} : y_i^{(t+1)} = \frac{\sum_{x \in CL_i{}^{(t)}} x}{|CL_i^{(t)}|}$$

These two steps are iterated until points converge to some set of points which will not change if we repeat described iteration. Choosing geometric center in second step is that because it minimize sum of squares of distances between that point and rest of points in a cluster which guarantee better and better stage in each step. Since in every step we are getting better and better results, iteration converges to some local minimum which can be arbitrary far from global minimum. Correspondence between result of algorithm and optimal solution can give smart choosing of initial points For more details about local minimality and convergence we refer reader to Lemma 3.7 and Theorem 3.8, where we prove these two things for more general problems.

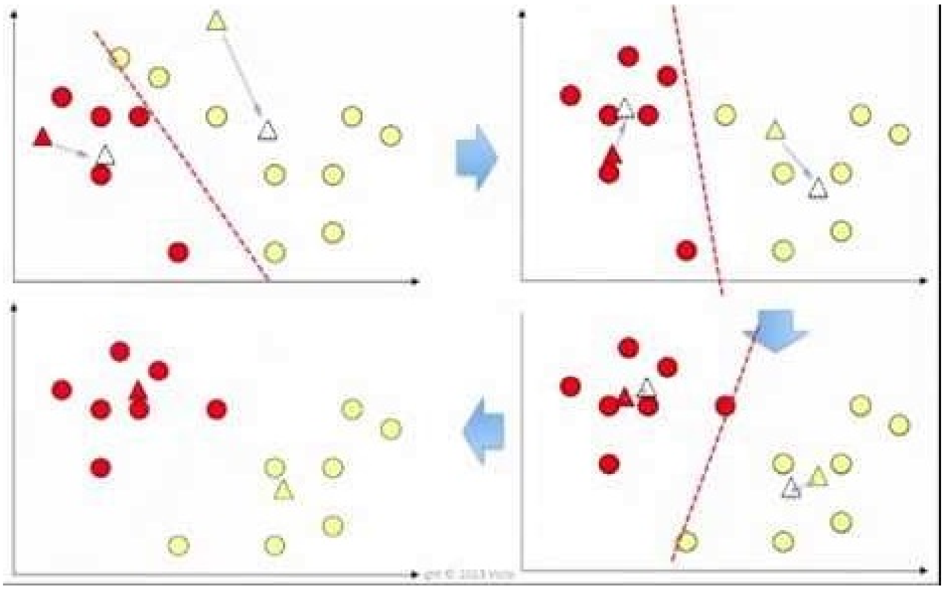


Figure 1: Example of k-means algorithm

To illustrate the potential of the k-means algorithm to perform arbitrarily poorly with respect to the objective function of minimizing the sum of squared distances of cluster points to the center of their assigned clusters, consider the example of four points in $\mathbb{R}^2$ that form an axis-aligned rectangle whose width is greater than its height. If $k = 2$ and the two initial cluster centers lie at the midpoints of the top and bottom line segments of the rectangle formed by the four input points, the k-means algorithm converges immediately, without moving these cluster centers. Consequently, the two bottom input points are clustered together and the two input points forming the top of the rectangle are clustered together. Because of that we obtained the clustering which is not an optimal because the width of the rectangle is greater than its height. Now, consider stretching the rectangle horizontally to an arbitrary width. The standard k-means algorithm will continue to cluster the points non-optimal, and by increasing the horizontal distance between the two data points in each cluster, we can make the algorithm perform arbitrarily poorly with respect to the k-means objective function.

2.3 Smart choosing of input points - k-means++

Since we do not know anything how bad or god can be obtained result with k-means algorithm we use randomized way to denote initial set of points such that we have some bounds on expected value of optimal solution. Idea is to randomly choose points from input set of points such that we want that dispersion of chosen points is high with some high probability. We can do it as:

1. Choose one center uniformly at random from among the input points.
2. For each input point x, compute $D(x)$, the distance between x and the nearest center that has already been chosen.
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to $D(x)^2$ so
$$p_x = \frac{D(x)^2}{\sum_{x \in I} D(x)^2}$$
.
4. Repeat two previous steps until k centers have been chosen.

As optimization guarantee we have the following [6].
If we denote ϕ the value of objective function in our algorithm, and ϕ_{OPT} the optimal value then we have that at the end of k-means++ method we have:

$$E(\phi) \leq 8(\log k + 2)\phi_{OPT}.$$

2.4 Problem of metric embedding

Metric embedding can be very useful tool when we want to solve some discrete optimization problem in ”continuous” manner. Since tools described tools described in the previous sections considers moving of points in continuous space, we indeed have the later situation. The problem of metric embedding is defined as follows. Suppose that we have an finite metric space (X, d). We want to find a mapping $f : X \to \mathbb{R}^m$ for some m such that

$$D(f) = \max_{a,b \in X} \max\{\frac{d(a,b)}{\|f(a) - f(b)\|_2}, \frac{\|f(a) - f(b)\|_2}{d(a,b)}\}$$

is minimized. It can be generalized for an arbitrary norm. However, in our definition we take 2-norm. Operator D(f) is called a distortion of metric embedding. Loosely speaking distortion is greatest relative deviation of new distance corresponding to distance in original metric space. In theory $D(f)$ is most used measure for describing the quality of embedding. However, for experimental purposes, we also introduce the following measure.

$$E(f,c) = \frac{\sum_{\{a,b\} \in X, a \neq b} 1(\max\{\frac{d(a,b)}{\|f(a)-f(b)\|_2}, \frac{\|f(a)-f(b)\|_2}{d(a,b)}\} < c)}{|X|(|X|-1)/2}$$

The measure $E(f, c)$ represents how many relative errors is less than some constant. Because in most cases it is impossible to make perfect embedding these two described measures tell us how good an embedding is. The following example shows that it is impossible to make perfect embedding.

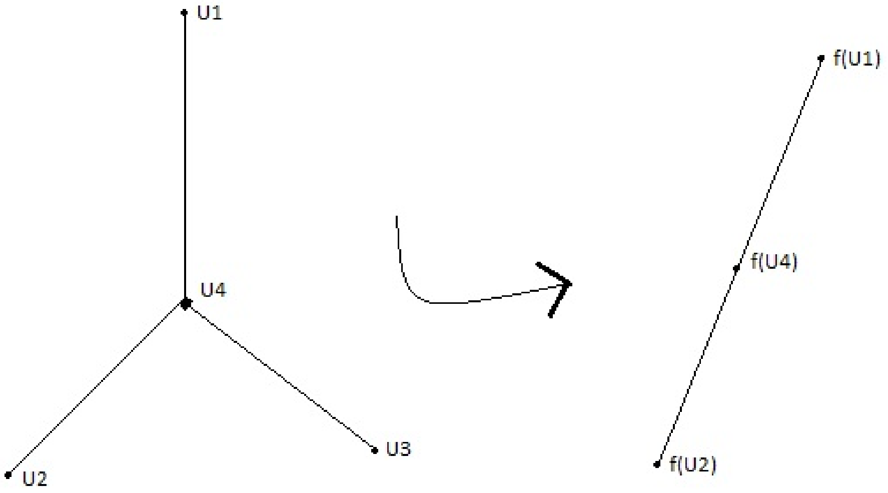


Figure 2: Counter example of embedding

On the left side on Figure 11 We have metric space where drawn distances are equal 1 and rest of them have distances like shortest paths between points. That is obviously metric space. Then, because $d(U_1, U_4) = 1$, $d(U_2, U_4) = 1$ and $d(U_1, U_2) = 2$, $f(U_1)$, $f(U_4)$ and$f(U_2)$ must be co-linear. Also by the same reasons $f(U_1)$, $f(U_4)$ and $f(U_2)$ must be co-linear. Because of distances between them we obtained that $f(U_2) = f(U_3)$, but since $d(U_2, U_3) = 2$ we have contradiction. So we got that there is no Euclidean space such that described metric space can be embedded into. There is a lot of work at this area and there are some algorithms developed for it but we will show that they are practically inefficient without some improvements.

2.5 Algorithms for metric embedding

The base for all famous results on this field is so called Frećhet embedding [3]. For given metric space (X, d) and positive integer m it can be described in this way:

- Choose subsets $S_1, S_2, \dots, S_m \subset X$
- Define embedding $f : X \to \mathbb{R}^m$ as $f(x) = (d(x, S_1), d(x, S_2), \dots, d(x, S_m))^T$ where $d(x, S) = \min_{s \in S} d(x, s)$ for $S \subseteq X$

This embedding have one nice property:

Lemma 2.3. *Let (X, d) be a finite metric space. Consider the Frećhet embedding f of(X, d) int r-dimensional Euclidean space equipped with 1-norm, for some sets $S_1, S_2 \dots S_m \subset X$ which correspond to the coordinates of the value of f. Then $\|f(x) - f(y)\|_1 \leq md(x, y)$*

Proof. We wish to show that for every $S \subseteq X$, $|d(x, S) - d(y, S)| \leq d(x, y)$. Let $d(y, S) = d(y, w)$ for some $w \in S$ (by definition of $d(x, S)$ such w exists). Also, by definition for every $w \in S$. $d(x, S) \leq d(x, w)$. Therefore $d(x, S) - d(y, S) \leq d(x, w) - d(y, w) \leq d(x, y)$ where last inequality follows from the triangular inequality. Now let $f(x) = (d(x, S_1), d(x, S_2), \dots, d(x, S_m))^T$ and $f(y) = (d(y, S_1), d(y, S_2), \dots, d(y, S_m))^T$. Then we have:

$$\|f(x) - f(y)\| = \sum_{i=1}^{m} |d(x, S_i) - d(y, S_i)| \leq \sum_{i=1}^{m} d(x, y) = m \cdot d(x, y).$$

□

This result holds also for 2-norm since $\|x\|_p \leq \|x\|_q$ for $p \geq q$ and Euclidean vector x. In described embedding we did not say anything about sets S_i and is there some intelligent way how to construct them.

Now we will present the most famous result on this field, Bourgian theorem.

Theorem 2.4. *Let (X, d) be an metric space on n points. Then there exists embedding f into p- norm Euclidean space, $p \geq 1$, of dimension in $O(\log n^2)$ such that $D(f) \in O(\log n)$*

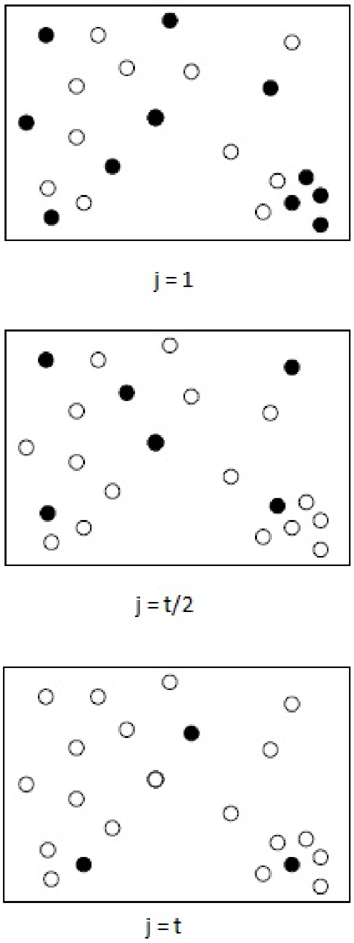
We will not give full proof about this result. It is construction proof and we will provide algorithm for it and intuition of proof.

Intuition behind the proof of theorem 3. and construction of algorithm 3: In Frećhet embeddings for each coordinate of the vectors we measure the distance of a point to a set. In Bourgian's theorem we will use Frećhet embeddings where the corresponding $A_{i,j}$ sets are constructed randomly by sampling independently the metric space with different probabilities $2^{-j}, j = 1, 2, \ldots \lceil \log n \rceil$ for many rounds $i = 1, 2, \ldots \Theta(\log n)$.

Algorithm 1: Bourgian embedding algorithm

```
Input: Metric space $(X, d)$
Output: Embedding $f$
$n = |X|$
$m = C \log n$ note: C is constant
$t = \lceil \log n \rceil$
for $j = 1$ to $t$ do
    for $i = 1$ to $m$ do
        Chose set $A_{i,j}$ with sampling probability of $2^{-j}$.
$f_{i,j} := d(x, A_{i,j})$
$f(x) = (d(x, A_{1,1}), d(x, A_{2,1}), \ldots, d(x, A_{m,1}), d(x, A_{1,2}), d(x, A_{2,2}), \ldots, d(x, A_{m,2}),$
$\ldots, d(x, A_{1,t}), d(x, A_{2,t}), \ldots, d(x, A_{m,t}))^T$
return $f$;
```

Then, we will show that with positive probability there exists an embedding which satisfies the requirements of the theorem. Clearly the same embedding must "work well" for the distance of every pair of points in the metric space. Hence, the reason why we use different probabilities (to sample points) has to do with the "structure" of the metric space. Also, for the same probability (used to independently sample elements from the metric space) we construct several sets. In figure 1 we give an intuitive example. Although, we use the plane to somehow refer to the notion of distance, keep in mind that the metric space is not (necessarily) Euclidean and drawing on the plane is done just for the sake of this intuitive demonstration. Before getting to the proof let us give some more intuition regarding why we need the two extreme cases, where the sampling probability is $\frac{1}{2}$ and $\frac{1}{n}$. Consider two points x, y to be far apart in the line. In one extreme we choose elements independently with probability $\frac{1}{2}$. In this case with high probability $A_{i,1}$ will contain points close both to x and to y ("no matter" how many times we will sample with the same sampling frequency). Therefore, we expect $|d(x, A_{i,j}) - d(y, A_{i,j})|$ to hardly contribute to $\|f(x) - f(y)\|_1$ (where f is the Frećhet embedding we are talking about) - actually in the example in the figure 4 the contribution is zero. In the other extreme the probability is $\frac{1}{n}$. In this case (if we sample with the same frequency for a sufficient number of times) with high probability we will have few points in $A_{i,j}$ which are close to x (or to y but not both). In this case $|d(x, A_{i,j} - d(y, A_{i,j}|$ is going to be close to $d(x, y)$.



j = 1

j = t/2

j = t

Figure 3: Example of Bourgian sampling: Black dots represents points sampled and included into set $A_{i,j}$ for particular j

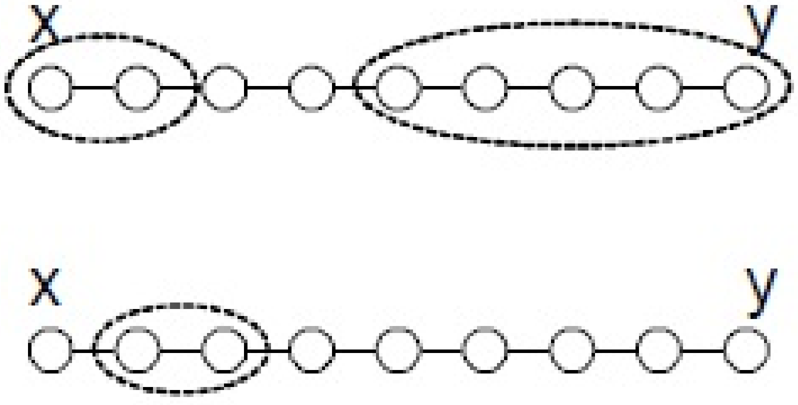


Figure 4: Example of Bourgian sampling 2: The rounded points represents points chosen into set $A_{i,j}$. Top: with probability $\frac{1}{2}$. Bottom: with probability $\frac{1}{2}$

Complete proof of Bourgian theorem can be found in paper [5].

There are also other theoretical results on this field which are in forms in existence theorems and they are not useful in algorithmic way.

- First lemma is about how to decrease dimension of Euclidean space in which we are embedding.

  Lemma 2.5. *We have given* $0 < \epsilon < 1$*, set* X *of* m *points in* $\mathbb{R}^N$ *for some* N *and number* $n \geq \frac{8 \ln m}{\epsilon^2}$*. Then there is embedding* $f : \mathbb{R}^N \to \mathbb{R}^n$*. such that:*

  $$(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2.$$

  for all $u, v \in X$

  Prof can be found in [4] Using this lemma we can improve Bourgian theorem decreasing dimension of Euclidean space in which we are embedding from $O(\log n^2)$ to $O(\log n)$, but this is only theoretical result of existence of such embedding.

- For second interesting lemma we need to define an average distortion $AVD(f)$ as

  $$AVD(f) = \frac{\sum_{\{a,b\} \in X, a \neq b} \max\{\frac{d(a,b)}{\|f(a)-f(b)\|_2}, \frac{\|f(a)-f(b)\|_2}{d(a,b)}\}}{|X|(|X|-1)/2}$$

  It is simply an average of distortions of all pairs of points from X. It is also one of measures of quality of embedding. Next lemma provides existence of embeddings with low average distortion.

Lemma 2.6. *For every finite metric space of size n there exist embedding of it into Euclidean space with maximal distortion in $O(\log n)$ and average distortion $O(1)$.*

More about this lemma can be found in [9]. This lemma is also not useful in practice since it is only existence lemma.

.

3 Euclidean approach for solving

In this chapter we will suppose that $A \subset \mathbb{R}^n$, and that $d(x, y) = \|x - y\|_2^2$. Choosing this as our distance function instead of $d(x, y) = \|x - y\|_2$ is directly connected with k-means problem because there we also have squares of distances instead of ordinary distance. Also here we do not want to have that $S \subseteq A$. Now as input we have set of coordinates of set A instead of given d. So our Euclidean version of problem is defined as follows. Find set $S \subseteq \mathbb{R}^{\ltimes}$ of size k and function $w : A \times A \to \mathbb{R}^+$ such that:

$$T(S, w) = \sum_{x \in A} \sum_{y \in S} w(x, y) \|x - y\|_2^2 \text{ is minimized} \tag{3.1}$$

subject to:

$$\forall x \in A : \sum_{y \in S} w(x, y) = c(x) \tag{3.2}$$

$$\forall y \in S : \sum_{x \in A} w(x, y) \leq b. \tag{3.3}$$

Main idea here is using already explain clustering methods to construct efficient algorithm to solve this optimization. The difference between K-means problem and Euclidean case of our problem is that in our problem we introduced now capacities on points and bound b. For easier understanding, bound b looked from view of k-means problem, naively can be explained as measure which represents how many points can be in one cluster. Because we did not mentioned anything about restrictions of that type in definition of k-means problem we will for beginning suppose that $b = \infty$ and try to extend k-means algorithm adding only capacities of points.

3.1 Solving problem for $b = \infty$

For fixed S we have the following inequality

$$\sum_{x \in A} \sum_{y \in S} w(x, y) \|x - y\|_2^2 \geq \sum_{x \in A} \sum_{y \in S} w(x, y) \min_{y \in S} \|x - y\|_2^2$$

$$= \sum_{x \in A} \min_{y \in S} \|x - y\|_2^2 \sum_{y \in S} w(x, y) = \sum_{x \in A} c(x) \min_{y \in S} \|x - y\|_2^2$$

Equality holds if and only if $w(x, y) = c(x)$ for $x \in CL_y$ and 0 otherwise. This conclusion means that if we fixed our s-d facilities, the best way of transporting resources is that that from one accumulation center all wood we need to transport to the nearest s-d facility. This is logic since we do not have any bounds for now how much wood can be processed in one that facility. So we got some similar form which we used as objective function in k-means problem: Find subset S of size k such that

$$\sum_{x \in A} c(x) \min_{y \in S} \|x - y\|_2^2$$

is minimized. We use similar approach as in standard k-means clustering algorithm using locally the best solution. At the beginning we will use k-means++ method for initial seeding of cluster centers but with one modification. In third step of method we will make our distribution based on values $c(x)D(x)^2$ instead of $D(x)^2$. We will call this seeding weighted k-means++, and points with defined $c(x)$.

Definition 3.1. Center of mass or geometric mean of set of weighted points $X \subset \mathbb{R}^k$ is:

$$m(X) = \frac{\sum_{x \in X} c(x) \cdot x}{\sum_{x \in X} c(x)}$$

About this seeding we will formulate next theorem which will be our main optimization guarantee theorem:

Theorem 3.2. *Let ϕ represents a value of our objective function after weighted k-means++ seeding and let ϕ_{OPT} be optimal value of it. Then we have guarantee:*

$$E(\phi) \leq 8(\log k + 2)\phi_{OPT}.$$

Proof. We will divide our proof in few lemmas because of easier following.

Lemma 3.3. *Let X be set of weighted points in Euclidean space, let $m(X)$ represents center of mass of those points, let c represents our weight function and let z be arbitrary point from that space. Then we have that:*

$$\sum_{x \in X} c(x)\|x - z\|^2 - \sum_{x \in X} c(x)\|x - m(X)\|^2 = \|m(X) - z\|^2 \sum_{x \in X} c(x)$$

Proof. Using formula from linear algebra that for vectors x and y holds $\|x + y\|^2 = \|x\|^2 + \|y\|^2 - 2\langle x, y\rangle$ where operator $\langle , \rangle$ is scalar product, we have $\sum_{x \in X} c(x)\|x - z\|^2 = \sum_{x \in X} c(x)\|x - m(X) + m(X) - z\|^2 = \sum_{x \in X} c(x)\|x - m(X)\|^2 + \|m(X) - z\|^2 \sum_{x \in X} c(x) - 2\sum_{x \in X} c(x)\langle x - m(X), m(X) - z\rangle$. Then we have that $\sum_{x \in X} c(x)\langle x - m(X), m(X) - z\rangle = \langle \sum_{x \in X} c(x)(x - m(X)), m(X) - z\rangle = \langle 0, m(X) - z\rangle = 0$ because of definition of $m(X)$. From that we obtained our formula. □

First to define some values: for subset X of A for fixed set S we define: $\phi(X) = \sum_{x\in A}\sum_{y\in S} w(x,y)\|x-y\|_2^2$ and $\phi_{OPT}(X)$ with same formula in case when S is optimal chosen.

Lemma 3.4. *Let X be cluster in optimal seeding and let we have clustering with one center which is chosen randomly from X with distribution denoted by weights of points from X. Then holds $E(\phi(X)) = 2\phi_{OPT}(X)$.*

Proof. Let x_0 be our chosen center. Then we have that

$$E(\phi(X)) = \sum_{x_0\in X}\frac{c(x_0)}{\sum_{y\in X}c(y)}\sum_{x\in X}c(x)\|x-x_0\|_2^2 =$$

$$\sum_{x_0\in X}\frac{c(x_0)}{\sum_{y\in X}c(y)}(\sum_{x\in X}c(x)\|x-m(X)\|^2 + \|m(X)-x_0\|^2\sum_{x\in X}c(x)) =$$

$$2\sum_{x\in X}c(x)\|x-m(X)\|^2 = 2\phi_{OPT}(x)$$

. □

Next lemma is analog of Lemma 3.4 but for further seeding.

Lemma 3.5. *Let X be cluster in optimal seeding and let we have clustering where we add one center from X using weighted k-means++ method. Then we have that: $E(\phi(X)) \leq 8\phi_{OPT}(X)$*

Proof. Probability to choose point x_0 from X as next center is equal: $\frac{c(x_0)D(x_0)^2}{\sum_{x\in X}c(x)D(x)^2}$. Furthermore after choosing some other point x from X will have contribution to objective function with $c(x)\min(D(x),\|x-x_0\|)^2$. So we have that

$$E(\phi(X)) = \sum_{x_0\in X}\frac{c(x_0)D(x_0)^2}{\sum_{x\in X}c(x)D(x)^2}\sum_{x\in X}c(x)\min(D(x),\|x-x_0\|)^2$$

. From triangle inequality we have that $D(x_0) \leq D(x) + \|x-x_0\|$. Using power mean inequality we have that: $D(x_0)^2 \leq (D(x)+\|x-x_0\|)^2 \leq 2D(x)^2+2\|x-x_0\|^2$ Multiplying with $c(x)$ we have $c(x)D(x_0)^2 \leq 2c(x)D(x)^2 + 2c(x)\|x-x_0\|^2$. With summing over x we obtain : $D(x_0)^2 \leq 2\frac{\sum_{x\in X}c(x)D(x)^2}{\sum_{x\in X}c(x)} + 2\frac{\sum_{x\in X}c(x)\|x-x_0\|^2}{\sum_{x\in X}c(x)}c(x)\|x-x_0\|^2$. So we have:

$$E(\phi(X)) \leq \frac{2}{\sum_{x\in X}c(x)}\sum_{x_0\in X}\frac{c(x_0)\sum_{x\in X}c(x)D(x)^2}{\sum_{x\in X}c(x)D(x)^2}\sum_{x\in X}c(x)\min(D(x),\|x-x_0\|)^2+$$

$$\frac{2}{\sum_{x\in X}c(x)}\sum_{x_0\in X}\frac{c(x_0)\sum_{x\in X}c(x)\|x-x_0\|^2}{\sum_{x\in X}c(x)D(x)^2}\sum_{x\in X}c(x)\min(D(x),\|x-x_0\|)^2$$

$$\leq \frac{2}{\sum_{x\in X}c(x)}\sum_{x_0\in X}\frac{c(x_0)\sum_{x\in X}c(x)D(x)^2}{\sum_{x\in X}c(x)D(x)^2}\sum_{x\in X}c(x)\|x-x_0\|^2+$$

$$\frac{2}{\sum_{x\in X} c(x)} \sum_{x_0\in X} \frac{c(x_0)\sum_{x\in X} c(x)\|x-x_0\|^2}{\sum_{x\in X} c(x)D(x)^2} \sum_{x\in X} c(x)D(x)^2 =$$

$$= 2\sum_{x_0\in X} \frac{c(x_0)}{\sum_{x\in X} c(x)} \sum_{x\in X} c(x)\|x-x_0\|^2 + 2\sum_{x_0\in X} \frac{c(x_0)}{\sum_{x\in X} c(x)} \sum_{x\in X} c(x)\|x-x_0\|^2$$

$$= 8\phi_{OPT}(X).$$

Last step follows from previous lemma. □

Here we have shown guarantee only when we are choosing our centers from optimal clusters. Next lemma will finish our proof showing that weighted k-means++ is logarithmic competitive.

Lemma 3.6. *Let C be arbitrary clustering. Now from optimal solution divide set A on $c+u$ clusters. Denote them as A_u and A_c. We will call A_c covered clusters and A_u uncovered clusters. Now we will add $t \leq u$ clusters to clustering C with weighted k-means++ method and denote that as C'. Let ϕ be value of objective function which corresponds to clustering C and ϕ' value corresponds to C' Then we have that holds:*

$$E(\phi') \leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1+H_t) + \frac{u-t}{u}\phi(A_u).$$

Here H_t denotes harmonic sum, $H_t = 1 + \frac{1}{2} + \cdots + \frac{1}{t}$

Proof. Proof will go by induction. We will prove that if result holds for pairs $(t-1,u)$ and $(t-1,u-1)$, it will hold also for (t,u). For base of induction we will prove that statement holds for $(0,u), u > 0$ and for $(1,1)$. For case $(0,u)$ we have that $E(\phi') = \phi$ and because $\frac{u-t}{u} = 1 + H_t = 1$ we have to prove that $\phi \leq \phi + 8\phi_{OPT}(A_u)$ which is obvious. For case $(1,1)$ we have that for sure $\phi' \leq \phi$ in every case since adding center will decrease objective function value. In this case probability that new center will be chosen from uncovered cluster is $\frac{\phi(A_u)}{\phi}$. In that case using previous lemma we have that $E(\phi') = E(\phi'(A_c) + \phi'(A_u)) \leq E(\phi(A_c) + \phi'(A_u)) = \phi(A_c) + E(\phi'(A_u)) \leq \phi(A_c) + 8\phi_{OPT}(A_u)$. In other case we know that $E(\phi') \leq \phi$. So we have that $E(\phi') \leq \frac{\phi(A_u)}{\phi}(\phi(A_c) + 8\phi_{OPT}(A_u)) + \frac{\phi(A_c)}{\phi}\phi \leq (\phi(A_c) + 8\phi_{OPT}(A_u)) + \phi(A_c) = 2\phi(A_c) + 8\phi_{OPT}(A_u)$. Since in this case $1 + H_t = 2$ and $\frac{u-t}{u} = 0$ we proved second case of induction base. Now suppose that statement is true for $(t-1,u)$ and $(t-1,u-1)$. Here we distinguish two cases:

1. Here we will denote ϕ'_t as value of ϕ' after adding t centers. Here we suppose that next center is chosen from covered cluster. That can happen with probability $\frac{\phi(A_c)}{\phi}$. In that case for fixed u we will use that $\phi'_t \leq \phi'_{t-1}$ so we have that $E(\phi') = E(\phi'_t) \leq E(\phi'_{t-1}) \leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t+1}{u}\phi(A_u)$

2. Suppose that next center is chosen from uncovered cluster X. Probability for that is $\frac{\phi(X)}{\phi}$. We will use inductive hypothesis that statement holds for pair $(t-1, u-1)$ in way that we will mark chosen cluster as covered and with that we have that we have $u-1$ uncovered clusters and point is chosen from covered cluster. Now let p_x be conditional probability that for chosen cluster X we chose point x as our center and let ϕ_x be value of respective objective function. Then for conditional expectation we have that it is less than:

$$\sum_{x \in X} p_x((\phi(A_c) + \phi_x + 8\phi_{OPT}(A_u) - 8\phi_{OPT}(X))(1 + H_{t-1}) + \frac{u-t}{u-1}(\phi(A_u) - \phi(X)))$$

$$\leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t}{u-1}(\phi(A_u) - \phi(X)))$$

The last step follows because from previous lemma we have $\sum_{x \in X} p_x \phi_x \leq 8\phi_{OPT}(X)$. Now for conditional expectation that center is chosen from uncovered cluster we have that it is less than:

$$\sum_{X \text{ is cluster in } A_u} \frac{\phi(X)}{\phi(A_u)}(\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t}{u-1}(\phi(A_u) - \phi(X)))$$

$$\leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{1}{\phi(A_u)}\frac{u-t}{u-1}(\phi(A_u)^2 - \sum_{X \text{ is cluster in } A_u} \phi(X)^2)$$

$$\leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{1}{\phi(A_u)}\frac{u-t}{u-1}(\phi(A_u)^2 - \frac{1}{u}\phi(A_u)^2)$$

$$= (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t}{u}\phi(A_u)$$

The last inequality holds from power mean inequality:

$$\phi(A_u)^2 = (\sum_{X \text{ is cluster in } A_u} \phi(X))^2 \leq \frac{1}{u} \sum_{X \text{ is cluster in } A_u} \phi(X)^2$$

Now combining these two cases we have that:

$$E(\phi') \leq \frac{\phi(A_c)}{\phi}((\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t+1}{u}\phi(A_u)) +$$

$$\frac{\phi(A_u)}{\phi}((\phi(A_u) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t}{u}\phi(A_u))$$

$$= (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t}{u}\phi(A_u) + \frac{\phi(A_c)}{\phi}\frac{1}{u}\phi(A_u)$$

$$\leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1}) + \frac{u-t}{u}\phi(A_u) + \frac{1}{u}(\phi(A_c) + 8\phi_{OPT}(A_u))$$

$$\leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_{t-1} + \frac{1}{u}) + \frac{u-t}{u}\phi(A_u)$$

$$\leq (\phi(A_c) + 8\phi_{OPT}(A_u))(1 + H_t) + \frac{u-t}{u}\phi(A_u).$$

The last inequality holds from $\frac{1}{u} \leq \frac{1}{t}$. □

We will finish our proof by applying previous lemma with $t = u = k - 1$. We will denote as X cluster from optimal seeding where we will chose our first point. then we have that conditionally on ϕ holds

$$E(\phi'|\phi) \leq (\phi(X) + 8\phi_{OPT} - 8\phi_{OPT}(X))(1 + H_{k-1}).$$

So we have that after we again put expected value we have

$$E(\phi') = E(E(\phi'|\phi)) \leq (E(\phi(X)) + 8\phi_{OPT} - 8\phi_{OPT}(X))(1 + H_{k-1})$$

$$= (2\phi_{OPT}(X) + 8\phi_{OPT} - 8\phi_{OPT}(X))(1 + H_{k-1})$$

$$\leq 8\phi_{OPT}(1 + H_t).$$

Result holds from fact that $H_{k-1} \leq 1 + \log k$.

□

Assignment does not change since weights do not influence to the making clusters. Update steps will change to the weighted mean because we will later prove weighted mean is the best locally optimal option for choosing new center, so the updating step will be:

$$\forall i \in \{1, 2...k\} : y_i^{(t+1)} = \frac{\sum_{x \in CL_i{}^{(t)}} c(x) \cdot x}{\sum_{x \in CL_i{}^{(t)}} c(x)}$$

Since we explain weights we need to solve if we have bounds on our s-d facilities

3.2 Solving problem for $b < \infty$

Since b is not anymore infinity we need to go back to our original Euclidean form of problem.

- $\forall x \in A \sum_{y \in S} w(x, y) = c(x)$
- $\forall y \in S \sum_{x \in A} w(x, y) \leq b$
- $\sum_{x \in A} \sum_{y \in S} w(x, y) \|x - y\|_2^2$ is minimized

For fixed S this problem is nothing more than but a linear program. The difference between the last section and this one is that it is not necessary that from one accumulation center, all wood will be transported to the one s-d facility as it will be result of linear program. Idea here is to use the linear program as a part of assignment step of algorithm. That means that for temporary cluster centers it will determine how much wood will go from one accumulation center to one s-d facility. Then updating step will find weighted mean of points using obtained transport distribution. Speaking in words of clustering we do not can exactly form clusters since we do not have that from one accomulation center all wood will be transported to one s-d facility. So here we will introduce term fuzzy clusters where one element can belong to more than one cluster which correspond to that from one accumulation center wood can be transported to more s-d facilities so in some sense one accumulation center "belongs" to more s-d facilities.

Algorithm written in more structural way is shown in: 2

Algorithm 2: Euclidean CFLP

```
Input: Set $A$, positive integer $k$, positive real $b$
Output: Set $S$ and function $w$
Use weighted k-means++ in order to initialize set $S$ of size $k$
while there is no significant improvement among cluster centers in set $S$ do
    For given set $S$ use linear program to compute function $w$
    For each $i \leq k$ denote new centers as $y_i^{'} = \frac{\sum_{x \in A} w(x,y_i) \cdot x}{\sum_{x \in A} w(x,y)}$
Use linear program to compute final $w$
return $S, w$;
```

Lemma 3.7. *Let $x_1, x_2, \ldots, x_n$ be points in $\mathbb{R}^d$ space, and $c(x_1), c(x_2)...c(x_n)$ real weights on them. Then their center of mass minimize*

$$f(y) = \sum_{i=1}^{n} c(x_i) \|x_i - y\|_2^2$$

Proof. Compute gradient of function:

$$\nabla f(y) = \sum_{i=1}^{n} 2c(x_i)(x_i - y).$$

Only point where this gradient is equal to 0 is a $y = \frac{\sum_{i=1}^{n} c(x_i) \cdot x_i}{\sum_{i=1}^{n} c(x_i)}$. Since function f is unbounded above (if we take point far enough we can obtain arbitrary big value), it is bounded at bottom by 0, it is differentiable at every point and it has only one point where gradient is 0. That means that in that point we must have global minimum, so that means obtained point y is point which minimize weighted squared distances which ends our proof. □

Since the algorithm 2 is iterative, it is natural to ask whether it converges. The following theorem gives the answer.

Theorem 3.8. *Algorithm 3 converges.*

Proof. Let $S^{(t)}$ be temporary S at step t, and let $w^{(t)}$ be the function w computed at time t. Set $S^{(t+1)}$ we obtain from step 4 from algorithm **??**. Because we have center of mass using Lemma 3.7 we have that $T(w^t, S^{t+1}) \leq T(w^t, S^t)$. Since we use again linear program which minimize our objective function to get function we obtain new function w^{t+1} at step $t+1$ and we have $T(w^{t+1}, S^{t+1}) \leq T(w^t, S^{t+1})$. So we got that

$$T(w^{t+1}, S^{t+1}) \leq T(w^t, S^{t+1}) \leq T(w^t, S^t)$$

This means that sequence $\{T(w^{(t)}, S^{(t)})\}_{t=1}^{\infty}$ is non-increasing. Because every member of sequence is positive that means it is bounded from below. Thus sequence converges. □

4 Results in euclidean space

This whole purpose of using this methods is not directly applying it to the geographic coordinates of the accumulation centers but applying it on them after using some transformation. But we did that direct application to the country of Austria and it's 1840 accumulation centers and we obtained next results:

- At the Figure 8 there are results about dependency between transportation costs and number of s-d facilities for $b = \infty$

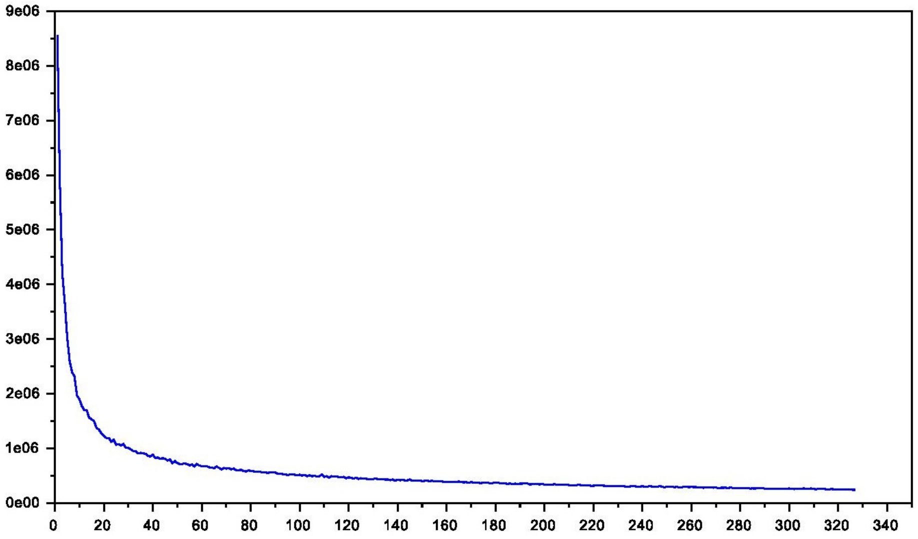
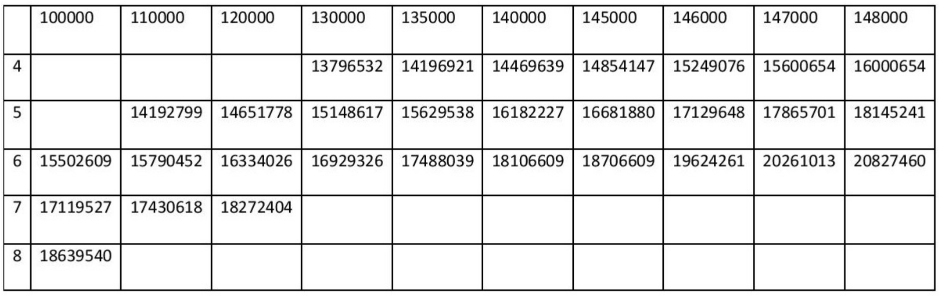


Figure 5: Graphic of transportation costs in unbounded case

- Here is shown table of summarize costs. First row represents bound b in kilograms and first column represents number of clusters. For Austria we have that at year 509000 kilograms is processed, and we put the prices as: $P_T = 0.13$, $P_{CAR} = 0.12$, $P_L = 0.5$, $P_{UL} = 0.5$ and some variations for building costs where for example for s-d facility of capacity 100000 we need 2 millions euros for it's building. As we can see from our table the best solution is to build 4 centers with capacity of 130000 tons.



	100000	110000	120000	130000	135000	140000	145000	146000	147000	148000
4				13796532	14196921	14469639	14854147	15249076	15600654	16000654
5		14192799	14651778	15148617	15629538	16182227	16681880	17129648	17865701	18145241
6	15502609	15790452	16334026	16929326	17488039	18106609	18706609	19624261	20261013	20827460
7	17119527	17430618	18272404							
8	18639540									

Figure 6: Results of Euclidean case

- Here we have example of visualized positions for $k = 5$ and $b = 120000$

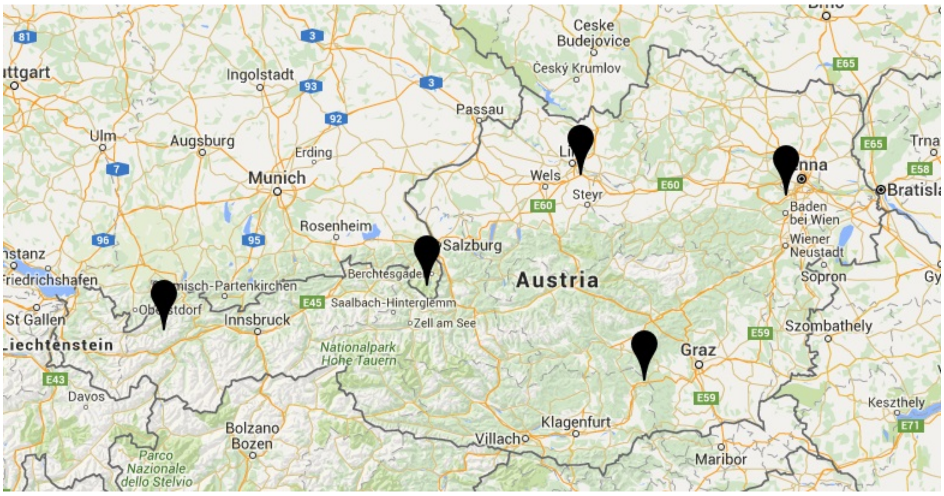


Figure 7: Example of one optimal solution

5 Problems with euclidean approach

Euclidean approach have lot of mistakes because Euclidean distances do not approximate in proper way real transport distances. One example is shown at Figure 8

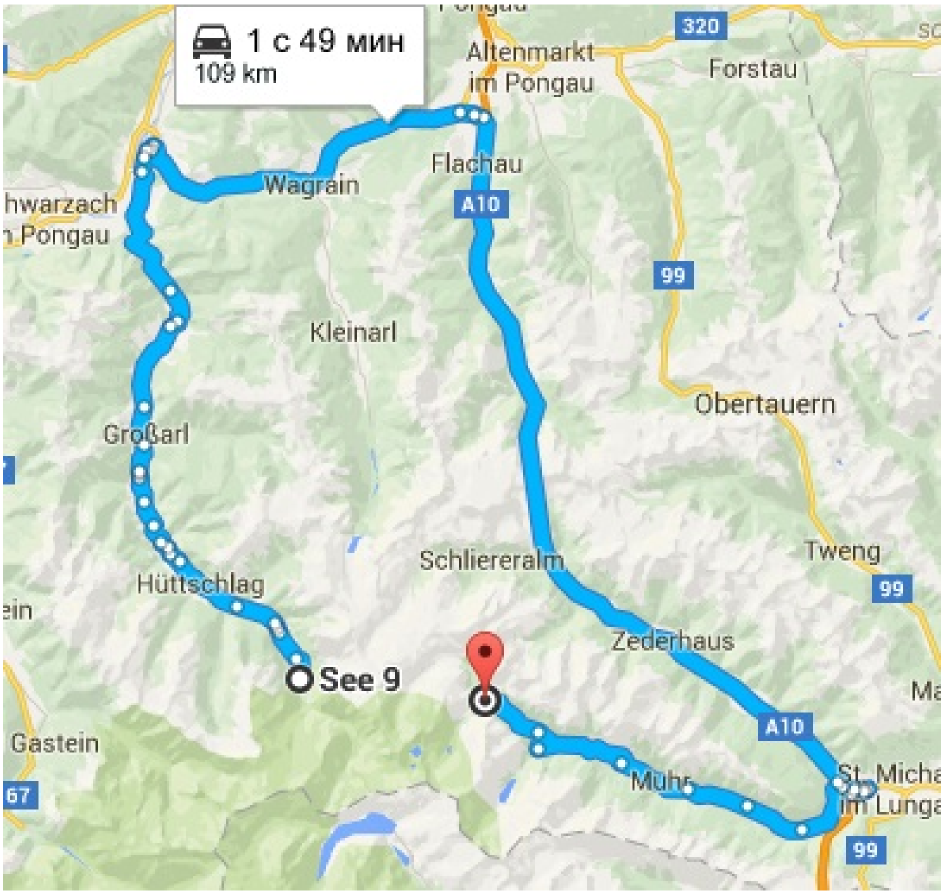


Figure 8: Example of inefficiency of Euclidean approach

where we have two accumulation centers which have air distance about 7 kilometers but nearest road distance between them is 109 kilometers which is huge difference. This is of course one of the pathological examples but at histogram9 we have better view about inefficiency

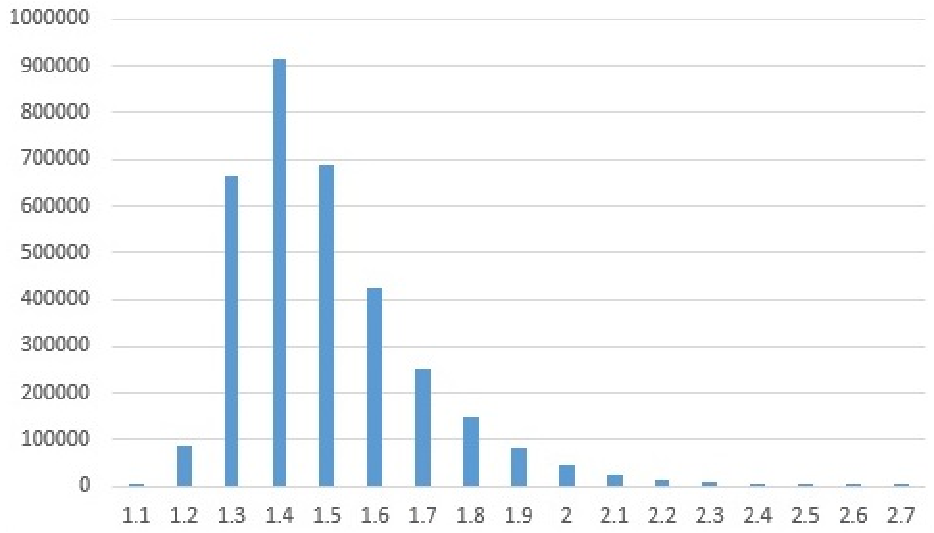


Figure 9: Histogram of ratio between real distances and Euclidean distances for country of Austria

On this histogram we calculated ratios between all of $\frac{1839*1838}{2}$ of road distances in country of Austria with their proper Euclidean distances and showed how those ratios are distributed. For example number on y-axis which corresponds to number 1.3 on x-axis represent how many roads have ratio between 1.2 and 1.3. From histogram we can see that the most of the ratios are between 1.2 and 1.5 with which is not satisfactory for us. We want better results and we will obtain that using metric embedding.
Second problem with Euclidean approach is that our formula which we are maximizing have square of distances in it's formulation. The main reason because we are using that is because on every iterating step it is easy to compute the best local solution. If we would use original formula for optimization then computing the best local solution is quite harder and more time-consuming procedure. About how to compute point in the plane that minimizes the sum of the transportation costs from this point to given destination points is so called Weber problem and more about that can be found in [7]. Now for returning from formula with squares to original formula we use also help from metric embedding where we will with special mapping made that at the end we have exactly what we need.

6 Embedding

6.1 Experimental results and usability of Bourgian algorithm

In this section we will test usability of Bourgian theorem in practice. We tested it at our road networks in Slovenia and Austria. We obtained next results:

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	>2
1	12%	10%	10%	9%	8%	8%	7%	5%	6%	5%	22%
2	18%	14%	12%	8%	6%	6%	5%	5%	4%	2%	21%
3	14%	9%	6%	6%	8%	7%	5%	2%	3%	3%	38%
4	3%	6%	8%	10%	11%	7%	6%	5%	4%	3%	40%
5	11%	8%	8%	5%	5%	4%	3%	5%	3%	3%	46%
6	3%	3%	4%	7%	4%	5%	4%	2%	4%	3%	63%
7	2%	0%	1%	4%	6%	7%	4%	4%	3%	4%	68%
8	2%	0%	0%	0%	1%	2%	2%	5%	4%	5%	80%
9	2%	0%	0%	1%	2%	3%	5%	6%	6%	4%	74%
10	2%	0%	0%	0%	0%	0%	1%	4%	5%	7%	82%

Table 0: Experimental results for Bourgian algorithm applied on Slovenian road network - distribution of distortions : First column represents values of constant C from algorithm and first row represents values for distribution description. For example value in some row which corresponds to value 1.6 from first row represents percentage of distances which distortion is between 1.5 and 1.6. Size of road network was 1225 distances between 50 places

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	>2
1	12%	10%	10%	9%	8%	8%	7%	5%	6%	5%	22%
2	18%	14%	12%	8%	6%	6%	5%	5%	4%	2%	21%
3	14%	9%	6%	6%	8%	7%	5%	2%	3%	3%	38%
4	3%	6%	8%	10%	11%	7%	6%	5%	4%	3%	40%
5	11%	8%	8%	5%	5%	4%	3%	5%	3%	3%	46%
6	3%	3%	4%	7%	4%	5%	4%	2%	4%	3%	63%
7	2%	0%	1%	4%	6%	7%	4%	4%	3%	4%	68%
8	2%	0%	0%	0%	1%	2%	2%	5%	4%	5%	80%
9	2%	0%	0%	1%	2%	3%	5%	6%	6%	4%	74%
10	2%	0%	0%	0%	0%	0%	1%	4%	5%	7%	82%

Table 1: Experimental results for Bourgian algorithm applied on Austrian road network - distribution of distortions : Description of a table is identical as in Slovenian case. Size of road network was 1690041 distances between 1839 places.

From these results we can see two weird things:

- Algorithm is giving is worse results when we are increasing dimensions. We do not know why this is happening and this will be as part of future work.
- We can see easily that there is no any significant improvement comparing to the positions of those places at earth since that is also one form of embedding. This is because we still have that more than 40% of distortions are greater than 1.5.

For now first problem is not important for us since if we have greater dimension, time complexity of our algorithm is greater. So for now we do not care why in high dimensions we have bad results. Now we want some way how to improve algorithm such that we can guarantee better embedding. This can be done using relaxation on distances where we will contract and expand some distances which have great distortion. One algorithm showed great experimental results: This is algorithm with easy logic. If some distortion is too big or too small, decrease and increase it respectively. And we are doing this relaxation for $\log n$ time where $n = |x|$.

Algorithm 3: Relaxation of embedded set

```
Input: Finite metric space $(X, d)$, Embedding $f$, constant $c$. Note: constant
       which is showing which
Output: Improved embedding f
while $\log |X| - - \geq 0$ do
    for $x \in X$ do
        for $y \in X$ do
            if $\max\{\frac{d(x,y)}{\|f(x)-f(y)\|_2}, \frac{\|f(x)-f(y)\|_2}{d(x,y)}\} \geq c$ then
                Contract or expand distance between $f(x)$ and $f(y)$ by moving
                $f(x)$ and $f(y)$ trough line denoted by $f(x)$ and $f(y)$ such that
                given value from if is equal to $c$
return $f$;
```

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	>2
1	76%	18%	5%	2%	1%	0%	0%	0%	0%	0%	0%
2	76%	18%	5%	2%	0%	0%	0%	0%	0%	0%	0%
3	77%	18%	5%	2%	1%	0%	0%	0%	0%	0%	0%
4	77%	17%	5%	2%	1%	0%	0%	0%	0%	0%	0%
5	77%	17%	5%	2%	0%	0%	0%	0%	0%	0%	0%
6	77%	17%	5%	2%	0%	0%	0%	0%	0%	0%	0%
7	77%	17%	5%	2%	0%	0%	0%	0%	0%	0%	0%
8	77%	17%	5%	2%	1%	0%	0%	0%	0%	0%	0%
9	77%	17%	5%	2%	1%	0%	0%	0%	0%	0%	0%
10	77%	17%	5%	2%	1%	0%	0%	0%	0%	0%	0%

Table 2: Experimental results for Relaxation of Bourgian algorithm with $c = 1.1$ applied on Slovenian road network - distribution of distortions : Description of a table is identical as in case without relaxation.

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	>2
1	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
2	70%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
3	70%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
4	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
5	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
6	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
7	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
8	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
9	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%
10	69%	20%	6%	2%	1%	1%	0%	0%	0%	0%	0%

Table 3: Experimental results for relaxation of Bourgian algorithm with $c = 1.1$ applied on Austrian road network - distribution of distortions : Description of a table is identical as in case without relaxation.

In tables 2 and 3 we finally have results which are significant improvement comparing to the positions on map. We have that in both cases we have that over 95% of distortions is less than 1.3. which can be considered as improvement.

6.2 Finalizing results of embeddings

Now we have standard questions: Are our results of embeddings good? Can we do it better? We know that lot of road distances in road network are calculated as part of shortest path distances in already given network. That means that in some potential perfect embedding we are used to have lot of co-linear points which can give bad results in described embedding methods. Can we improve "quality" of given finite metric space such that we can have almost perfect embedding? We have good news! We do not need to care on the original metric spaces given. Let we return to our counter example of prefect embedding given in this section. Suppose that we have square root of distances despite real in initial metric space. Then we have perfect embedding. Just $f(U_4) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, $f(U_1) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $f(U_2) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $f(U_3) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$. With easy checking we can see that this is perfect embedding. So we improved our metric space by square-rooting initial distance. We can for sure apply that on our initial set of distances, but what influence on our result it can have. Now we need to remember some things from the beginning of this paper. We have that in our main objective function distances between centers appear without any power. We constructed optimization algorithm in which objective formula we have square of distances between points. Now if we can do perfect embedding on square-rooted metric space, and then apply our optimization algorithm with squared distances, at the end we will have exactly formula which we need. First we will again provide experimental results of embeddings of square-rooted metric space. As we can see at figures 4 and 5 we got that over 99% of distortions are less than 10% which in some way can be considered as almost perfect embedding. Improvement in embedding can be explained by our counter example: by square-rooting of distances we improved quality of metric space since a large number of sets of points do not need to be co-linear in potential embedding.

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	>2
1	82%	19%	0%	0%	0%	0%	0%	0%	0%	0%	0%
2	90%	12%	0%	0%	0%	0%	0%	0%	0%	0%	0%
3	93%	9%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	92%	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%
5	92%	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%
6	92%	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%
7	92%	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%
8	92%	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%
9	91%	11%	0%	0%	0%	0%	0%	0%	0%	0%	0%
10	91%	11%	0%	0%	0%	0%	0%	0%	0%	0%	0%

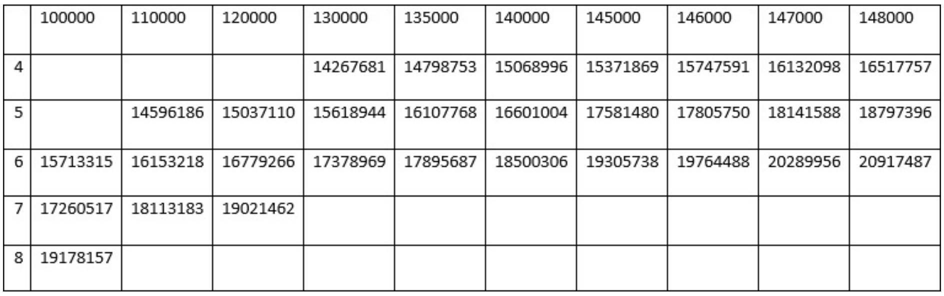
Table 4: Experimental results for Relaxation of Bourgian algorithm with $c = 1.1$ applied on Slovenian square-rooted road network - distribution of distortions : Description of a table is identical as in case without relaxation.

	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	>2
1	93%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%
2	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
3	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
5	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
6	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
7	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
8	94%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%
9	94%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%
10	94%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 5: Experimental results for relaxation of Bourgian algorithm with $c = 1.05$ applied on Austrian square-rooted road network - distribution of distortions : Description of a table is identical as in case without relaxation.

7 Results

Here we will present results of application of our optimization algorithm into obtained high dimensional Euclidean space.



	100000	110000	120000	130000	135000	140000	145000	146000	147000	148000
4				14267681	14798753	15068996	15371869	15747591	16132098	16517757
5		14596186	15037110	15618944	16107768	16601004	17581480	17805750	18141588	18797396
6	15713315	16153218	16779266	17378969	17895687	18500306	19305738	19764488	20289956	20917487
7	17260517	18113183	19021462							
8	19178157									

Figure 10: Experimental results for optimization algorithm in embedded space of size $\log n^2$

At table 10 we can see improvement in every field comparing to the table 6. Improvement is not too much significant since Euclidean case is giving good results.

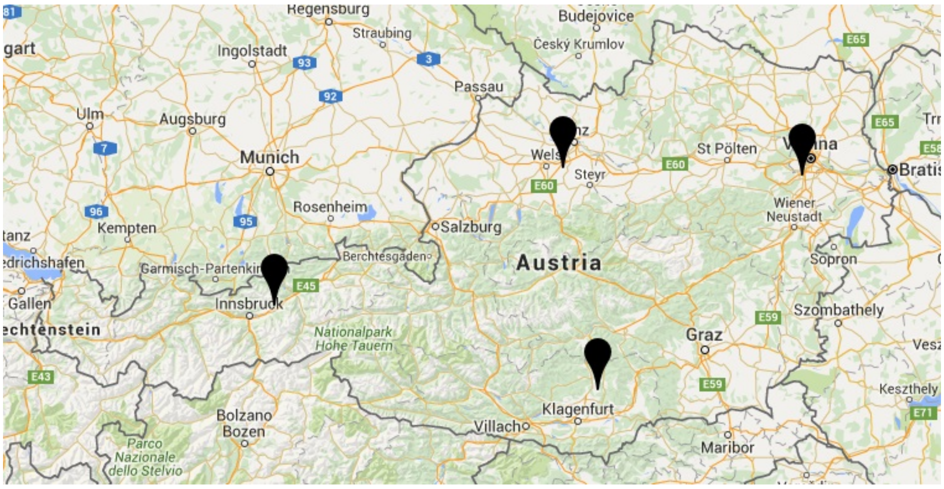


Figure 11: Example of optimal solution obtained using embedding

On figure 11 we can se that difference between maps is really small since Euclidean results is still satisfactory because of small difference between results.

8 Conclusions and future work

We have constructed algorithmic system for solving one complex optimization problem. The truth is that we did not provide enough results on some basic results in algorithm analysis. We did not provide results on time complexity, experimental testing of quality of solution and similar things. Because of that we can divide our future work into three parts:

- **Finding some bound for time complexity.** Analyzing time complexity of iteration is not too easy thing especially if we add to that one running of simplex algorithm in every iteration step. Algorithm in practice is fast giving results in reasonable time, but for sureness we need to find theoretical bounds for it.

- **Comparing to other works and software from similar theme.** There is a lot of work on this theme until today and lot of software are made. So we will have a task to find those which are available and compare their efficiency with our algorithm.

- **Analyzing of adaptability of our decomposition.**Proving that our decomposition of Facility Location Problem into problems of clustering and metric embedding can have sub-optimal algorithm very close to optimal solution (we will try with $1 + \epsilon$ approximation). With this we will prove adaptability of described decomposition.That means that it can be used in many ways depending if you want better optimization or better time complexity which is giving some advantage comparing to MILP.

9 Povzetek naloge v slovenskem jeziku

V zaklučni nalogi smo predstavljali eno novo formo za reševanje znanega problema postavitve objektov. Motivacija za reševanje problema se nahaja v težavi računanja optimalne rešitve za veliko število podatkov. Globlja motivacija je projekt Evropske Unije za reševanje problema akumulacije starega lesa. Poenostavljeno imamo lokacije nabiranja starega lesa kateri se do zdaj ni uporabljal, in zdaj želimo narediti tvornice za obdelavo lesa ampak to želimo narediti da minimiziramo skupne stroške produkcije. Definicija takšnega problema je sktrita točno v definiciji problema postavitve objektov. Do zdaj znane rešitve večinoma uporabljajo mešani celoštevilski linearen porgram kot osnovo za reševanje problema. Težava z takšnimi algoritmi je v tem da so zelo neučinkoviti za velike množice podatkov. Zarad tega tukaj mi pokazujemo eden nov pristop za reševanje kateri se sestoji iz dve znani metodi katere prilagajamo potrebam našega problema. Prva metoda je takoimenovano gručanje podatkov na dano število gruč. Gruče se formirajo tako da minimiziramo varianco med podatki. Tisto gručanje uporabljamo najprej da rešimo evkildsko formo problema. To je forma kjer potne razdalje zamenjamo z geografskim. Tukaj uporabljamo najbolj znan iterativen postopek reševanja za problem gručanja kjer v vsakem koraku vzamemo lokalno najboljšo rešitev. Tisti postopek ne zagotavlja da iteracija bo konvergirala v globalen optimum ampak z metodami pametne začetne izbire centrov se da omejiti pričakovana vrednost rezultata algoritma. Zarad potreb problema dodajamo v postopak še dodatnih podatkov kot so kapacitete centrov katere obslužujemo in centrov s katerimi jih bomo obsuževali. Zdaj ker je problem več kompliciran uporabljamo linearen program za iskanje lokalno optimalne rešitve. Zdaj za še boljše rezultate našega problema se ne moremo zadovoljiti samo z evklidskom analizom. Za reševanje te razilike uporabljamo ugrajanje težinskega grafa, kateri je sam metričen prostor, v več dimenzijski evklidski prostor. Seveda to se ne more perfektno narediti v evklidskim prostorima majhnih dimenzij (manj kot logaritem od števila podatkov), ampak obstajajo načini kje se to lahko naredi z zelo veliiko natančnostjo. Metode za vgrajanje uporabljajo najprej randomizirano izbirno enega števila množic in potem koordinate dobimo tako što za dano točko izračunamo razdalje od vsih izbranih množic. Na takšen način ne dobimo zelo dober rezultat ampak zarad poteb uporabljamo relaksacijo razdalj kjer za dano razdaljo premikamo njene točke da dobimo razdaljo blizu orignalne razdalje iz grafa. Z takšnim postopkom relaksacije dobimo zelo dobre rezultate kjer več od 99% razdalj ima manjšo relativno napako od 10% kar je zelo dober rezultat za naše potrebe. Zdaj ko smo dobili takšen prostor delamo gručanje podatkov v takšnem prostoru kar se lahko naredi ker sam postopek gručanja ni omejen na dimenzijo, ampak želimo da imamo čim manjšo dimenzijo ker dimenzija direktno utika na časovno zahtevnost izvajanja algoritma. Takšen pristop reševanju se izkazal kot zelo hiter na mestih kjer že obstoječi algoritmi niso vspeli izračunati v realnem času, ampak kljub temu da imamo teoretične garancije za rezultat, algoritem ni še dovolj eksperimentalno izpitan in kot bodoče delo bo usmerjeno na temeljiti analizi rezultatov kjer bomo naše rezultate vsporedili z rezultatam z že obstoječimi reševalnikimi za majhne vhodne podatke katere takšni reševalniki lahko rešujejo optimalno. Tudi opisano reševanje obe opisane metode ni edino in tudi bomo posusili pokazati da z drugimi metodami za reševanje opisanih delov algoritma dobimo lahko rešitev poljubno blizu optimalni rešitvi ampak to seveda ne uporabljamo ker je čas izvajanja superpolinomski.

10 Bibliography

[1] E. F. Lambin and P. Meyfroidt, Global land use change, economic globalization, and the looming land scarcity. *Prentice-Hall, 1981* 108 (2011) 3465–3472. *(Cited on page 2.)*

[2] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, The effectiveness of Lloyd-type methods for the k-means problem. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 2000, 25–42. *(Cited on page 6.)*

[3] M. M. Fréchet, Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22 (1906) 1–72. *(Cited on page 10.)*

[4] S. Dasgupta, and A. Gupta, An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms* 22 (2003) 60–65. *(Cited on page 14.)*

[5] N. Linial, E. London, and Y. Rabinovich, The geometry of graphs and some of its algorithmic applications. *Combinatorica* 15 (1995) 215–245. *(Cited on page 14.)*

[6] D. Arthur and S. Vassilvitskii, k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, 1027–1035. *(Cited on page 8.)*

[7] Z. Drezner, K. Klamroth, A. Schöbel, and G. O. Wesolowsky, 1 The Weber Problem. (2000) . *(Cited on page 27.)*

[8] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, The planar k-means problem is NP-hard. In *International Workshop on Algorithms and Computation*, 2009, 274–285. *(Cited on page 6.)*

[9] Y. Rabinovich, On average distortion of embedding metrics into the line and into L 1. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003, 456–462. *(Cited on page 15.)*

[10] J. Vygen, Approximation algorithms facility location problems. In *Forschungsinstitut für Diskrete Mathematik, Rheinische Friedrich-Wilhelms-Universität*, 2005, 0–59. *(Cited on page 2.)*