



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Voroniovské dláždění v rovině s Manhattanskou metrikou

Voronoi tiling in the plane with Manhattan metric

Bakalářská práce

Autor: **Jan Prokop**
Vedoucí práce: **Ing. Petr Ambrož, Ph.D.**
Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Jan Prokop
Studijní program:	Aplikace přírodních věd
Studijní obor:	Matematické inženýrství
Studijní zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Voronoiovské dláždění v rovině s manhattanskou metrikou
Název práce (anglicky):	Voronoi tiling in the plane with Manhattan metric

Pokyny pro vypracování:

- 1) Nastudujte definici a vlastnosti delonovských množin a voronoiovských buněk.
- 2) Proveďte rešerši známých algoritmů pro konstrukci voronoiovského dláždění.
- 3) Vytvořte program, který pro seznam bodů v rovině (s eukleidovskou metrikou) nalezne jeho voronoiovské dláždění.
- 4) Diskutujte potřebné změny algoritmu hledajícího voronoiovské dláždění v případě, že místo eukleidovské metriky budeme používat metriku manhattanskou.
- 5) Porovnejte dláždění získaná pro obě metriky pro různé delonovské množiny, např. pro spektra kubických komplexních pisotovských jednotek s vlastností (F).

Doporučená literatura:

- 1) F. Aurenhammer, Voronoi diagrams — A survey of a fundamental geometric data structure. ACM Comput. Surv. 23 (3), 1991, 345–405. (DOI: 10.1145/116873.116880)
- 2) S. Fortune, Voronoi diagrams and Delaunay triangulations. In 'Computing in Euclidean Geometry' D.Z. Du, F. Hwang (eds.), World Scientific, 1992, 193–233. (DOI: 10.1142/9789814355858_0006)
- 3) A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, Spatial tessellations: Concepts and applications of Voronoi diagrams. John Wiley and Sons, 2009.
- 4) T. Hejda, E. Pelantová, Spectral properties of cubic complex Pisot units. Math. Comput. 85 (297), 2016, 401–421. (DOI: 10.1090/mcom/2983)

Jméno a pracoviště vedoucího bakalářské práce:

Ing. Petr Ambrož, Ph.D.

KM FJFI ČVUT v Praze, Trojanova 13, 120 00 Praha 2

Jméno a pracoviště konzultanta:

Datum zadání bakalářské práce: 31.10.2020

Datum odevzdání bakalářské práce: 7.7.2021

Doba platnosti zadání je dva roky od data zadání.

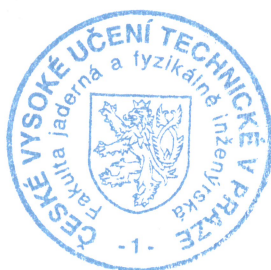
V Praze dne 30.10.2020

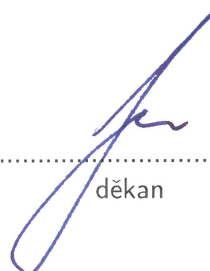


.....
garant oboru



.....
vedoucí katedry





.....
děkan

Poděkování:

Čtěl bych zde poděkovat především svému školiteli Ing. Petru Ambrožovi, Ph.D. za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 3. ledna 2022

Jan Prokop

Název práce:

Voroniovské dlážďení v rovině s manhattanskou metrikou

Autor: Jan Prokop

Obor: Matematické inženýrství metody

Zaměření: Aplikované matematicko-stochastické

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Petr Ambrož, Ph.D., ČVUT, Fakulta jaderná a fyzikálně inženýrská, Katedra matematiky

Abstrakt: Tato bakalářská práce se zabývá tematikou Voroného diagramů v prostorech s euklidovskou a manhattanskou metrikou. Práce obsahuje podrobnou rešerši ke konstrukci Voroného diagramu, využití diagramů a jejich zobecnění. Dále jsou v zde obsaženy nejpoužívanější algoritmy ke konstrukci Voroného diagramů s podrobným postupem a odvozením složitosti. Poté se zabýváme implementací vlastního algoritmu, které konstruuje Voroného diagramy v dvourozměrném prostoru s euklidovskou a manhattanskou metrikou, spolu s úvodem do Delonovských množin, pisotových čísel a aplikací algoritmu na dané Deloného množiny.

Klíčová slova: Algebraická čísla, Delonovské množiny, Divide and conquer, Fortunův algoritmus, Inkrementální konstrukce, Pisotova čísla, Voroného diagram, Zobecněný Voroného diagram.

Title:

Voronoi tiling in the plane with Manhattan metric

Author: Jan Prokop

Abstract: This bachelor thesis considers Voronoi diagrams in space with euclidan and manhattan metric. This work contains detailed overview of construction of Voronoi diagrams, its applications of and some generalized versions. The work also includes most popular algorithms to compute Voronoi diagrams with derivation of their complexity. Next thesis comprise own implementation of algorithms, witch can compute Voronoi diagrams using euclidan and manhattan metrics, along with introduction to Delone sets, Pisot numbers and testing algorithms to given Delone sets.

Key words: Algebraic numbers, Delone sets, Divide and conquer, Fortunes algorithm, Generalized Voronoi diagram, Half-plane intersection, Incremental construction, Pisots numbers, Voronoi diagram.

Obsah

Úvod	8
1 Uvedení do problému	9
1.1 Stručná historie	9
1.1.1 Co je Voroného diagram	9
1.2 Aplikace a výskyt	9
1.2.1 Motivační příklad	10
1.2.2 Řešení geometrických problémů	12
1.2.3 Molekulární chemie	13
1.2.4 Aplikace Voroného Power diagramů	13
1.2.5 Geografie a Meteorologie	15
1.2.6 Biologie	15
1.2.7 Krystalografie	15
2 Voroného diagram	16
2.1 Základní definice a vlastnosti	16
2.2 Zobecnění Voroného diagramu	23
2.2.1 Voroného diagram v jiných metrikách	23
2.2.2 Voroného diagram kružnic	24
2.2.3 Voroného diagram úseček	25
2.2.4 Laguerreho Voroného diagram / Power Voroného diagram	26
2.2.5 Voroného diagram vyšších řádů	27
2.2.6 Centralizovaný/Těžišťový Voroného diagram	28
2.2.7 Další zobecnění Voroného diagramu	28
3 Základní algoritmy	30
3.1 Složitosti algoritmů a dolní odhad složitosti Voroného diagramu	30
3.1.1 Složitosti algoritmů	30
3.1.2 Dolní odhad složitosti algoritmu konstrukce Voroného diagramu	32
3.2 Průnik polorovin	33
3.3 Inkrementální konstrukce	34
3.4 Rozděl a panuj	35
3.5 Fortunův algoritmus	41

4 Implementace algoritmu	44
4.1 Princip algoritmu	44
4.2 Ukázka algoritmu	49
5 Delonovské množiny, teorie čísel a algebra	51
5.1 Úvod do teorie čísel a algebry	51
5.1.1 Algebraická čísla	51
5.1.2 Pisotova čísla	51
5.1.3 Řešení rovnice $X^3 + X^2 + X - 1 = 0$	51
5.2 Úvod do Delonovských množin	52
5.2.1 Základní pojmy a definice	52
5.2.2 Množina $X^m(\beta)$	52
5.2.3 Konstrukce naší množiny	54
6 Voroného diagram pro naši množinu	55
6.1 Voroného diagram naší množiny v euklidovské metrice	57
6.2 Voroného diagram naší množiny v manhattanské metrice	58
Závěr	60

Úvod

Tato bakalářská práce se věnuje teorii potřebné ke studiu Voroného diagramů spolu se základními algoritmy, které jsou vhodné ke konstrukci Voroného diagramů. Dále se práce kromě využití a některých zobecnění Voroného diagramů zabývá vlastní implementací jednoho algoritmu pro dvě metriky a jeho otestování na konkrétních Delonovských množinách.

Kapitoly v bakalářské práci se věnují postupně stručné historii Voroného diagramů a elementárnímu popisu toho co je Voroného diagram, motivačnímu příkladu a aplikacím Voroného diagramu napříč vědními obory.

Dále následuje podrobná diskuze konstrukce a vlastností Voroného diagramu. A také zmínění několika zobecnění Voroného diagramů.

Třetí kapitola je věnována základním algoritmům vhodným pro hledání diagramů spolu s jejich analýzou a odvozením složitosti spolu s teorií potřebnou k této analýze.

Ve čtvrté kapitole je popis vlastní implementace algoritmu pro dvě metriky spolu s diskusí jak se algoritmus mění pro jiné metriky a s ukázkou programu.

Předposlední kapitola je věnována teorii čísel a úvodu do Delonovských množin a Pisotových čísel a množinám tvořeným komplexními kubickými Pisotovými čísly.

V poslední kapitole se práce věnuje aplikaci vlastní implementace na množiny tvořené Pisotovými čísly.

Kapitola 1

Uvedení do problému

1.1 Stručná historie

První zmínka o Voroného diagramu viz obr. 1.1a nebo také Voroného teselaci sahá až do šestnáctého století, kdy Johannes Kepler používal Voroného diagramy při jeho studiích sněžných vloček. Také jej v sedmnáctém století René Descartes využil ve své práci *Principy filosofie* k popisu rozložení hmoty ve sluneční soustavě a vesmíru. Vícerozměrný případ využil Johan Peter Gustav Lejeune Dirichlet při svém studiu kvadratických forem v polovině devatenáctého století. V roce 1854 využil Voroného dláždění britský fyzik John Snow k popisu šíření cholery v jedné londýnské čtvrti. Voroného diagram nese jméno po Georgiji Feodosjeviči Voroném [10], který se na počátku dvacátého století věnoval studiu n -rozměrné formy dláždění.

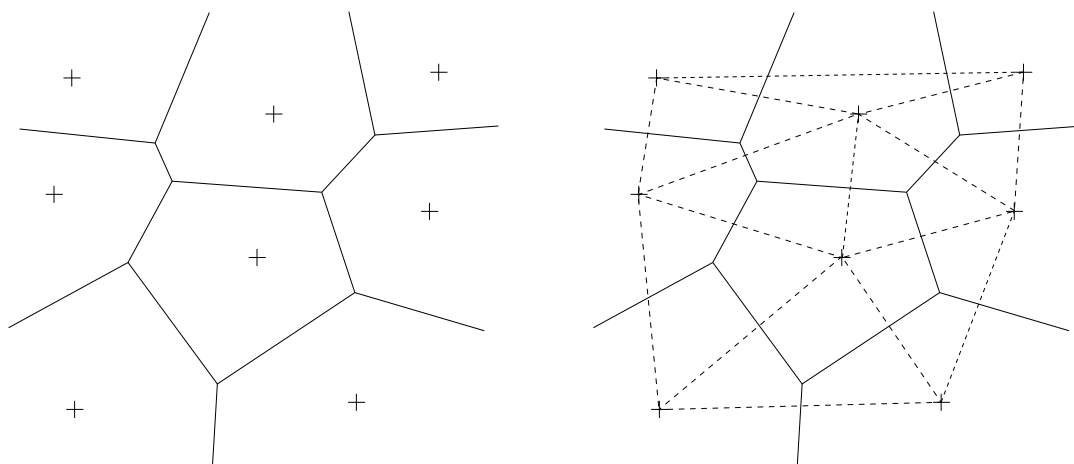
1.1.1 Co je Voroného diagram

Voroného diagram (definice 2.1.13) je definován množinou bodů v metrickém prostoru. Tuto množinu nazýváme *generující množinou*. Ke každému bodu z generující množiny, náleží jeden *Voroného region* (definice 2.1.9). Voroného region je množina všech bodů z metrického prostoru, které jsou blíže ke generujícímu bodu, než k ostatním generujícím bodům. Voroného diagram je sjednocení všech Voroného regionů, které přísluší jednotlivým generujícím bodům. Celý Voroného diagram je složen tedy z generujících bodů a jim příslušným Voroného regionům, které jsou složeny z hranic nebo-li *Voroného hran* (definice 2.1.13) a Voroného vrcholů (definice 2.1.13) to jsou body ve kterých začínají/končí Voroného hrany.

K Voroného diagramu existuje i duální struktura *Delaunayho triangulace* viz obr. 1.1b, kterou získáme tak, že úsečkou spojíme všechny dvojice bodů ze zadané generující množiny, právě tehdy když jejich Voroného regiony mají společnou hranu.

1.2 Aplikace a výskyt

Voroného diagram a jeho zobecnění (sekce 2.2) se využívá v mnoha vědních oborech, převážně ve výpočetní geometrii [20], matematice, optimalizačních úlohách, informatice, robotice, molekulární chemii, geografii, krystalografii, biologii a nebo v územním plánování (sekce 1.2.1), např. pro nalezení nejlepší lokace pro cukrárnu, nemocnici, školu, fastfood nebo úřad, tak aby byla daná budova nejlépe umístěna vzhledem k zákazníkům.



(a) Voroného dlážďení v rovině. Křížky vyznačují body z generující množiny, plné křivky hranice Voroného diagramu nebo-li Voroného hrany.

(b) Delaunayho triangulace a Voroného diagram v rovině. Křížky vyznačují body z generující množiny, plné křivky hranice Voroného diagramu nebo-li Voroného hrany a čárkovými křivkami je znázorněna Delaunayho triangulace.

Obrázek 1.1: Znárodnění Voroného dlážďení a Delaunayho triangulace v rovině.

1.2.1 Motivační příklad

Uvažujme modelový, ne zas až tak nereálný příklad, kdy vlastníme několik poboček cukrárny, ve kterých pečeme domácí jemné pečivo z kvalitních surovin a víme, že lepší cukrárny v této části Prahy (viz obr. 1.2) nejsou. Chceme aby si každý Pražan v této části Prahy mohl během chvilky z domu „skočit“ pro své oblíbené pečivo a pochutiny do jedné z našich poboček a aby každý z obyvatel této části Prahy věděl, kterou pobočku má nejbliže.

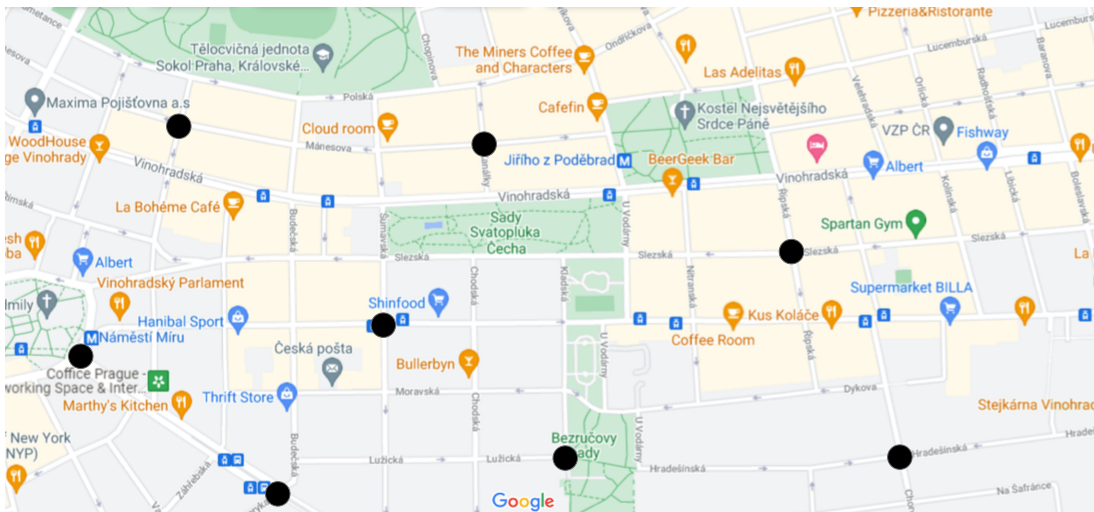
Úkol zní, vymyslete a nakreslete mapu oblastí tak, aby každý potenciální zákazník nacházející se v dané části Prahy věděl, která pobočka naší cukrárny je nejbliže.

Jako řešení by se nabízelo použít nějaké dlážďení prostoru, tak aby každý bod z prostoru (obyvatel), přesně věděl která pobočka je k němu nejbliže. Což je shodou okolností definice Voroného diagramu.

Zkusme tedy tento prostor vydláždít pomocí Voroného diagramu se vzdáleností definovanou tak, jak ji známe tedy euklidovskou metrikou (definice 2.1.3). Po nalezení dlážďení dostaneme mapu jako na obrázku 1.3.

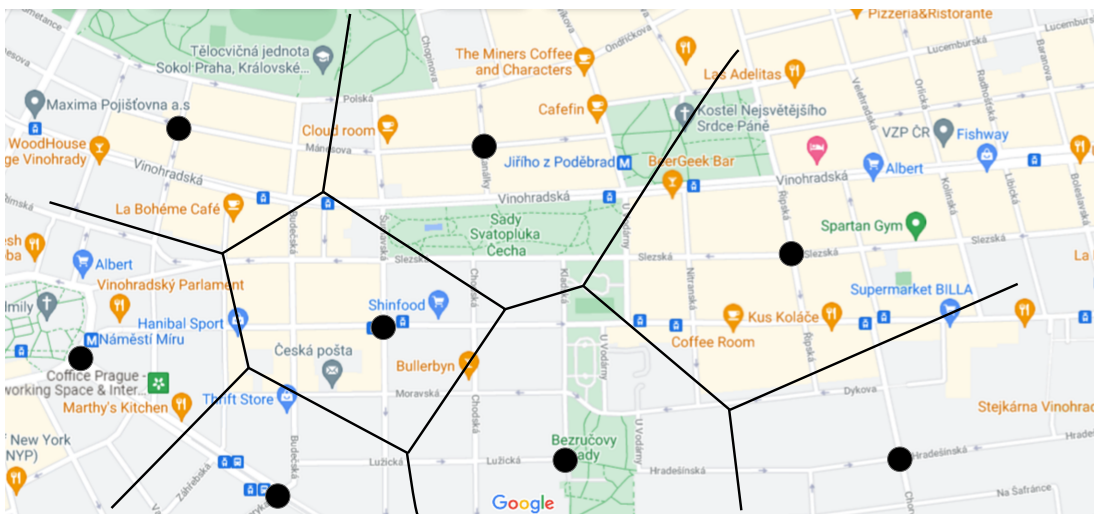
Toto dlážďení nám určitě dobře poslouží a přiblíží představu o tom, jaká pobočka je pro daného obyvatele Prahy nejbliže. Problém je to, že použitá metrika počítá jako nejkratší vzdálenost vzdušnou čarou a přitom ignoruje všechny překážky, které by potenciálnímu nakupujícímu mohly překážet (například celé budovy a bloky budov). Kdybychom měli obdobný příklad zasazen na rovině, kde nejsou žádné překážky, pak bychom dostali optimální řešení, ovšem v blokovém městě jako je tato část Prahy musíme překážky započítat do našeho modelu.

Proto zvolíme jinou metricku, zkusme metricku manhattanskou (definice 2.1.4), která započítává takovéto překážky, pokud je město staveno do bloků jako je na obrázku 1.2. Tato její vlastnost je důsledek toho jak je manhattanská metrika definovaná. Sestrojíme-li Voroného di-



Obrázek 1.2: Mapa dané části Prahy, kde černé tečky znázorňují pobočky naší cukrárny.

Diagram s manhattanskou metrikou nad generující množinou poboček cukráren dostaneme mapu s dlážděním, která je znázorněna na obrázku 1.4. Takto sestrojený Voroného diagram počítá s překážkami v podobě bloků a proto je jako naše mapa pro potenciální zákazníky vhodnější.



Obrázek 1.3: Voroného dláždění části mapy s euklidovskou metrikou pro pobočky cukrárny.

Stačilo by tedy na naše sociální síť a stránky vyvěsit mapku s tímto dlážděním jako na obrázku 1.4, kde by si každý obyvatel našel svoje bydliště a okamžitě by věděl, která pobočka je nejbližší s tím, že ti co by bydleli přímo na hranicích jednotlivých dlaždiček by měli na výběr ze dvou poboček a ti co by bydleli přímo v průsečících jednotlivých hranic by měli na výběr dokonce ze tří poboček.

předem zadanou množinou bodů tak, abychom prošli přes všechny body. α -tvary – extrahování tvaru z předem dané množiny (vizualizace dat a rozpoznávání vzorců). *Nejkratší cesty* – ověřování jestli nejkratší vzdálenost dvou bodů na rovinném planárním grafu nepřekročí daný násobek euklidovské vzdálenosti (robotika).

Clustering pro danou množinu dat znamená nalezení podmnožin (*Clusters*), jejichž prvky jsou si podobné a prvky napříč clusteru si jsou nepodobné/rozdílné vzhledem nějaké podmínce podobnosti. Clustering dat je hojně využíván v mnoha vědních odvětvích a Voroného diagramů se při Clusteringu využívá, když je podobnostní podmínka definována nějakou formou vzdálenosti. Jde tedy například o hledání shluků v datech nebo vzorců a tvarů v jejich rozdělení. Patří sem také *Dílčí clustering* a *Hierarchický clustering*. Další aplikací je třeba komprese obrázků, hledání optimálních pravidel pro kvadraturu, optimální reprezentace kvantizace a shlukování. Více v [39, 33].

Zobecněné Voroného diagramy, především úseček (sekce 2.2.3), kružnic (sekce 2.2.2) a polygonů [7], nachází uplatnění v *Plánování pohybu*, které je nejvíce využíváno v robotice a v přepravních problémech. Předpokládejme Robota, pro jednoduchost bezrozměrného (tedy bodového) a chceme po něm, aby se dostal z místa A do místa B . Problém je, že má v daném plánu/rovině množinu o n překážkách, které jsou tvořené např. nepřekonatelnými úsečkami. Jak takového robota navigovat, aby se dostal z místa A do místa B a přitom nenaboural do žádné překážky (úsečky). Robot není schopen vidět (není v něm žádná kamera se snímačem). Pouze chceme robotovi říct po jakých bodech a křivkách se pohybovat aby došel do cíle.

Sestrojíme Voroného diagram úseček (pokud jsou překážky úsečky) nebo Voroného diagram kružnic (pokud jsou překážky kružnice) ev. kombinaci nějakých verzí zobecnění Voroného diagramů (sekce 2.2). Poté co je Voroného diagram sestaven, nalezneme nejkratší možnou cestu z naší pozice A na Voroného hranu a pošleme robota po Voroného hranách, dokud nebude jeho vzdálenost od cílového bodu B nejkratší, pak už stačí pouze z Voroného hrany přejít po nejkratší cestě do cílového bodu B . Do modelu můžeme přidat i více parametrů, například to, že rozměry robota nebudou nulové, tím se nám jistě průchody mezi překážkami uzavřou a je potřeba si dát na takové případy pozor. Znázornění viz obr.1.5. Více v [7, 32].

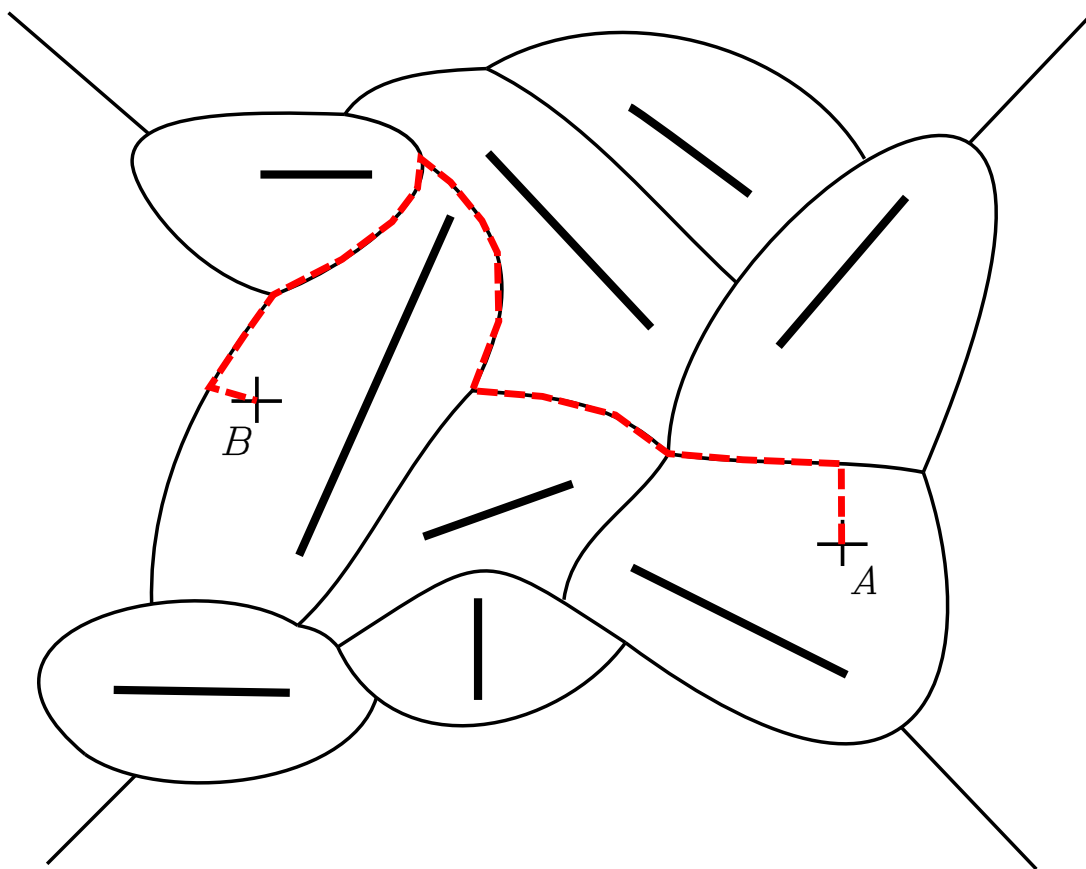
1.2.3 Molekulární chemie

Voroného diagramy a Delaunayho triangulace se také využívají při analýze tunelů v molekulách, především při vývoji léků. Umístění a šířka tunelů mezi jednotlivými proteiny určuje vlastnost dané molekuly (jak moc můžou proteiny *komunikovat* s molekulami). Pokud dokážeme těmto chemickým procesům porozumět, tak jsme pak schopni je i ovlivňovat – například zvětšováním těchto tunelů nebo naopak jejich zmenšováním. K hledání těchto tunelů se využívá Voroného diagramů, více v [28, 11].

Dalším příkladem využití Voroného diagramu respektive jeho zobecněním na generující množinu kružnic je modelování růstu a vývoje částic a molekul, více v [5].

1.2.4 Aplikace Voroného Power diagramů

Voroného Power diagramy (sekce 2.2.4), jsou zobecněním k standardním Voroného diagramům a mají využití v analýze d -rozměrného prostoru, při hledání efektivních řešeních geometrických problémů, více v [6].



Obrázek 1.5: Znázornění cesty (červenou čárkovanou čarou), bez nárazu do překážky, robota z bodu A do bodu B .

Osvětlenost koulí: mějme d -rozměrný prostor a necht' V a B jsou podmnožiny tohoto prostoru. Říkáme, že V osvětluje B , pokud pro každý bod x na hranici B existuje bod y ve V takový, že úsečka spojující body x a y neprotíná/neprochází množinou B . Necht' B je sjednocení n koulí z prostoru, pak body osvětlenosti jsou vrcholy Voroného Power diagramu a úlohu lze redukovat na hledání Voroného Power diagramů. Více v [6].

Detekce k -množin: mějme množinu M s n body v d -rozměrném prostoru a $M_1 \subseteq M$ tak, že $|M_1| = k$, kde $(1 \leq k \leq n - 1)$. M_1 nazýváme k -množinou množiny M , pokud existuje poloprostor h tak, že $h \cap M = M_1$. k -množiny mohou být charakterizovány pomocí Voroného Power diagramů, více v [6].

Poznámka. Zápisem $|M_1|$ máme na mysli mohutnost množiny M_1 resp. počet prvků v množině M_1 .

Dále se pak Voroného Power diagramy používají ke konstrukci: *Multiplicative weighted Voronoi diagrams* a *Voroného diagramy koulí*, [6].

1.2.5 Geografie a Meteorologie

Voroného diagramy se v geografii využívají například při analýze osídlení. Pokud na dané území aplikujeme Voroného dláždění tak, že generující body budou rozmístěny podle hustoty obyvatel, potom krátké hranice Voroného diagramu značí hustě obydlené oblasti a dlouhé hranice řídké obydlené oblasti (aplice clusteringu na shluky dat). Aproximace pomocí Voroného diagramu se používá při předpovědi počasí, pro zjednodušení výpočtů. Pro tyto účely se využívá zobecnění centralizovaného nebo-li těžišťového Voroného diagramu (sekce 2.2.6). Více v [39, 13]. Voroného diagramy mají své využití i v modelování změn a predikci klimatu a při simulaci šíření kinetické energie, kterou v sobě má proudění vzduchu v atmosféře. Obecně v geologii a meteorologii jako pomůcka při simulaci proudění např. vzduchu, tepla, lávy . . . Více v [19, 33].

1.2.6 Biologie

Voroného diagram se například používá při modelování šíření a okupování oblasti daným druhem rostliny. Každá samostatná rostlina se bere jako generující bod a pomocí postupného přidávání bodů jsme schopni namodelovat šíření rostlin. Podobným způsobem můžeme modelovat růst buněk nebo korálů. Jinými slovy se jedná pouze o aplikaci již zmíněného clusteringu na hledání a predikování shluků, více v [33, 39, 28, 11].

1.2.7 Krystalografie

Voroného digram slouží jako dobrá aproximace modelování růstu krystalů. Za předpokladu, že krystaly rostou stejnou rychlostí, můžeme v prostoru zvolit generující body, jako místa kde začnou krystaly růst. Jak se postupně jejich objem zvětšuje, jakmile se dotknou tak jejich růst pokračuje v jiném směru, v tom kde je ještě místo. Pokud bychom uvažovali rozdílnou rychlost růstu krystalů, museli bychom každému generujícímu bodu přiřadit jinou váhu, výsledkem takového dláždění by byla mozaika podobná té, která se tvoří z mýdlových bublin. Více v [39, 5].

Kapitola 2

Voroného diagram

V této kapitole si uvedeme základní pojmy, definice a vlastnosti Voroného diagramu. Budeme uvažovat metrický prostor \mathbb{R}^2 s euklidovskou nebo manhattanskou metrikou a konečnou diskrétní (definice 5.2.1) neprázdnou množinou bodů S . Více podrobností lze nalézt v [7, 17].

2.1 Základní definice a vlastnosti

Začneme definicí obecné metriky pro n -rozměrný *metrický prostor*, který si zdefinujeme s ní a na kterém se budeme v celé práci „pohybovat“.

Definice 2.1.1. Dvojici (\mathcal{M}, ρ) nazveme *metrickým prostorem*, kde \mathcal{M} je libovolná neprázdná množina a ρ je zobrazení: $\mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$, které nazýváme *metrikou*, a které splňuje následující axiomy $\forall x, y, z \in \mathcal{M}$:

- Axiom nezápornosti: $\rho(x, y) \geq 0$,
- Axiom totožnosti: $\rho(x, y) = 0 \iff x = y$,
- Axiom symetrie: $\rho(x, y) = \rho(y, x)$,
- Axiom trojúhelníkové nerovnosti: $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.

Poznámka. Podle definice metrického prostoru tedy dostáváme, že prostor \mathbb{R}^n , $n \in \mathbb{N}$ s euklidovskou nebo manhattanskou metrikou je metrickým prostorem, více v [15].

Definice 2.1.2. Nechť (\mathbb{R}^n, d) je n -dimenzionální metrický prostor, pak definujeme *obecnou metriku* na tomto prostoru jako:

$$d(x, y) := \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}, \quad (2.1)$$

kde x a y jsou libovolné body z \mathcal{M} a x_i je i -tá složka bodu x a y_i je i -tá složka bodu y , a kde p a n jsou přirozená čísla.

Poznámka. Pokud v definici 2.1.2 za p dosadíme dvojku a za n také dvojku dostaneme definici euklidovské vzdálenosti (metriky) 2.1.3 na dvourozměrném metrickém prostoru. Naopak pokud za p dosadíme jedničku a místo n necháme dvojku, dostaneme definici manhattanské vzdálenosti (metriky) na dvourozměrném metrickém prostoru.

Nyní rigorózně zadefinujeme obě vzdálenosti zmíněné v poznámce 2.1.

Definice 2.1.3. Nechť body $x = (x_1, x_2) \in \mathbb{R}^2$ a $y = (y_1, y_2) \in \mathbb{R}^2$ pak definujeme jejich Euklidovskou vzdálenost d_e jako:

$$d_e(x, y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}. \quad (2.2)$$

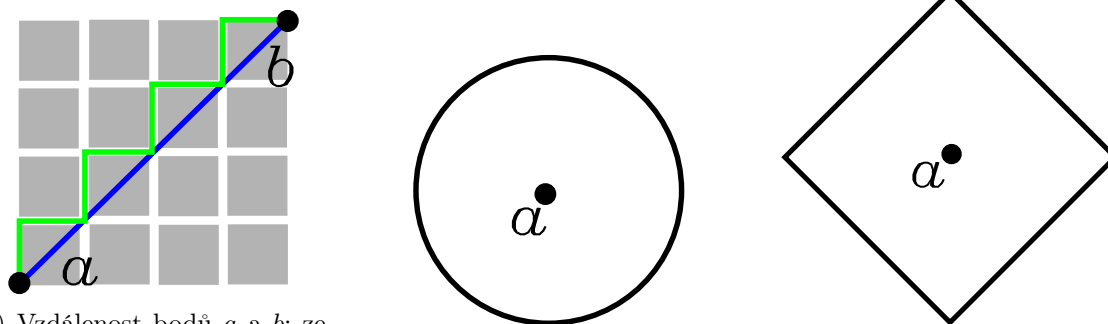
Takto definovaná vzdálenost, splňuje „běžnou“ představu o vzdálenosti v našem světě (pokud by naše Země byla placatá).

Definice 2.1.4. Nechť body $x = (x_1, x_2) \in \mathbb{R}^2$ a $y = (y_1, y_2) \in \mathbb{R}^2$ pak definujeme jejich Manhattanskou vzdálenost d_m jako:

$$d_m(x, y) := |x_1 - y_1| + |x_2 - y_2|. \quad (2.3)$$

Definice 2.1.5. Nechť (\mathbb{R}^2, d) je metrický prostor s libovolnou metrikou $d(\cdot, \cdot)$ z definice 2.1.2. A nechť bod $x \in \mathbb{R}^2$ a $r \in \mathbb{R}^+$. Pak množinu $C(x) := \{y \in \mathbb{R}^2 \mid d(y, x) = r\}$ nazveme *kružnicí* se středem v bodě x a poloměrem r .

Pro představu jaký je rozdíl mezi euklidovskou a manhattanskou metrikou, ve vzdálenosti dvou bodů a ve tvaru kružnice, je přiložen obrázek 2.1.



(a) Vzdálenost bodů a a b : zelená křivka pro Manhattanskou metriku a modrá pro Euklidovskou.

(b) Kružnice se středem a v Euklidovské metrice.

(c) Kružnice se středem a v Manhattanské metrice.

Obrázek 2.1: Porovnání Euklidovské a Manhattanské metriky.

Definice 2.1.6. Mějme body $x, y \in \mathbb{R}^2$ a nechť $d(\cdot, \cdot)$ je metrika na \mathbb{R}^2 , pak definujeme $B(x, y)$ bisektor těchto bodů jako množinu všech bodů v prostoru \mathbb{R}^2 pro, které platí, že jejich vzdálenost od bodu x je stejná jako jejich vzdálenost od bodu y .

$$B(x, y) := \{z \in \mathbb{R}^2 \mid d(x, z) = d(y, z)\}. \quad (2.4)$$

Z Definice 2.1.6 můžeme vyčíst, že bisektor je vlastně osa úsečky, která spojuje body x a y a jedná se tedy o normálu pomyslné přímky, která by procházela body x a y . Toto ovšem neplatí obecně pro všechny metriky, ale pouze pro metriku euklidovskou např. u manhattanské metriky může nastat celkově sedm případů tvarů bisektoru viz obr. 4.1.

Definice 2.1.7. Necht' body $x, y \in \mathbb{R}^2$ a necht' $d(\cdot, \cdot)$ je metrika na \mathbb{R}^2 . Pak definujeme *polorovinu* $D(x, y)$ bodů x a y jako:

$$D(x, y) := \{z \mid d(x, z) \leq d(y, z)\}. \quad (2.5)$$

$D(x, y)$ je tedy množina všech bodů, jejichž vzdálenost od bodu x je menší nebo rovna než jejich vzdálenost od bodu y .

Můžeme si všimnout tedy, že obě poloroviny $D(x, y)$ i $D(y, x)$ pro body $x, y \in S$, obsahují bisektory $B(x, y)$ a $B(y, x)$ tzn. $\forall x, y \in S, B(x, y) \subset D(x, y)$.

Definice 2.1.8. Necht' S je množina bodů v prostoru \mathbb{R}^2 a necht' d je metrika na \mathbb{R}^2 . Pak *Voroného region* $VR(x, S)$, generovaný bodem $x \in S$ definujeme jako:

$$VR(x, S) = \{y \in \mathbb{R}^2 \mid \forall z \in S, d(y, x) \leq d(y, z)\}. \quad (2.6)$$

$VR(x, S)$ tedy obsahuje všechny body z \mathbb{R}^2 , pro které platí, že bod x je k nim nejbližší ze všech ostatních bodů z množiny S . Z předchozí definice lze vyzorovat alternativní definici Voroného regionu: Voroného region generovaný bodem x je průnik všech polorovin bodu x , s ostatními body z množiny S formálně zapsáno:

Definice 2.1.9 (Alternativní definice Voroného regionu). Necht' S je množina bodů v Euklidovském prostoru \mathbb{R}^2 . Pak *Voroného region* $VR(x, S)$, generovaný bodem $x \in S$ definujeme jako:

$$VR(x, S) = \bigcap_{y \in S, y \neq x} D(x, y). \quad (2.7)$$

Všimněme si, že Voroného region vzniká průnikem právě $n - 1$ polorovin. Kde n je počet bodů v množině S . Vzhledem k tomu, že v euklidovském prostoru jsou všechny poloroviny konvexní množiny, tak jejich průnik musí být také konvexní, tedy Voroného region je konvexní množina.

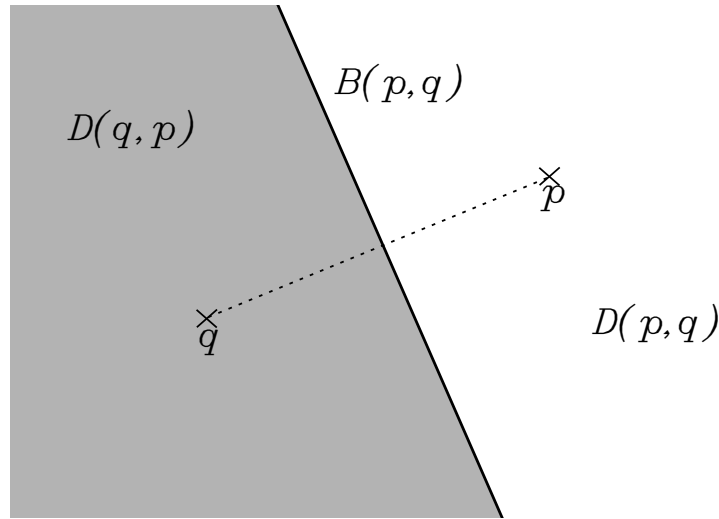
Definice 2.1.10. *Otevřenou kouli*, se středem v bodě x a poloměrem $r \in \mathbb{R}^+, r > 0$ na metrickém prostoru \mathbb{R}^n rozumíme množinu $B_r(x) = \{y \in \mathbb{R}^n \mid d(y, x) < r\}$. Kde $d(\cdot, \cdot)$ je metrika na \mathbb{R}^n .

Definice 2.1.11. *Uzavřenou kouli*, se středem v bodě x a poloměrem $r \in \mathbb{R}^+, r > 0$ na metrickém prostoru \mathbb{R}^n rozumíme množinu $B_r^*(x) = \{y \in \mathbb{R}^n \mid d(y, x) \leq r\}$. Kde $d(\cdot, \cdot)$ je metrika na \mathbb{R}^n .

Definice 2.1.12. *Hranicí koule*, se středem v bodě x a poloměrem $r \in \mathbb{R}^+, r > 0$ na metrickém prostoru \mathbb{R}^n rozumíme množinu $H_{x,r} = \{y \in \mathbb{R}^n \mid d(y, x) = r\}$ tzn. $H_{x,r} = B_r^*(x) - B_r(x)$. Kde $d(\cdot, \cdot)$ je metrika na \mathbb{R}^n .

Poznámka. Z definice 2.1.12 můžeme vidět, že definice hranice koule za předpokladu, že $n = 2$ je ekvivalentní s definicí kružnice (definice 2.1.5) s poloměrem r .

Připomeňme, že množinu nazveme *omezenou* pokud existuje koule s konečným poloměrem tak, že tato koule obklopí celou množinu. V opačném případě je množina *neomezená*. Množinu nazveme *uzavřenou* pokud obsahuje i svoji hranici. Množina bez svojí hranice, je množina *otevřená*, více v [17].



Obrázek 2.2: Uvažujme metrický prostor \mathbb{R}^2 s euklidovskou metrikou. Na obrázku je čárkovane znázorněna spojnice bodů q a p , kde její délka je vzdálenost těchto bodů. Šedou barvou je znázorněna množina všech bodů v prostoru, které jsou blíže k bodu q tedy množina $D(q, p)$ a bílou barvou znázorněna množina všech bodů v prostoru, které jsou blíže k bodu p tedy množina $D(p, q)$. $B(p, q)$ je bisektor k bodům p a q tedy množina všech bodů v prostoru, které jsou od bodů p a q vzdáleny stejně.

Definice 2.1.13. Necht $S = \{x_1, x_2, \dots, x_n\}$ je množina bodů v metrickém prostoru \mathbb{R}^2 a necht $x_i \in S$, $\forall i \in \hat{n}$ a $\text{VR}(x_i, S)$ je Voroného region příslušný k bodu x_i . Pak společnou hranici mezi dvěma Voroného regiony nazveme *Voroného hranou* právě tehdy, když tato hranice obsahuje více jak jeden bod z prostoru \mathbb{R}^2 . Koncové body (pokud existují) Voroného hran nazveme *Voroného vrcholy*. Tyto vrcholy leží na společné hranici tří a více Voroného regionů. *Voroného diagramem* $V(S)$ nazveme sjednocení všech Voroného regionů, které jsou generovány všemi body z množiny S .

$$V(S) = \bigcup_{i \in \hat{n}} \text{VR}(x_i, S). \quad (2.8)$$

Z definic výše můžeme vypožorovat, že pokud bychom si označili Voroného hranu mezi Voroného regiony, které jsou generovány body x a y , jako a . Tak tato hrana je vlastně část bisektoru, který je generován body x a y , navíc pokud by množina S obsahovala pouze tyto dva body pak je tento bisektor a Voroného hrana totéž, tedy Voroného hrana by byla přímka (pokud bychom uvažovali euklidovskou metrikou).

Věta 2.1.1. Necht S je množina bodů v metrickém prostoru \mathbb{R}^2 . Necht $C(x)$ je kružnice se středem v bodě $x \in \mathbb{R}^2$ a necht p, q a o jsou prvky S .

- Pokud je $C(x)$ kružnice s nejmenším poloměrem tak, že $C(x) \cap S = p$, pak bod x leží uvnitř Voroného regionu, který je generován bodem p , tzn. $x \in \text{VR}(p, S)$.
- Pokud je $C(x)$ kružnice s nejmenším poloměrem tak, že $C(x) \cap S = \{p, q\}$, pak bod x leží na Voroného hraně, která přísluší Voroného regionům, které jsou generovány body p a q , tzn. $x \in \{\text{VR}(p, S) \cap \text{VR}(q, S)\}$.

- Pokud je $C(x)$ kružnice s nejmenším poloměrem tak, že $C(x) \cap S = \{p, q, o, \dots\}$, pak bod x leží na Voroného vrcholu, který přísluší Voroného regionům, které jsou generovány body $\{p, q, o, \dots\}$, tzn. $x \in \{VR(p, S) \cap VR(q, S) \cap VR(o, S) \cap \dots\}$.

Důkaz. Pokud se kružnice protne právě s jedním bodem p , pak je bod p prvek z S , který je k bodu x nejbližší. To znamená, že $x \in D(p, r)$, $\forall r \in S \setminus \{p\}$.

Pokud kružnice C protne právě dva body p a q , tak víme, že tyto dva body jsou od bodu x stejně vzdáleny a tím pádem se nachází na bisektoru, který je generován body p a q . A zároveň platí, že $x \in D(p, r) \wedge x \in D(q, r)$, $\forall r \in S \setminus \{p, q\}$. Z definice 2.1.13 víme, že toto platí pouze pro Voroného hranici mezi Voroného diagramy, které jsou generovány body p a q . analogickou úvahou můžeme dokázat i poslední případ. \square

Definice 2.1.14. *Stupeň* Voroného vrcholu, je počet Voroného hran, které do tohoto vrcholu vstupují, nebo-li počet přilehlých Voroného regionů viz obr. 2.3.

Definice 2.1.15. Mějme S konečnou množinu bodů takovou, že $S = \{x_1, x_2, \dots, x_n\}$. Potom *konvexní kombinací* prvků $\{x_1, x_2, \dots, x_n\}$ nazveme lineární kombinací $\sum_{i=1}^n \alpha_i x_i$, která splňuje $\sum_{i=1}^n \alpha_i = 1$ a $\alpha_i \geq 0$ pro každé $i \in \hat{n}$. Množinu všech konvexních kombinací nazýváme *konvexním obalem* prvků $\{x_1, x_2, \dots, x_n\}$ a značíme jej $[x_1, x_2, \dots, x_n]_{\mathcal{K}}$. Platí tedy:

$$[x_1, x_2, \dots, x_n]_{\mathcal{K}} := \left\{ \sum_{i=1}^n \alpha_i x_i \mid \sum_{i=1}^n \alpha_i = 1, \alpha_i \in \mathbb{R} \text{ a } \alpha_i \geq 0 \text{ a pro každé } i \in \hat{n} \right\}.$$

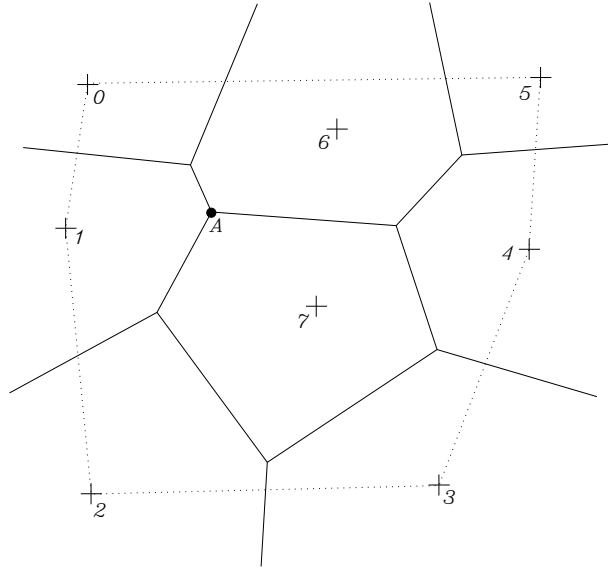
Z definice 2.1.13 vyplývá, že každý Voroného vrchol má stupeň minimálně tři a Voroného diagram je dekompozice prostoru s tím, že jednotlivé Voroného regiony se nepřekrývají až na Voroného hrany a Voroného vrcholy.

Na Voroného diagramu definujeme jeho *Konvexní obal* viz obr. 2.3 jako nejmenší konvexní množinu bodů, která obsahuje celou generující množinu S . Konvexní obal množiny S je *konvexní polygon* a je ohraničen $h \leq n$ tzv. *extrémními body* viz obr. 2.3 pro které platí $h \in S$.

Věta 2.1.2. *Nechť S je množina bodů z \mathbb{R}^2 , která generuje Voroného diagram $V(S)$ a mějme bod $p \in S$. Pak bod p leží na hranici konvexního obalu množiny S právě tehdy, když je jeho Voroného region neohrazen, tzn. jedná se o neomezenou množinu.*

Důkaz. Uvažujme BÚNO případ, který je na obrázku 2.4. Voroného region generovaný bodem x je neomezený právě tehdy, když existuje nějaký bod $y \in S$, takový, že Voroného diagram obsahuje polopřímku ležící na $B(x, y)$ jako Voroného hranu. Nechť bod $p \in B(x, y)$ a zároveň nechť $C(p)$ je kružnice se středem v bodě p , jejíž poloměr se zvětšuje dokud neprojde skrz body x a y viz obr. 2.4. Bod $p \in V(S)$ právě tehdy, když kružnice $C(p)$ neobsahuje žádný jiný bod z S . Když pohybuje bodem p směrem doprava (od spojnice bodů x a y), podél bisektoru $B(x, y)$, průnik kružnice $C(p)$ s pravou stranou roste a průnik s levou stranou se naopak zmenšuje. Pokud se v pravé polorovině nachází další bod z množiny S , pak ho musí kružnice časem protnout a tím způsobí, že zde bude Voroného hrana končit. V opačném případě se musí všechny ostatní body z množiny S nacházet v levé polorovině. Pak budou body x a y prvky konvexního obalu množiny S . \square

Řekněme, že je množina S v *konvexní pozici*, pokud všechny její body leží na hranici konvexního obalu. Pak jako důsledek věty 2.1.15 dostaneme, že všechny její Voroného regiony jsou



Obrázek 2.3: Tečkovaně je na obrázku znázorněn konvexní obal pro Voroného diagram, složený z osmi bodů. Voroného vrchol A má stupeň roven třem. Body s indexy $0, 1, 2, 3, 4, 5$, jsou body extrémní.

neomezené množiny.

Nyní provedeme rychlou odbočku do teorie grafů, více v [14].

Definice 2.1.16. Necht V je libovolná neprázdná konečná množina bodů, B je množina všech dvouprvkových podmnožin množiny V a necht $E \subset B$. Potom uspořádanou dvojici (V, E) nazýváme *neorientovaným grafem*, značíme $G(V, E)$.

Definice 2.1.17. *Cesta v grafu $G(V, E)$* je posloupnost vrcholů a hran $(v_0, e_1, v_1, \dots, e_t, v_t)$, kde $\forall i \in \hat{t}$, $e_i = \{v_{i-1}, v_i\} \in E$ a $\forall j \in \hat{t}$, $v_j \in V$.

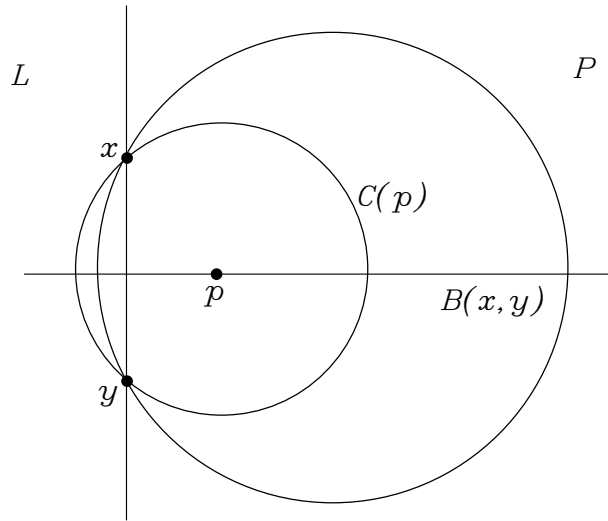
Definice 2.1.18. Graf $G(V, E)$, kde $|V| = n$, nazveme *souvislý*, jestliže $\forall u, v \in V$, existuje cesta v grafu $G(V, E)$ z u do v .

Pro jednoduchost budeme uvažovat tedy *souvislé* grafy, protože Voroného diagram bude vždy graf souvislý.

Věta 2.1.3 (Eulerův vzorec pro grafy). *Necht se žádné dvě hrany grafu $G(V, E)$ v rovině neprotínají a necht je graf souvislý. A označme si $|V|$ jako počet vrcholů v grafu, $|E|$ jako počet hran a $|F|$ jako počet uzavřených oblastí v grafu, vymezených jeho hranami. Pak platí:*

$$|V| - |E| + |F| = 1. \quad (2.9)$$

Důkaz. Ukážeme, že libovolný graf, který splňuje předpoklady věty, lze zredukovat na jeden bod (izolovaný uzel), pro který rovnost zjevně platí. Máme dva způsoby jak graf redukovat:



Obrázek 2.4: Jak se bod p posouvá doprava, tak se průnik kružnice s levou stranou zmenšuje zatímco průnik s pravou stranou se zvětšuje.

1. Pokud v grafu narazíme na tzv. volnou hranu, tedy takovou hranu, která vstupuje do vrcholu do kterého nevstupuje žádná jiná hrana, pak odstraníme tuto hranu spolu s uvolněným vrcholem, který by jinak zůstal sám a tím by nám vznikl nesouvislý graf. Tím se nám zmenší $|V|$ a $|E|$ o 1 a $|F|$ se nezmění. Tedy tento postup neměl ve výsledku na vyslovený vzorec opět žádný vliv.
2. Odebereme hranu na hranici nějaké ohraničené oblasti. Odstraněním hrany, ale žádného vrcholu na obou koncích hrany, se zmenší číslo $|E|$ o 1, číslo $|V|$ zůstane stejné a číslo $|F|$ se zmenší o 1. Ve výsledku se tedy rovnice $|V| - |E| + |F| = 1$ nezmění. Změny hodnot $|E|$ a $|F|$ se vzájemně vyruší viz obr. 2.5.

Takto můžeme postupně celý graf redukovat, dokud nám nezůstane pouze jeden izolovaný uzel, pro který tento předpis platí, což si můžeme ověřit jednoduchým dosazením. \square

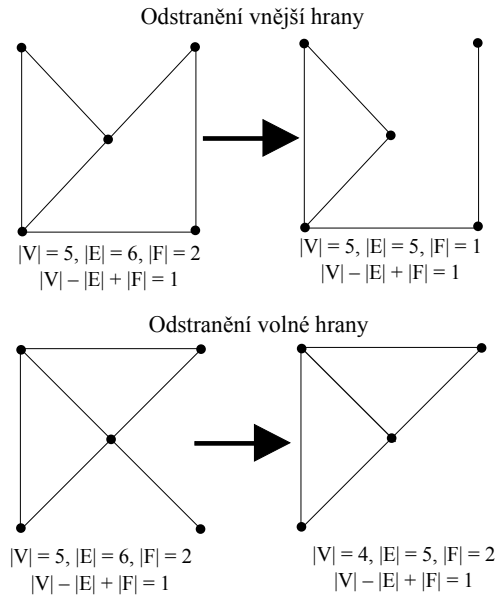
Nyní, vyzbrojeni *Eulerovým vzorcem pro grafy*, se můžeme vrátit k Voroného diagramu a vyslovit větu, která nám udá závislost počtu vrcholů a hran na počtu generujících bodů.

Věta 2.1.4. *Nechť S je množina generující Voroného diagram a obsahující $n \geq 3$ prvků. Pak Voroného diagram $V(S)$ má $\mathcal{O}(n)$ Voroného hran a vrcholů. A platí, že Voroného diagram má nejvýše $2n - 5$ Voroného vrcholů a maximálně $3n - 6$ Voroného hran.*

Důkaz. Použijeme dříve dokázaný Eulerův vzorec pro grafy, nemůžeme jej přímo aplikovat na Voroného diagram, protože ve Voroného diagramu se vyskytují polopřímky, které „běží“ do nekonečna, což se vymyká naší definici grafu. Abychom se takovéto komplikaci vyhnuli, přidáme do Voroného diagramu jeden bod, který bude plnit funkci „nekonečna“ a ze všech polopřímek ve Voroného diagramu uděláme křivky tak, že jejich část která neměla konec „pošleme“ do našeho nově přidaného bodu.

Nyní můžeme použít Eulerův vzorec pro grafy a dostaneme následující rovnost:

$$|V| + 1 - |E| + n - 1 = 1 \tag{2.10}$$



Obrázek 2.5: Znárodnění odstranění volné a vnější hrany z rovinného neorientovaného grafu.

V rovnici je $n - 1$ počet uzavřených oblastí. Víme, že každý bod až na dva (jeden z nich je ten, který zastupuje „nekonečno“ a druhý je jeden bod z S , který tvoří konvexní obal S a k němu příslušný Voroného region je neomezený) má svoji vlastní oblast. Navíc víme, že každá hrana má dva vrcholy, pokud tedy sečteme stupně všech vrcholů, dostaneme dvojnásobek počtu hran. Každý vrchol má minimální stupeň tři, tak dostaneme:

$$2|E| \geq 3(|V| + 1) \quad (2.11)$$

Kombinací těchto rovnic 2.10 a 2.11 dostaneme platnost věty, stačí vyjádřit buď E nebo V z jedné rovnice a dosadit do druhé. \square

Poznámka. Symbolem $\mathcal{O}(\cdot)$ ve větě 2.1.4 máme na mysli asymptotické chování, více o třídě funkcí $\mathcal{O}(\cdot)$ v sekci 3.1.1 a v [21, 16].

2.2 Zobecnění Voroného diagramu

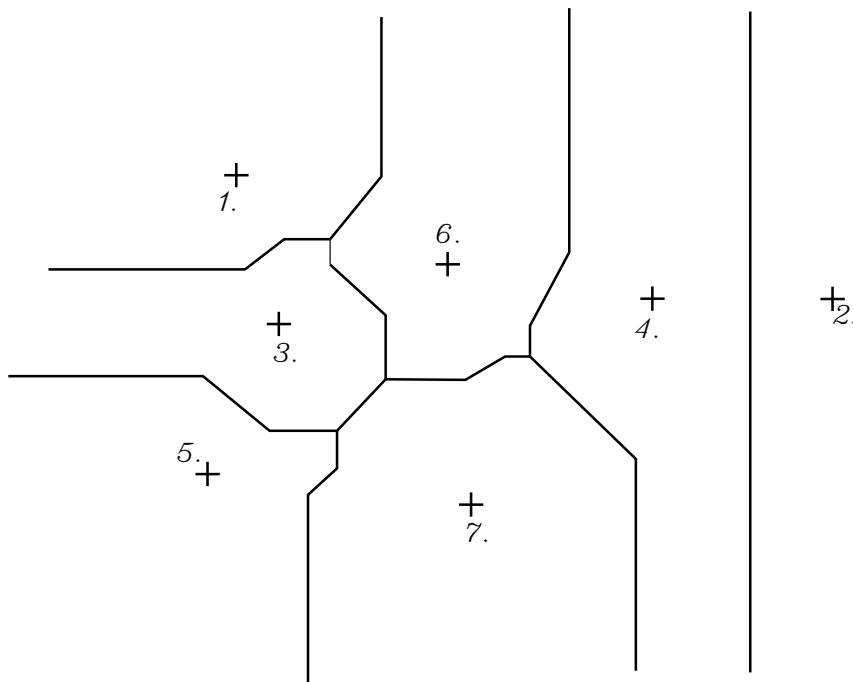
Nyní si uvedeme několik zobecnění Voroného diagramu se kterými se můžeme setkat.

2.2.1 Voroného diagram v jiných metrikách

Nejjednodušším zobecněním Voroného diagramu je použití jiné metriky než euklidovské. Metriku na metrickém prostoru \mathbb{R}^n jsme si zadefinovali v 2.1.2 jako:

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}. \quad (2.12)$$

Kde pro p a n platí, že $p, n \in \mathbb{N}$. Pokud za položíme $p = 2$, tak dostáváme euklidovskou metriku pro n -rozměrný metrický prostor. Pokud dosadíme za $p = 1$, tak dostaneme metriku, které se říká manhattanská (taxíková, šachovnicová, definice 2.1.4), více v [27]. Za p můžeme dosazovat i jiná čísla z množiny přirozených čísel a dosáhneme tak různých metrik a tedy i různých vzhledů Voroného diagramu, více v [3, 7]. Definice a věty o Voroného diagramu pro jinou než euklidovskou metriku popsané v sekci 2.1 se neliší, až na zvolené p . Ilustrace Voroného diagramu s manhattanskou metriku viz obr. 2.6.



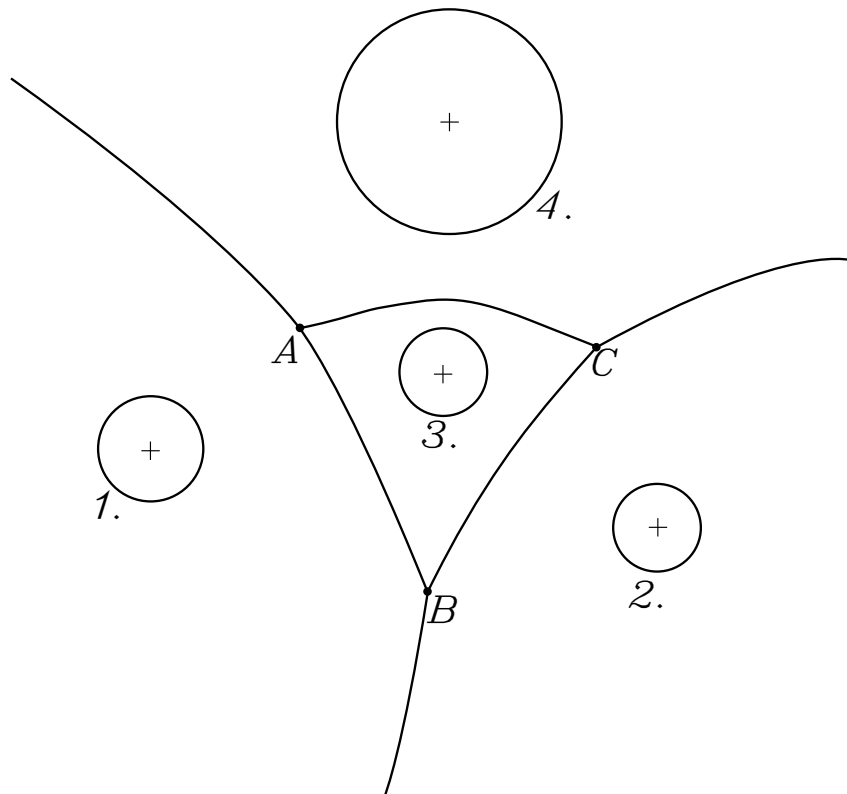
Obrázek 2.6: Znázornění Voroného diagramu s manhattanskou metriku.

2.2.2 Voroného diagram kružnic

Pro další zobecnění budeme předpokládat, že místo generující množiny obyčejných bodů, budeme uvažovat jako generující množinu, množinu kružnic (více v [12, 5]). Aby se dal Voroného diagram správně zkonstruovat tak je potřeba zadefinovat nové pojmy, jako např.: vzdálenost bodu od kružnice, která je definována bodem, který tvoří její střed a poloměrem. Tato vzdálenost je definována jako nejmenší euklidovská vzdálenost daného bodu od kružnice, nebo-li vzdálenost daného bodu od středu kružnice mínus poloměr kružnice v absolutní hodnotě.

Pro zjednodušení můžeme předpokládat, že poloměr každé kružnice je nenulový a žádné dvě kružnice se neprotínají v jednom a více bodech [12].

Pokud máme jako generující množinu, množinu tvořenou body, tak Voroného hrany jsou buď jen části nebo celé přímky. Ovšem pokud máme generující množinu tvořenou kružnicemi s předpoklady, které jsme vyslovili výše, tak Voroného hrany daného Voroného diagramu budou části nebo celé hyperboly viz obr. 2.7. Jedinou výjimkou budou kružnice jejichž poloměry by byly stejné, v tomto případě by tvořila Voroného hranu přímka.



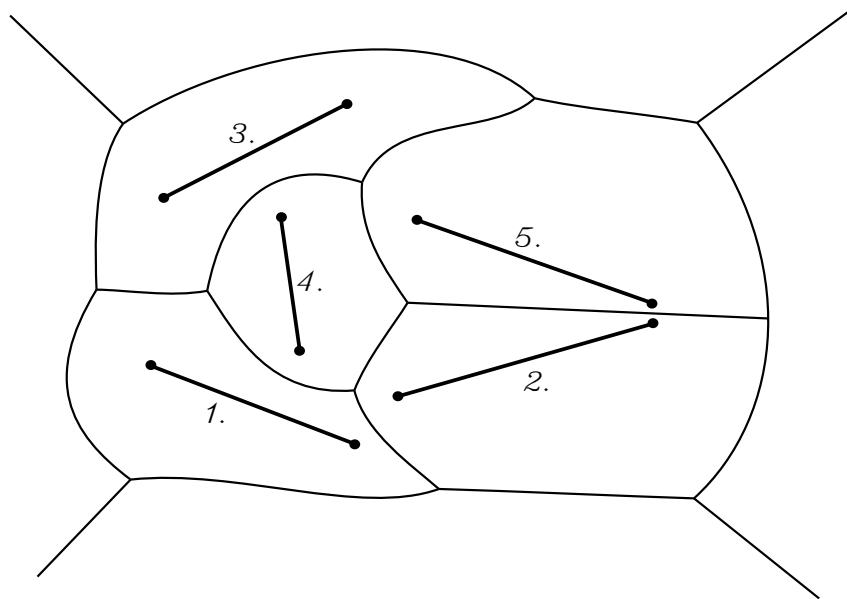
Obrázek 2.7: Znázornění Voroného diagramu, kde jsme jako generující množinu brali množinu kružnic. Čísla znázorňují jednotlivé generující objekty a písmena průsečíky Voroného hran, které tvoří hyperboly.

2.2.3 Voroného diagram úseček

Nyní předpokládáme, že generující množinu tvoří množina úseček, respektive dvojice bodů (více v [31, 20]). Pro zjednodušení uvažujme, že se každé dvě úsečky nikdy neprotínají a ani nesdílejí společný koncový bod, nebo-li jsou disjunktní. A chceme sestavit Voroného diagram pro tyto dvojice bodů/úsečky.

Z definice Voroného diagramu je, jako v předchozím případě, jasné, jak budeme postupovat a konstruovat Voroného diagram. Otázka je, jaký tvar budou mít bisektory mezi jednotlivými úsečkami. Vzdálenost objektu od bodu definujeme obdobně jako v minulém zobecnění, tedy jako nejkratší euklidovskou vzdálenost objektu od daného bodu.

Při konstrukci bisektorů můžeme dostat dva geometrické útvary ve třech případech [20]. Částečnou nebo celou parabolu jako část bisektoru obdržíme, pokud je nejbližším bodem jedné úsečky její koncový bod a u druhé úsečky je k tomuto bodu nejbližší bod z vnitřní části úsečky. V ostatních případech dostáváme jako část bisektoru přímku/polopřímku/úsečku. Výsledný Voroného diagram je tedy kombinace jednotlivých úseček, polopřímek/přímek a částí hyperbol viz obr. 2.8.



Obrázek 2.8: Znárodnění Voroného diagramu, kde jsme jako generující množinu brali množinu úseček.

2.2.4 Laguerreho Voroného diagram / Power Voroného diagram

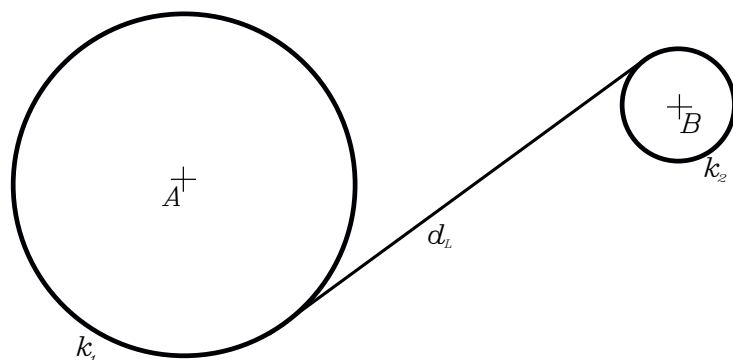
Jedná se vlastně o Voroného diagram, kde každému bodu přiřadíme jistou váhu nebo také sílu z anglického *power*, která udává jak velký vliv má daný bod na své okolí více v [2, 7, 30, 12]. Výše zmíněný Voroného diagram kružnic je vlastně speciálním případem tohoto zobecnění na dvou-dimenzionální prostor.

Prvky naší generující množiny jsou tedy $d+1$ -dimenzionální. Prvních d hodnot jsou souřadnice v prostoru a poslední hodnota zastupuje váhu daného bodu – poloměr koule ev. kružnice, jejíž střed je definován prvními d hodnotami. Oproti zobecnění s kružnicemi se liší definice vzdálenosti bodu a objektu z generující množiny. Vzdálenost dvou objektů z generující množiny se řídí tzv. *Silovou funkcí* [7]:

$$pow(x, y) = (x - y)^T(x - y) - w(p), \quad (2.13)$$

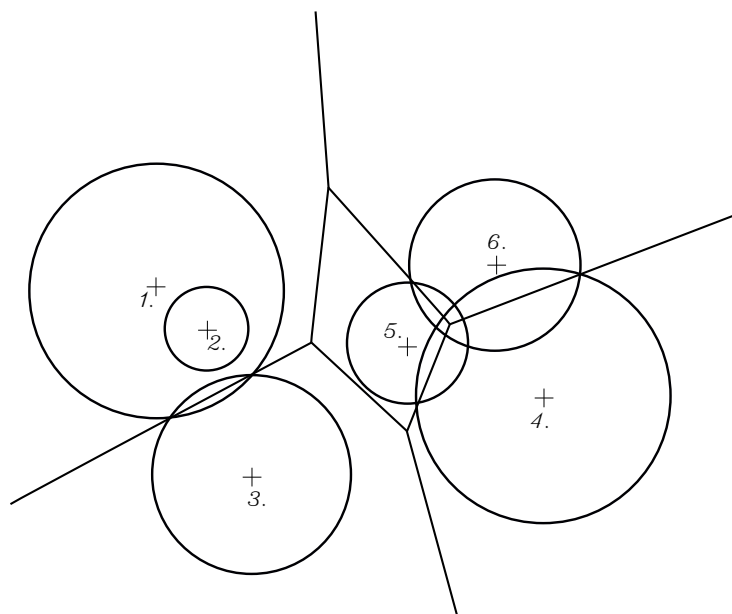
kde x je libovolný bod z prostoru na kterém se pohybujeme, p objekt z generující množiny S a $w(p)$ má význam toho, jak moc ovlivňuje bod p svoje okolí. Pokud je hodnota silové funkce $w(p)$ je kladné nebo nulové číslo pro každý prvek z generující množiny, tak se na generující množinu můžeme dívat jako na množinu koulí ev. kružnic. Proto budeme dále předpokládat tento scénář. Vzdálenost dvou prvků z generující množiny (BÚNO dvou kružnic) definujeme tak, že sestrojíme společnou tečnu pro obě kružnice a daná vzdálenost bude vzdálenost těchto dvou míst doteku viz obr. 2.9.

Pro takto definovanou vzdálenost platí, že množina bodů stejně vzdálených od dvou kružnic je přímka. Pokud se kružnice neprotínají, tak jimi tato přímka neprochází a je kolmá na úsečku tvořenou středy daných kružnic. Jestliže se kružnice protínají v jednom nebo více bodech, pak přímka stejně vzdálených bodů prochází přímo body ve kterých se kružnice protínají. Pokud leží jedna kružnice v druhé a neprotínají se v žádném bodě, pak přímka stejně vzdálených bodů leží



Obrázek 2.9: Znázornění vzdálenosti dvou kružnic, k_1 a k_2 se středy v bodech A a B , zavedené pro konstrukci Legaurreho Voroného diagramu.

mimo obě kružnice a je kolmá na přímkou tvořenou středy kružnic [12]. Výsledný diagram poté vypadá jako na obrázku 2.10.

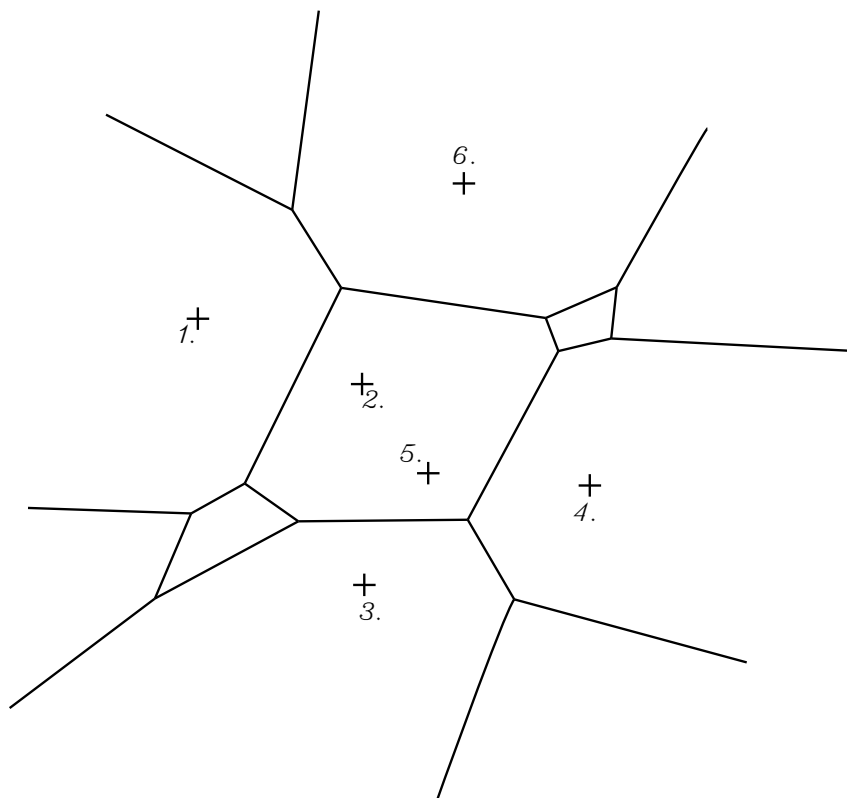


Obrázek 2.10: Znázornění Power Voroného diagramu pro generující množinu s kladným „vlivem“/silou.

2.2.5 Voroného diagram vyšších řádů

Voroného diagram prvního řádu je „standardní“ Voroného diagram definovaný na začátku této kapitoly. Vyšší řády Voroného diagramu, více v [7, 31, 38], mohou být až $n - 1$, kde n je počet prvků v generující množině. Voroného diagram k -tého řádu, kde $k \in \widehat{n - 1}$, je Voroného diagram, u kterého nebereme pro konstrukci bisektorů a Voroného regionů *nejbližší* bod, ale k -tý nejbližší bod. Jistým extrémem je pokud za k volíme $n - 1$. Pak se jedná Voroného diagram, kde

místo nejbližšího bodu bereme naopak bod nejvzdálenější. Na toto zobecnění (zvyšování řádů) můžeme aplikovat již zmíněné nebo v budoucnu zmíněné zobecnění a dosáhnout tak zajímavých dláždění. Znázornění Voroného diagramu řádu dva viz obr. 2.11.



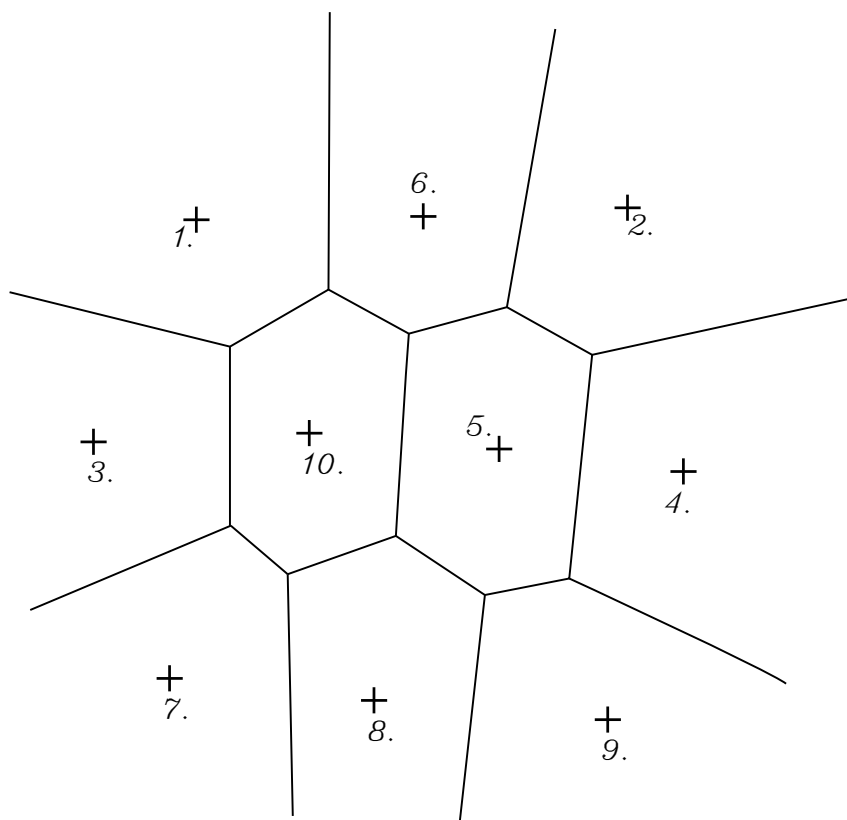
Obrázek 2.11: Znázornění Voroného diagramu druhého řádu.

2.2.6 Centralizovaný/Těžišťový Voroného diagram

Dalším příkladem zobecnění je Těžišťový Voroného diagram [39, 33], u kterého je každý bod z generující množiny umístěn přesně v těžišti daného Voroného regionu. Dostáváme tak Voroného diagram s Voroného regiony stejného tvaru (výjimkou jsou regiony, které jsou tvořeny body co se nacházejí na konvexním obalu generující množiny). Znázornění Těžišťového Voroného diagramu viz obr. 2.12.

2.2.7 Další zobecnění Voroného diagramu

Zobecnění Voroného diagramu existuje mnoho a díky vzájemné kombinaci různých zobecnění je takřka nemožné a zbytečné všechny shrnout do jednoho pojednání, zmiňme alespoň pár dalších zobecnění: Möbioův diagram, Anizotropní diagram, Voroného diagram konvexních polygonů. Více v [2, 7].



Obrázek 2.12: Znázornění Těžišťového Voroného diagramu.

Kapitola 3

Základní algoritmy

V této kapitole si představíme několik základních algoritmů pro sestavení Voroného diagramu. Podíváme se na jejich složitosti a pokusíme se zjistit, který z nich je neoptimálnější vzhledem k dolnímu odhadu konstrukce Voroného diagramu.

3.1 Složitosti algoritmů a dolní odhad složitosti Voroného diagramu

3.1.1 Složitosti algoritmů

Abychom mohli najít neoptimálnější algoritmus ke konstrukci Voroného diagramu, musíme zjistit jakou má daný algoritmus *složitost*. Otázka zní co to vlastně složitost algoritmu je? Proto si složitosti rigorózně zdefinujeme.

Jak nejlépe odhadnou dobu, jak dlouho nějaký náš algoritmus bude zpracovávat daná vstupní data? Jedna z možností co nás napadne je algoritmus naprogramovat a změřit čas než zpracuje všechna data. Tento způsob je z teoretického hlediska nevhodný, protože na rychlosti zpracování dat může mít vliv: programovací jazyk, operační systém, parametry (hardware) počítače, architektura hardwaru, formát vstupních dat, . . .

Další možnost je použití nějaké metody pro odhad rychlosti nezávislé na technických parametrech. Pokusíme se najít tzv. *časovou složitost algoritmu*, která nám řekne jak přibližně dlouho daný algoritmus poběží v závislosti na velikosti vstupu. Pro naše úvahy se můžeme omezit na případ, kdy platí, že čím větší vstup tím delší doba běhu algoritmu (ne vždy tomu tak musí být).

Jak „zjednodušeně“ najít složitost algoritmu? Určíme maximální možný počet elementárních operací (sčítání, odčítání, násobení, dělení, porovnání, logické *or* a *and*, . . .) ev. jejich nejlepší horní odhad. Dostaneme funkci, která nám pro vstupní soubor o n prvcích vrátí součet několika (ev. jednoho) členů v závislosti na velikosti vstupního souboru ozn. $f(n)$.

Počítače v dnešní době jsou dostatečně rychlé na to, že pro malé vstupy běží různé algoritmy pro zadanou úlohu téměř stejně rychle. Proto nás malé vstupy nemusí trápit. Důležité pro nás je jak se budou různé algoritmy chovat pro velké vstupy. Pro ty ovšem platí, že v zobrazení $f(n)$ se nejvíce projeví „nejméně silnější“ členy. Proto ve výsledné formuli budeme chtít ponechat pouze ty členy, které se s zvětšujícím se n rostou nejrychleji a ostatní vynecháme, stejně tak nás nezajímají konstanty před všemi členy $f(n)$, které se u „slabších“ členů nedokážou projevit a u „silných“ členů nemají vliv na výslednou časovou složitost.

Pro zjednodušení bychom mohli tvrdit, že v $f(n)$ jsou nejsilnější exponenciální členy (např. 2^n (ovšem pouze pro základ větší než jedna)) „>“ polynomiální členy (např. n^3) „>“ lineární členy (např. $3n$) „>“ logaritmické členy (např. $3 \cdot \log n$ (základ logaritmu nemá na nerovnosti vliv)) „>“ konstantní členy (např. 1000). Proto zavádíme třídy funkcí: \mathcal{O} , Ω a Θ , které vyjadřují tzv. *asymptotickou časovou složitost* a používáme je k popisu složitostí algoritmů.

Definice 3.1.1 (Třídy funkcí $\mathcal{O}(g(n))$). Mějme dvě funkce $f, g : \mathbb{N} \rightarrow \mathbb{R}$. Řekněme, že funkce $f(n)$ je třídy $\mathcal{O}(g(n))$, jestliže existuje taková kladná reálná konstanta $c \in \mathbb{R}_0^+$ a $n_0 \in \mathbb{N}$ tak, že pro všechna $n \geq n_0$ platí $f(n) \leq cg(n)$. Funkci $g(n)$ se pak říká *asymptoticky horní odhad funkce $f(n)$* .

Definice 3.1.2 (Třídy funkcí $\Omega(g(n))$). Mějme dvě funkce $f, g : \mathbb{N} \rightarrow \mathbb{R}$. Řekněme, že funkce $f(n)$ je třídy $\Omega(g(n))$ pokud existuje taková kladná reálná konstanta $c \in \mathbb{R}_0^+$ a $n_0 \in \mathbb{N}$ tak, že $\forall n \geq n_0$ platí $f(n) \geq cg(n)$. Funkci $g(n)$ říkáme *asymptoticky dolní odhad funkce $f(n)$* .

Definice 3.1.3 (Třídy funkcí $\Theta(g(n))$). Řekněme, že funkce $f(n)$ je třídy $\Theta(g(n))$, jestliže $f(n)$ je jak třídy $\mathcal{O}(g(n))$, tak třídy $\Omega(g(n))$.

Poznámka. V kapitolách Rozděl a panuj 3.4 a Průnik polorovin 3.2 se setkáme s rekurzivním algoritmem, kde budeme jeho celkovou složitost značit $\mathcal{T}(n)$.

Definice 3.1.4 (Třída \mathcal{P}). Třídou \mathcal{P} nazýváme množinu všech problémů, které jsme schopni řešit s časovou složitostí $\mathcal{O}(n^c)$ pro nějakou konstantu $c > 0$, tj. v *polynomiálním čase*. Problémům v třídě \mathcal{P} se jinak říká *efektivně řešitelné*.

Velmi užitečným nástrojem při odhadu složitosti je tzv. *redukce* daného problému na jiný. Redukci využíváme v situacích kdy místo přímého řešení daného problému, tento problém převedeme na problém podobný, který je již vyřešený. Těto operace se hojně využívá např. v algoritmizaci, kdy už máme nějakou knihovnu základních algoritmů a úkol je řešit nějaký nový problém. Místo psaní speciálního algoritmu je jednodušší aktuální nový problém zredukovat na nějaký již známý problém, ke kterému už máme hotový algoritmus v naší knihovně. Vzhledem k tomu, že všechny problémy na které narazíme budou řešitelné maximálně v polynomiálním čase, tak si nyní formalizujeme polynomiální převod dvou problémů.

Definice 3.1.5. Mějme dva problémy P_1, P_2 . *Polynomiálním převodem/redukci P_1 na P_2* rozumíme algoritmus $R \in \mathcal{P}$ takový, že pro všechny vstupy w platí, že $P_1(w) = P_2(R(w))$.

Z definice polynomiálního převodu tedy dostáváme, že pokud umíme efektivně řešit problém P_2 , pak problém P_1 také můžeme efektivně vyřešit tím, že jej převedeme na známý problém P_2 .

Pokud víme, že problém P_2 je časově obtížně řešitelný a polynomiálně jej převedeme na problém P_1 , pak problém P_1 je alespoň tak složitý jako problém P_2 .

Příklad 3.1.1. *Pokusme se najít složitost třídícího algoritmu Bubble-sort, který dokáže seřadit n prvků podle velikosti vzhledem k nějaké váze. Nechť vektor $[n]$ je vektor o n prvcích, který chceme seřadit.*

Z řadícího algoritmu 1 můžeme vidět, že vnitřní cyklus provede při nejhorším $(n-1) + (n-2) + \dots + (1)$ operací (celkově jde tedy o $n-1$ sčítanců). To je ale součet $n-1$ členů aritmetické posloupnosti: $S_n = (n-1) \cdot \frac{(n-1)+1}{2} = \frac{n^2-n}{2}$. Nejrychleji rostoucí člen je očividně n^2 , ostatní členy tedy můžeme zanedbat. Vzhledem k tomu, že se jedná o nejhorší, ale i nejlepší případ, tak výsledná složitost algoritmu Bubble-sort bude $\Theta(n^2)$.

Více informací ke složitosti algoritmů a o samotných algoritmech lze nalézt v [21, 16, 20].

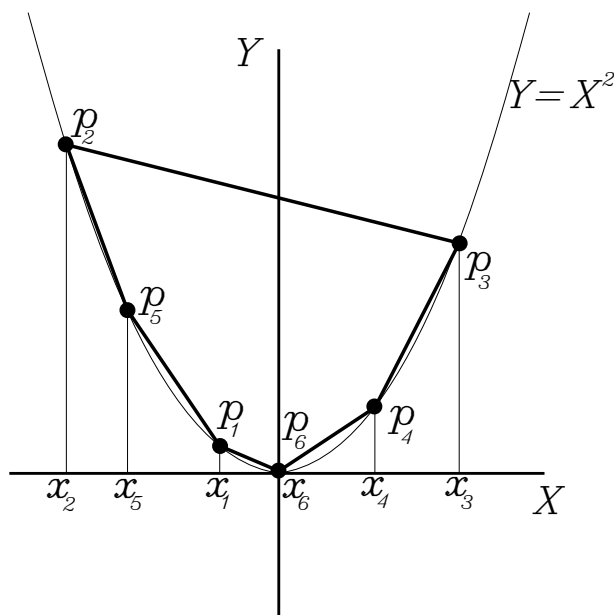
Algorithm 1 Bubble-sort

```
for i = 0 ; i < n ; i++ do
  for j = 0 ; j < n - i - 1 ; j++ do
    if vektor[j+1] < vektor[j] then
      temp = vektor[j+1];
      vektor[j+1] = vektor[j];
      vektor[j] = temp;
    end if
  end for
end for
```

3.1.2 Dolní odhad složitosti algoritmu konstrukce Voroného diagramu

Ještě než se podíváme na první algoritmus vhodný ke konstrukci Voroného diagramu, tak musíme zjistit jaký je dolní odhad složitosti konstrukce samotného Voroného diagramu. Pak budeme schopni ověřit, jestli jsme našli optimální algoritmus a nebo jestli ho ještě můžeme vylepšit, více v [7, 22].

Mějme n reálných čísel $x_1, \dots, x_n : \forall i \in \hat{n}, x_i \in \mathbb{R}$. Z Voroného diagramu zkonstruovaného nad generující množinou S definovanou jako $S = \{p_i = (x_i, x_i^2) \mid \forall i \in \hat{n}\}$, jsme schopni [7] najít vrcholy konvexního obalu množiny S v lineárním čase. Posloupnost všech bodů p_i z S , která začíná bodem nejvíce nalevo, je seřazena podle rostoucích hodnot x_i viz obr. 3.1. Díky [22] dostáváme, že nalezení konvexního obalu množiny a Voroného diagramu je alespoň stejně složité jako seřazení n reálných čísel, což vyžaduje složitost $\Theta(n \log(n))$.



Obrázek 3.1: Důkaz že složitost $\Omega(n \log(n))$ je spodní hranice konstrukce Voroného diagramu.

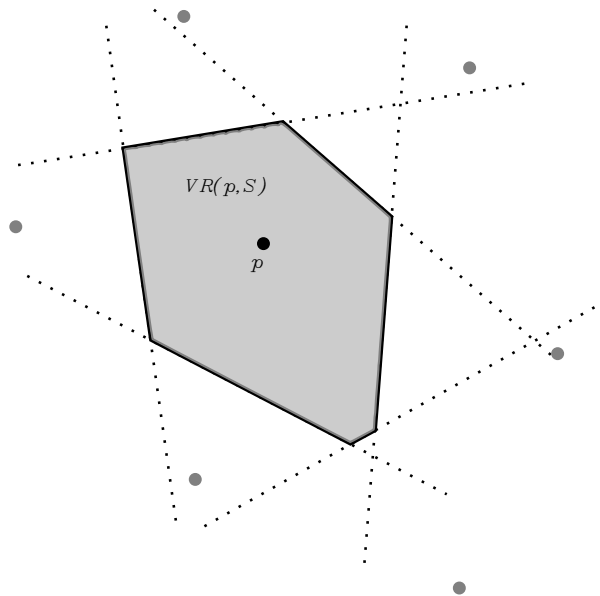
Víme tedy, že úlohu seřazení n reálných čísel lze polynomiálně redukovat na konstrukci Vo-

roného diagramu a tedy, že složitost konstrukce Voroného diagramu je v nejhorším případě $\Omega(n \log(n))$.

3.2 Průnik polorovin

Jedná se o přímou aplikaci definice Voroného diagramu. Napočítáme ke každému bodu z množiny S Voroného region zvlášť. Z alternativní definice 2.1.9 víme, že Voroného region pro jeden daný bod je průnik $n - 1$ polorovin, jejichž hranice jsou bisektory daného bodu s ostatními body. Tento postup aplikujeme na všechny body v množině S .

Otázka zní jaká bude složitost takového algoritmu? Víme, že v množině S je n bodů a pro každý z nich musíme vytvořit Voroného region (až na poslední bod, kdy bude poslední Voroného region dopočítán komplementárně). Stačí nám tedy zjistit složitost pro jeden Voroného region a tuto složitost vynásobit $n - 1$. Abychom zjistili složitost nalezení jednoho Voroného regionu, uděláme odbočku do *Výpočetní geometrie - problém průniku polorovin*, více v [20, 22].



Obrázek 3.2: Znázornění průniku polorovin pro výchozí bod p . Tečkovanou křivkou jsou znázorněny jednotlivé hranice polorovin a šedou barvou je vyznačena oblast vzniklého Voroného regionu.

Věta 3.2.1. *Průnik n polorovin může být nalezen se složitostí $\Theta(n \log(n))$.*

Důkaz. Navrhněme rekurzivní, *Divide & Conquer-D&C* (česky: *Rozděl a panuj*), algoritmus (více o tomto algoritmu v sekci 3.4):

Označme $\mathcal{T}(n)$ celkovou složitost tohoto rekurzivního algoritmu, pak:

$$\mathcal{T}(n) = 2\mathcal{T}(n/2) + \mathcal{O}(n) = \mathcal{O}(n \log(n)) \quad (3.1)$$

Důkaz toho, že rovnice 3.1 platí, si ukážeme v sekci 3.4 více v [24, 25, 21, 16].

Algorithm 2 D&C algoritmus pro průnik polorovin

Vstup: n polorovin definovaných pomocí přímk a bodů do nich náležících.

Výstup: průnik těchto n polorovin jako konvexní polygon.

1. Rozdělení polorovin na dvě množiny stejné velikosti.
 2. Rekurzivní provedení průniku pro jednotlivé podproblémy.
 3. Sjednocení jednotlivých řešení průnikem dvou konvexních polygonů.
-

Díky našemu algoritmu 2 jsme našli horní hranici pro nalezení průniku n polorovin. Stačí tedy najít spodní hranici a ukázat že je rovna $\Omega(n \log(n))$. To uděláme tak, že ukážeme, že problém *třídění* je polynomiálně redukovatelný na problém *průniku polorovin* [21].

Mějme n reálných čísel $x_1, \dots, x_n \mid \forall i \in \hat{n}, x_i \in \mathbb{R}$ a necht' $H_i, \forall i \in \hat{n}$ jsou poloroviny, které definujeme přímkou sklonu x_i tečny k parabole $y = x^2$. Průnik těchto polorovin je konvexní množina jejíž po sobě jdoucí hrany jsou seřazeny podle sklonu. Jakmile je tento průnik utvořen, jsou hodnoty x_i seřazené podle velikosti. Nalezení průniku polorovin je tedy alespoň tak složité jako seřazení n čísel. Tím dostáváme hledaný dolní odhad, [22]. \square

Víme tedy, že složitost napočítání jednoho Voroného regionu, je $\mathcal{O}((n-1) \log(n-1))$ s tím, že tuto operaci provádíme pro $n-1$ bodů, tedy celková složitost *Průniku polorovin* je $(n-1) \cdot \mathcal{O}((n-1) \log(n-1)) = \mathcal{O}((n^2 - 2n + 1) \log(n-1)) = \mathcal{O}(n^2 \log(n))$. Tento algoritmus tedy není moc efektivní, jsme schopni najít i algoritmy, které dokážou napočítat celý Voroného diagram se složitostí $\mathcal{O}(n \log(n))$. Více v [22, 20, 18, 8, 38].

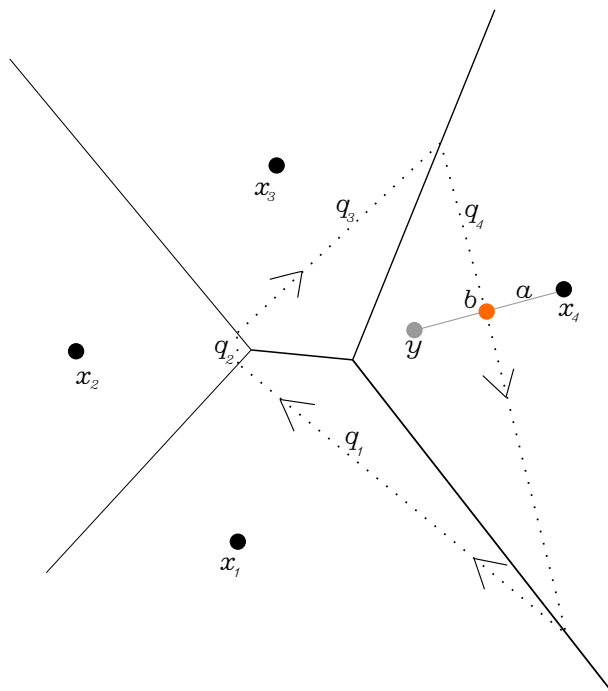
3.3 Inkrementální konstrukce

Začneme tím, že najdeme Voroného diagram pro jednoduchý případ např. pro dva nebo tři body. A poté postupně přidáváme jednotlivé body a dopočítáváme pro každý z nich nový Voroného region do již vzniklého diagramu.

Přidání nového bodu probíhá tak, že k novému bodu (bod y na obr. 3.3) najdeme nejbližší bod, který jsme k hledání Voroného diagramu již použili (bod x_4) a ke spojnici (úsečka a) těchto dvou bodů, sestrojíme osu (osa q_4), která bude procházet bodem na této spojnici (bod b), tento bod bude stejně daleko od obou bodů. Nyní „půjdeme“ podél této kolmice ke spojnici BÚNO ve směru hodinových ručiček a jakmile narazíme na bod (průsečík os q_4 a q_1), kde se tato kolmice protne s již existující částí Voroného diagramu (nějaká Voroného hrana), najdeme k této Voroného hraně body (body x_1 a x_4), k jejichž spojnici tvoří kolmici a k tomuto bodu (bod x_1) vytvoříme spojnici s naším nově přidaným bodem a kolmice k této spojnici bude další nová Voroného hrana. Tento postup opakujeme dokud se nevrátíme do místa odkud jsme začali nebo do „nekonečna“.

Nový bod může mít až $n-1$ hran, kde $|S| = n$, z toho dostáváme celkovou složitost *Inkrementální konstrukce* $\mathcal{O}(n^2)$. Vidíme tedy, že už inkrementální konstrukce je rychlejší než průnik polorovin. Pořád jsme se ale dostatečně nepřiblížili k dolní hranici.

Nyní se podíváme na algoritmy, které bychom již mohli považovat za optimální. Více o inkrementální konstrukci v [26, 40, 38].



Obrázek 3.3: Znázornění *Inkrementální konstrukce* pro tvorbu Voroného diagramu, kde šedý bod (y) je bod, který jsme nově přidali, tečkovaně je naznačená trajektorie jak se změní Voroného hrany a oranžový bod, je bod kde začínáme a zároveň končíme s tvorbou nových hran. Body ($x_1 - x_4$) jsou body „původního“ Voroného diagramu.

3.4 Rozděl a panuj

V anglické literatuře se můžeme setkat s názvem *Divide & conquer* nebo zkráceně *D&C*. Tento typ algoritmu jsme již zmínili v kapitole 3.2 *Průnik polorovin* ve které jsme nechali nedokázanou rovnici složitosti k danému rekurzivnímu algoritmu 3.1, který je ovšem platný ve vší obecnosti, jak si ukážeme.

Princip D&C algoritmu: pomocí pomyslné BÚNO vertikální přímky M , rozdělíme množinu S na levou L a pravou P část, které jsou podobně veliké (z pohledu mohutnosti, tedy počtu bodů/prvků) tak, že každý bod v L leží nalevo od M a každý bod v P leží napravo od M . Pokud by se náhodou nějaké body nacházely přesně na M , tak tyto body napůl rozdělíme mezi L a P .

Tento postup opakujeme dokud se jednotlivé části neskládají dvou nebo tří bodů. Voroného diagram takové množiny (tvořené pouze dvěma nebo třemi body z S) jsme schopni spočítat přímo se složitostí $\mathcal{O}(1)$. Poté rekurzivně přidáváme ostatní části množiny S .

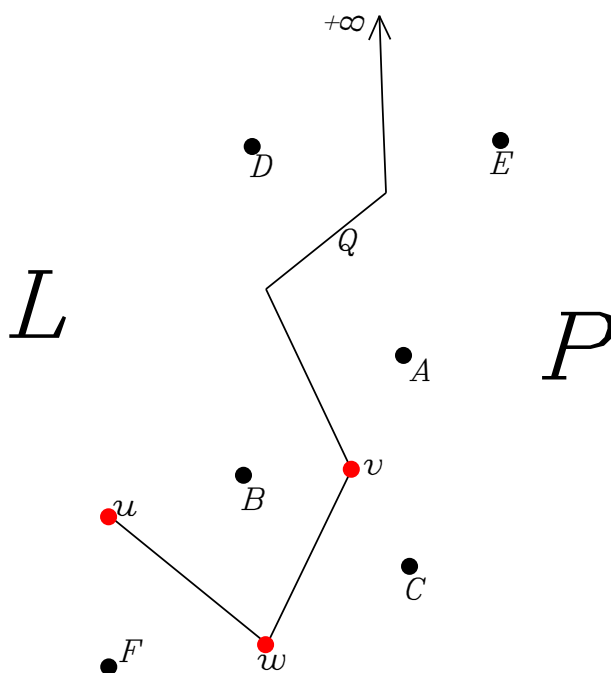
Pokusíme se najít Voroného diagram množin L a P zvlášť, ozn. $V(L)$ a $V(P)$ a jejich sjednocením poté dostaneme Voroného diagram celého S tzn. $V(S)$. Uvažujme křivku Q , jako množinu bodů z prostoru, které jsou stejně vzdálené od bodů v L a P . Q tedy bude křivka složená s Voroného hran z $V(S)$, které sdílejí $V(i)$, $i \in \hat{L}$ a $V(j)$, $j \in \hat{P}$. Ukážeme, že Q je polygonální křivka.

Věta 3.4.1. *Každá horizontální přímka se protne s Q alespoň v jednom bodě.*

Důkaz. Uvažujme horizontální přímkou H a libovolné body $u \in L$, $v \in P$ tak, že oba neleží na M . Necht' bisektor bodů u a v protne H v nějakém bodě z . Víme že takový bod existuje, protože jediná situace kdy by neexistoval by nastala pokud by oba body (u a v) ležely na M . Označme $LEVO(H, u, v)$ jako množinu všech bodů z H , které se nachází nalevo od z a $PRAVO(H, u, v)$ jako množinu všech bodů z H , které se nachází nalevo od z . Nyní udělejme průnik všech množin $LEVO(H, u, v)$ přes všechny u a v , které splňují předpoklady a stejně tak i pro $PRAVO(H, u, v)$. Tyto průniky budou určitě neprázdné. Víme, že H určitě obsahuje body, které jsou neblíže k nějakému bodu z L a také určitě obsahuje body, které budou nejbliže k nějakému bodu u P . Ze spojitosti je jisté že tedy na H leží bod, který je stejně vzdálen od P a L . \square

Věta 3.4.2. *Křivka Q má průnik s každou horizontální přímkou maximálně v jednom bodě tzn. Q je monotónní.*

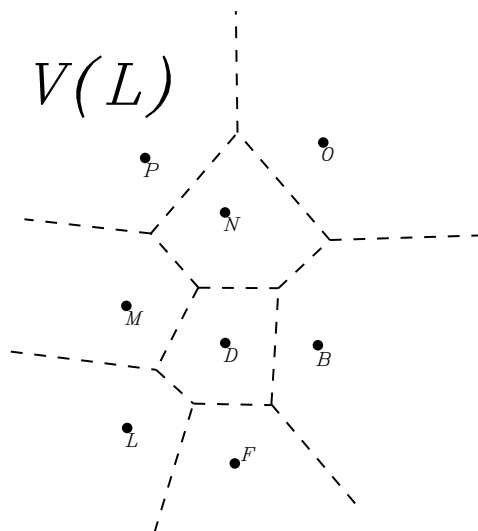
Důkaz. Q se skládá pouze z úseček a polopřímek. Ukážeme, že pokud se pohybujeme po Q ze shora (kladné „nekonečno“ na ose y) směrem dolů (záporné „nekonečno“ na ose y) tak, že se y -ová složka aktuální pozice na Q nezvětšuje. Pro spor předpokládejme, že w je první bod ve kterém se Q otočí směrem nahoru viz obr. 3.4. Hrana s počátkem a koncem v bodech v a w je bisektor bodů C a B . A hrana s počátkem a koncem v bodech u a w je bisektor bodů F a B . Vzhledem k tomu, že konstrukce Q rozděluje L a P dostáváme, že body F a C se nacházejí v P a bod B je v L ev. body F a C se nacházejí v L a bod B je v P . V obou případech dostáváme spor, který způsobil bod F . \square



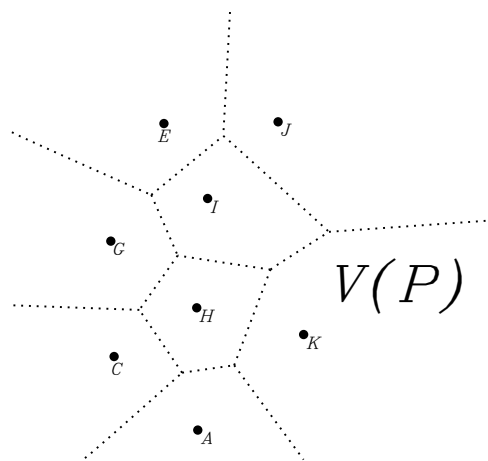
Obrázek 3.4: Důkaz toho, že Q je monotónní křivka. Malými písmeny a červenou barvou značíme hraniční body jednotlivých úseček ze kterých je Q složena. Velkými písmeny značíme křivku Q , jednotlivé body z S popřípadě L a P .

Nyní víme, že Q je monotónní polygonální křivka a má tedy smysl mluvit o „pravé“ a „levé“ straně od Q . Křivka Q má vlastnost, že pro každý bod, který se od ní nachází nalevo platí, že je nejbližší nějakému bodu v L a každý bod který se nachází napravo platí, že je nejbližší nějakému bodu v P .

Uvažujme nyní Voroného diagramy $V(L)$ (obr. 3.5) a $V(P)$ (obr. 3.6) a jejich sdruženou verzi obr. 3.7.



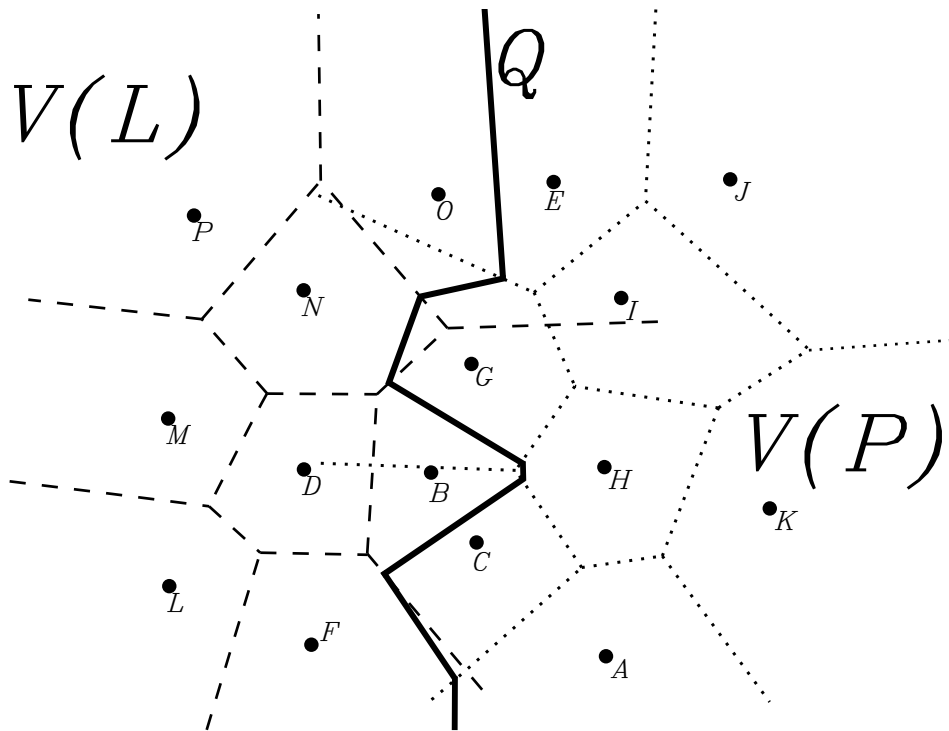
Obrázek 3.5: Znázornění Voroného diagramu pro body z množiny L (čárkovaně).



Obrázek 3.6: Znázornění Voroného diagramu pro body z množiny P (tečkovaně).

Ty části $V(P)$, které leží nalevo od Q nemají žádný vliv na konečný Voroného diagram, stejně tak i části $V(L)$, které jsou napravo od Q . Krok sjednocení $V(L)$ a $V(P)$ je tedy odříznutí těch částí, které nemají na výsledný Voroného diagram $V(S)$ vliv. Náznak algoritmu:

Důležité je jak rychle jsme schopni najít křivku Q , vzhledem k tomu, že v nejhorším případě bude mít na tvar Q vliv každý bod z L a P , tak časová složitost nalezení Q bude $\mathcal{O}(n)$, kde $|S| = n$. Pro zkonstruování Q budeme muset začít nalezením těch částí Q , které jsou v „nekonečnu“, tedy



Obrázek 3.7: Znázornění Voroného diagramů pro body z množin L a P sjednocené do jednoho obrázku a rozlišené tečkovanými a čárkovanými křivkami.

polopřímek. Z vlastností Voroného diagramu víme, že Voroného hrany, které jsou polopřímky, jsou tvořeny generujícími body, které se nachází na konvexním obalu množiny S . Musíme proto nalézt body z konvexního obalu množiny S takové, které nejsou přítomny v konvexním obalu množiny L nebo P .

Věta 3.4.3. *Mějme Voroného diagram nad množinou S tak, že $|S| = n$. Pak konvexní obal množiny S jsme schopni nalézt v lineárním čase.*

Důkaz. Prošetřujeme Voroného hrany dokud nenalezneme polopřímku r (neomezená Voroného hrana resp. Voroného hrana je polopřímka). Řekněme, že Voroného region nalevo od r je $VR(p_i)$, kde $p_i \in S : i \in \hat{n}$. Pak vrchol p_i je prvek konvexního obalu nad množinou S . Prohledávejme další hrany $VR(p_i)$ dokud opět nenalezneme další polopřímku (neomezenou Voroného hranu),

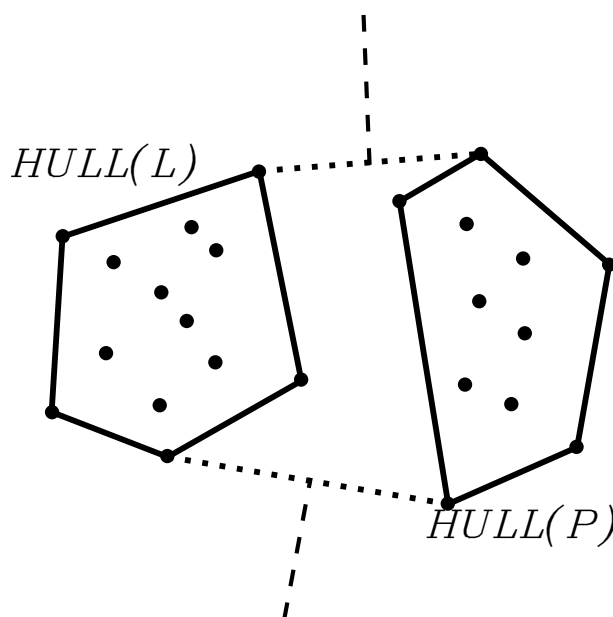
Algorithm 3 D&C algoritmus-náznak

1. Rozdělíme S na dvě množiny podobné mohutnosti L a P , to můžeme provést se složitostí $\mathcal{O}(n)$.
 2. Najdeme $V(L)$ a $V(P)$ rekurzivně s časovou složitostí $\mathcal{T}(n/2)$.
 3. Zkonstruujeme Q .
 4. „Zahodíme“ všechny segmenty, které nemají vliv na výsledný Voroného diagram, tzn. ty části $V(L)$, které jsou napravo od Q a ty části $V(P)$, které jsou nalevo od Q .
-

tím dostaneme další prvek konvexního obalu $p_j : i \neq j$. Nyní budeme skenovat region $VR(p_j)$ atd. dokud se nenavrátime do $VR(p_i)$. Hrana je nalezena pouze tehdy pokud je prohledán jeden region který pomáhá tvořit. Vzhledem k tomu, že každá hrana je přilehlá pouze dvěma regionům, tak není žádná hrana nalezena více jak dvakrát a stačí nám tedy lineární složitost. \square

Poznámka. Algoritmus vhodný k nalezení konvexního obalu sjednocení dvou konvexních polygonů lze nalézt v [22].

K nalezení polopřímek v Q použijeme $V(P)$ a $V(L)$ a jejich konvexní obaly $HULL(L)$ a $HULL(P)$, které budeme schopni najít v lineárním čase díky větě (3.4.3). Polopřímky jsou poté bisektory úseček, které vzniknou spojením $HULL(L)$ a $HULL(P)$ viz obr. 3.8.

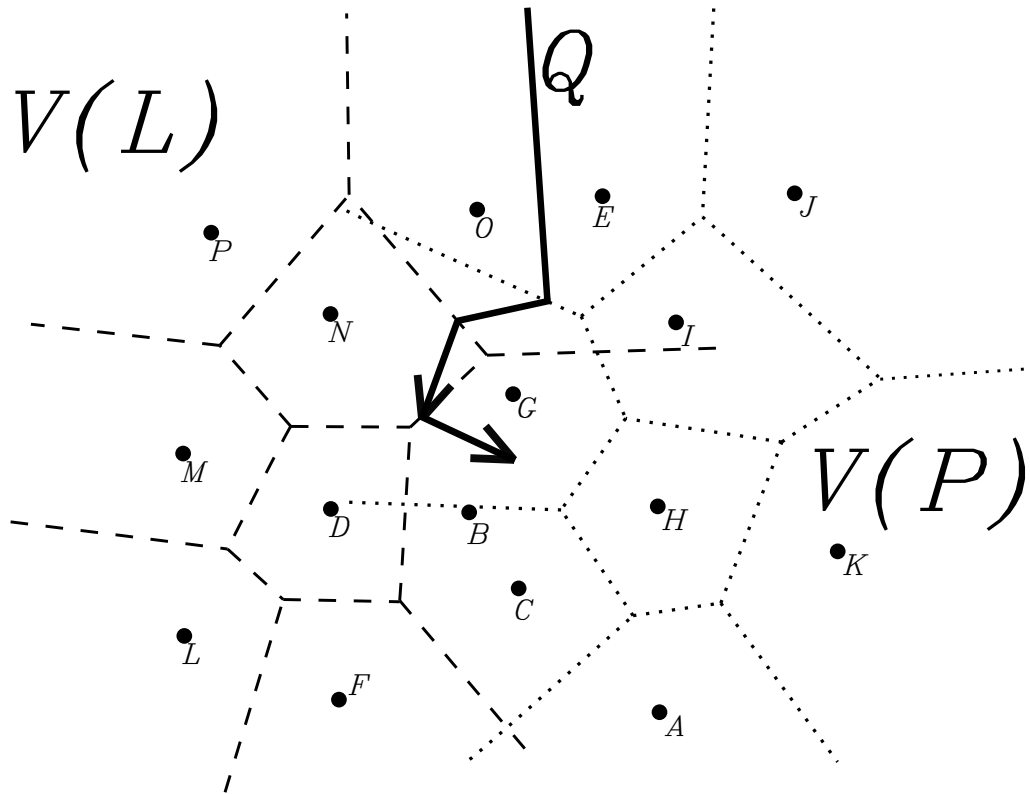


Obrázek 3.8: Znázornění nalezení polopřímek Q jako bisektorů úseček, které vzniknou jako spojnice bodů při hledání konvexního obalu ze sjednocení dvou konvexních polygonů. Konvexní polygony značíme $HULL(L)$ a $HULL(P)$ a úsečky u_1 a u_2 , bisektory jsou naznačeny čárkovanou křivkou a tečkovaně značíme úsečky.

Podařilo se nám nalézt polopřímky, kterými Q začíná a končí. Nyní se odkážeme na obrázek 3.9. První polopřímka je tvořena body O a E , jeden její konec je tedy někde v „nekonečnu“ a ten druhý je bod, je průsečík ležící na první Voroného hraně se kterou se Q protne, v našem případě se jedná o hranu tvořenou regiony bodů G a E . Na další části se budou podílet dva body a každý z nich bude z jiné množiny, jeden z nich z L a ten druhý z P . Průsečík bude tedy Voroného vrchol a bude mít stupeň tři tzn. z množin L a P k němu budou stejně vzdáleny tři body, kde vždy alespoň jeden z nich bude z jiné množiny než ty další dva. Navíc víme, že Q je monotónní a tedy víme přesně které body se budou na tvaru Q podílet. Takto jsme schopni nalézt celou křivku Q se složitostí $\mathcal{O}(n)$ protože $|L| + |P| = n$.

Víme z [21, 16], že celková složitost tohoto rekurzivního algoritmu je: $\mathcal{T}(n) = 2\mathcal{T}(n/2) + \mathcal{O}(n)$.

Uvažujme rekurzivní algoritmus, který vstup rozloží na a podproblémů velikosti n/b a z jejich výsledků složí výsledné řešení v čase $\Theta(n^c)$. Za předpokladu, že n je mocninou čísla b s tím, že



Obrázek 3.9: Znázornění nalezení jednotlivých částí Q .

případné zaokrouhlování zanedbáme bude příslušná rekurence vypadat následovně:

$$\begin{aligned} \mathcal{T}(1) &= 1, \\ \mathcal{T}(n) &= a \cdot \mathcal{T}(n/b) + \Theta(n^c). \end{aligned} \tag{3.2}$$

Platí tedy, že v našem případě bude poslední člen rovnice 3.2 $\Theta(n^c)$ roven $\Theta(n)$, resp. $c = 1$. Zároveň pro jednoduchost nám stačí předpokládat, že $a = b$, protože přesně takto jsme algoritmus popsali. a a b jsou pouze čísla, navíc jsou si rovna a jak můžeme vidět v rovnici 3.4, tak se hned na druhém řádku pokrátí, i kdyby se nepokrátila, tak se pořád jedná o nějaké konečné číslo, které nemá na výslednou složitost žádný vliv. Proto nám stačí uvažovat, že $a = b = 2$ a máme:

$$\begin{aligned} \mathcal{T}(1) &= 1, \\ \mathcal{T}(n) &= 2 \cdot \mathcal{T}(n/2) + \Theta(n). \end{aligned} \tag{3.3}$$

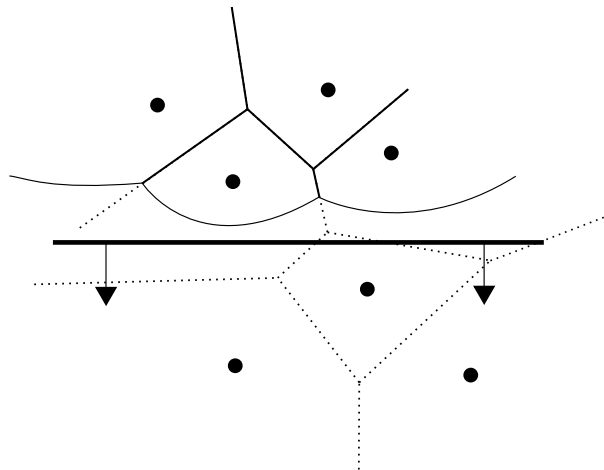
Lze dokázat že složitost 3.3 je $\mathcal{O}(n \log(n))$. Použijeme substituční metodu. Předpokládejme, že $\mathcal{T}(n) = 2\mathcal{T}(n/2) + \mathcal{O}(n)$. A předpokládejme, že jsme odhadli přímé vyjádření ve tvaru: $\mathcal{T}(n) = \mathcal{O}(n \log(n))$ a chceme ukázat (z definice 3.1.1), že $\mathcal{T}(n) \leq Cn \log(n)$, pro nějaké $C > 0$. Stanovíme indukční předpoklad $\mathcal{T}(n/2) \leq C(n/2) \log(n/2)$. Dosadíme indukční předpoklad do rekurentního vztahu a ukážeme jeho platnost vyjádřením původně odhadnutého vztahu.

$$\begin{aligned}
\mathcal{T}(n) &\leq 2(C(n/2) \log(n/2)) + n, \\
\mathcal{T}(n) &\leq Cn \log(n/2) + n, \\
\mathcal{T}(n) &= Cn \log(n) - Cn \log(2) + n, \\
\mathcal{T}(n) &= Cn \log(n) - Cn + n, \\
\mathcal{T}(n) &\leq Cn \log(n).
\end{aligned}
\tag{3.4}$$

Dostáváme tedy, že celková složitost algoritmu je $\mathcal{O}(n \log(n))$. Více v [22, 20, 21, 24, 30, 25, 16, 38].

3.5 Fortunův algoritmus

V literatuře se také můžeme setkat s názvy: *Plane sweep*, *Sweep line*, *Sweep algorithm*, *zametací metoda*. Po rovině se pohybuje *řídící přímka*, pohybuje se směrem shora dolů, jakmile narazí na nějaký bod z S tak ho zpracuje. Pro každý bod, který je nad ní vytvoří parabolu tak, že pro každý stav pohybující se přímky je daný bod ohniskem této paraboly. Jak se přímka pohybuje, tak se mění i parabola, protože její ohnisko se od přímky vzdaluje viz obr.3.10.



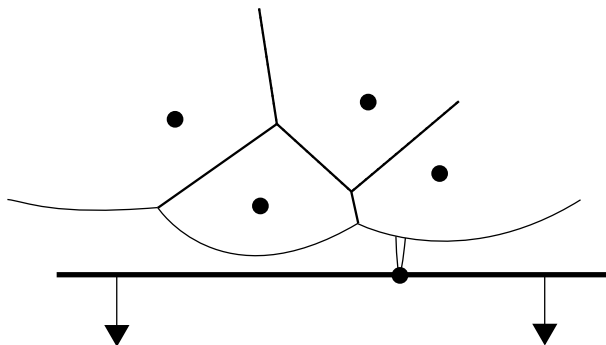
Obrázek 3.10: Znázornění postupu přímky (nejtlustší čára), která zametá jednotlivé body. Body, nad přímkou začínají vytvářet paraboly. Čárkovaně je znázorněn budoucí Voroného diagram.

Jak tedy *Fortunův algoritmus* funguje? Po rovině sjíždí řídící přímka směrem dolů, za „hotovou“ oblast nad přímkou považujeme tu část ze zameteného prostoru (prostor nad přímkou), která má blíže k již zametenému místu než k přímce samotné (v této části už je vše sestrojeno a nemůže se tedy nic změnit). Hranice hotové oblasti je tvořena body jejichž vzdálenost od daného generujícího bodu je stejná jako od řídící přímky a tvoří tedy paraboly. Hranice je tedy tvořena posloupností parabolických oblouků, kde oblouky, které se nacházejí na okraji „jdou“ do nekonečna. Tuto hranici budeme z literatury [7, 20] nazývat *pobřežní křivkou* (*beachline*).

Při tom, jak se řídící přímka v prostoru pohybuje směrem dolů, tak se mění i pobřežní křivka. V místech, kde se protnou jednotlivé parabolické oblouky, vznikají Voroného hrany, protože pro

každý takovýto průsečík platí, že jeho vzdálenost od zametací přímky je stejná jako od dvou různých generujících bodů.

Jakmile řídicí přímka narazí na nový generující bod z množiny S , vznikne nová parabola, tuto situaci nazýváme *místní událost* viz obr. 3.11, při tom jak se řídicí přímka pohybuje, tak tato nová parabola je na úplném začátku pouze polopřímka, poté se z ní stane parabolický oblouk jehož ramena se s pohybující se řídicí přímkou roztahují a tvoří novou zvětšující se Voroného hranu, která časem narazí na již vzniklé Voroného hrany.



Obrázek 3.11: Znázornění *místní události*.

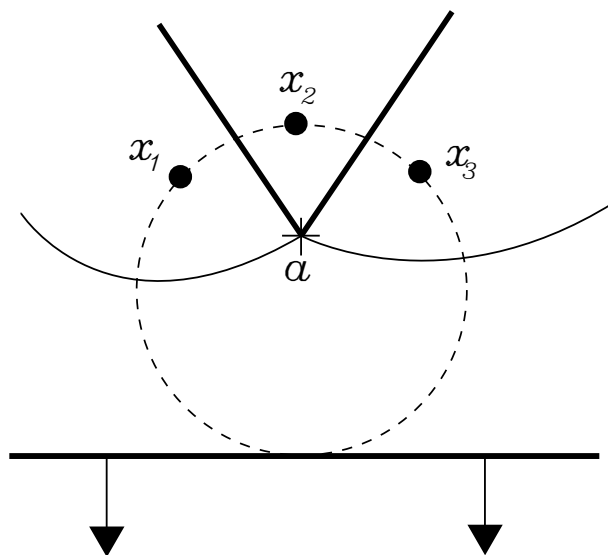
Může také nastat situace, že se nějaká parabola roztáhne natolik, že pohltí jiné paraboly a ty zcela zmizí z pobřežní křivky, to se stane v okamžiku, když se Voroného hrany setkají v jednom bodě. V takovémto případě s k tomuto budoucímu průsečíku blíží tři parabolické oblouky a jakmile se dostanou do bodu kde se mají protnout, prostřední parabola zanikne, v takovémto místě máme bod - Voroného vrchol, k němuž vedou tři Voroného hrany. Zároveň platí, že v tomto okamžiku je vzdálenost řídicí přímky stejná od tohoto Voroného vrcholu jako je vzdálenost tohoto vrcholu od jednoho z generujících bodů (všechny jsou vzdáleny stejně). Omezili jsme se na takový případ, že maximálně tři body leží na jedné kružnici. Jinými slovy tedy platí, že bod z řídicí přímky a všechny tři body leží na jedné kružnici, tomuto jevu říkáme *kružnicová událost* viz obr. 3.12.

Algoritmus si tedy bude držet v paměti nějaký seznam událostí a řídicí přímka bude přeskakovat mezi těmito událostmi, místní události bude mít již naplánované a kružnicové bude měnit průběžně podle změny pobřežní křivky.

Takovýto je princip algoritmu, nyní už stačí zvolit vhodné datové struktury, více k implementaci a Fortunovu algoritmu v [7, 20, 29, 21].

Věta 3.5.1. *Fortunův algoritmus má složitost $\mathcal{O}(n \log(n))$.*

Důkaz. Množina S obsahuje n generujících bodů a nastane tady n místních událostí a stejně tak i n kružnicových událostí, kružnicová událost smaže jednu parabolu z pobřežní křivky, ale ty přibývají pouze při místních událostech. Složitost vyhledání nebo nalezení prvku ve frontě (u nás seznamu) jsme z [22, 20] schopni najít v čase $\mathcal{O}(\log(n))$, jednu událost tedy zvládneme řešit se složitostí $\mathcal{O}(\log(n))$. V seznamu je celkem n prvků, tedy celková složitost Fortunova algoritmu je $\mathcal{O}(n \log(n))$. \square



Obrázek 3.12: Znázornění kružnicové události.

Kapitola 4

Implementace algoritmu

Úkolem bylo napsat program, který by byl schopen najít Voroného diagram pro zadanou diskrétní množinu bodů v euklidovské a manhattanské metrice. Veřejně přístupné kódy algoritmů, které jsou schopny sestavit Voroného diagram pracují s pokročilými programátorskými technikami a tudíž nejsou pro běžného uživatele moc „čitelné“.

V této práci jsem se rozhodl napsat kód za pomoci „jednoduchých“ programátorských technik. Tak, že pokud do kódu nahlédne i programátor amatér, tak bude schopen plně porozumět všem krokům. Myšlenka celého kódu je založena na poznacích z analytické geometrie (střední škola) a lineární algebry (základní kurzy vyšší matematiky) a celý algoritmus je tedy velmi intuitivně napsán.

Od toho se také odráží použití datové struktury v kódu, literatura [7, 20] nám radí použít *double connected edge list*. Já se rozhodl pro vlastní datovou strukturu a to takovou, že bod byl zapisován jako dvojice čísel, jedno pro x -ovou složku a druhý pro y -ovou složku. Úsečka je sedmice čísel uložená ve vektoru, první tři čísla jsou směrnice a posunutí přímky v obecném tvaru a další čtyři čísla jsou dvě dvojice okrajových bodů, které leží na této přímce, značící konec a začátek úsečky.

Co se týče algoritmu, tak jsem se rozhodl aplikovat definici Voroného diagramu 2.1.13, resp. implementovat průnik polorovin 3.2. Celý algoritmus byl napsán jazykem C++ 17.

4.1 Princip algoritmu

Vstup do programu byl `.txt` soubor se souřadnicemi jednotlivých generujících bodů. A jako výstup jsem zvolil `.svg` soubor, ve kterém byly souřadnice generujících bodů a souřadnice počátečních a koncových bodů jednotlivých Voroného hran. Nyní popíši princip jednotlivých funkcí a jejich aplikace. Funkce, které „nevracejí“ žádnou hodnotu tzn. začínají `void` se na svůj výstup odkazují pomocí ukazatelů.

```
void stredni (double x1, double x2, double y1, double y2, double *x, double *y)
```

Funkce najde pro dvojici vstupních bodů, jež jsou zadány pomocí x -ových a y -ových souřadnic, bod, který se nachází přesně mezi nimi, resp. bod, který je z pohledu euklidovské metriky k oběma nejbližší a zároveň je od obou stejně daleko. Jedná se o jednoduchou algebraickou operaci,

kde sečteme x -ové a y -ové souřadnice a podělíme je dvěma, tak dostaneme bod přesně uprostřed spojnice našich vstupních bodů.

```
double vzdalenosteuklid (double x1, double x2, double y1, double y2)
```

Funkce najde pro dvojici vstupních bodů, jež jsou zadány pomocí x -ových a y -ových souřadnic, jejich vzdálenost v euklidovské metrice. Jedná se o přepsání definice euklidovské vzdálenosti.

```
double vzdalenostmanhattan (double x1, double x2, double y1, double y2)
```

Funkce najde pro dvojici vstupních bodů, jež jsou zadány pomocí x -ových a y -ových souřadnic, jejich vzdálenost v manhattanské metrice. Jedná se o přepsání definice manhattanské vzdálenosti.

```
void prusecik (double a, double aa, double b, double bb, double c, double cc,
              double *xova, double *yova)
```

Funkce najde pro dvě trojice čísel, které představují koeficienty v obecné rovnici přímek, x -ovou a y -ovou souřadnici průsečíku těchto dvou přímek, pokud průsečík neexistuje, tak nevypíše nic resp. „průsečík neexistuje“. Abych se vyhnul nepřesnostem iterativních metod tak jsem zvolil metodu přímkou: *Gaussovu eliminační metodu*. Jde o přímý výpočet soustavy rovnic, která je zapsaná pomocí rozšířené matice 4.1, podle Gaussovy eliminační metody.

$$\left(\begin{array}{cc|c} a & b & -c \\ aa & bb & -cc \end{array} \right) \text{ a řešením soustavy je vektor } (x_1, x_2)^T. \quad (4.1)$$

Kde, a, b, c jsou koeficienty první přímky v obecném tvaru a aa, bb, cc jsou koeficienty druhé přímky v obecném tvaru.

```
void normalaeuklid (double x1, double x2, double y1, double y2)
```

Funkce najde pro dvojici vstupních bodů, jež jsou zadány pomocí x -ových a y -ových souřadnic, množinu všech bodů, které jsou od každého z těchto bodů stejně vzdáleny vzhledem k euklidovské metrice, tedy osu úsečky, která vznikne spojením těchto dvou bodů.

```
vector<vector<double>> normalamanhattan (double x1, double y1, double x2, double
y2)
```

Funkce najde pro dvojici vstupních bodů, jež jsou zadány pomocí x -ových a y -ových souřadnic, množinu bodů, které jsou stejně vzdáleny od obou vstupních bodů v manhattanské metrice. Výstupem funkce je vektor vektorů, kde každý vektor definuje úsečku nebo polopřímku ze kterých se množina stejně vzdálených bodů v manhattanské metrice skládá. Obecně může nastat několik případů:

1. jedna ze souřadnic, x -ová nebo y -ová, je stejná, potom je množina stejně vzdálených bodů v manhattanské metrice ekvivalentní množině stejně vzdálených bodů v euklidovské metrice, tedy jedná se o přímku (osu spojnice vstupních bodů), první případ na obrázku, viz obr. 4.1a (ještě může nastat situace, že souřadnice obou bodů prohodíme (zaměníme) a graf se nám tak otočí o devadesát stupňů).

2. x -ové i y -ové souřadnice se liší, a jeden bod má obě souřadnice větší než ten první, zároveň absolutní hodnota rozdílu x -ových souřadnic je větší/menší než absolutní hodnota rozdílu y -ových souřadnic, pak nastává druhý/třetí případ na obrázku, viz obr. 4.1b/4.1c.
3. x -ové i y -ové souřadnice se liší a jeden bod má větší x -ovou souřadnici, druhý bod má větší y -ovou souřadnici, zároveň absolutní hodnota rozdílu x -ových souřadnic je menší/větší než absolutní hodnota rozdílu y -ových souřadnic, pak nastává čtvrtý/pátý případ na obrázku, viz obr. 4.1d/4.1e.
4. Jako poslední případ může nastat ten, že oba body leží na nějaké přímce, jejíž normála bude spojnicí počátku $(0, 0)$ a $(1, 1)$ to znamená, že spojnice těchto bodů bude svírat s osou x a osou y úhel $\frac{\pi}{2}$ rad = 45° . V tomto případě nelze v konstrukci Voroného diagramu pokračovat, protože množina stejně vzdálených bodů od dvojice bodů je z části úsečka a z části oblast, která je navíc neomezená viz obr. 4.1f (ještě může nastat situace, kdy souřadnice obou bodů prohodíme a graf se nám tak otočí o devadesát stupňů, tedy oba body by ležely na přímce s normálou tvořenou spojnicí počátku $(0, 0)$ a $(1, -1)$ a jejich spojnice by svírala s osou x úhel $\frac{3\pi}{4}$ rad = 135°).

Zatímco v případech 4.1a a 4.1f je jasné jak vše vzniklo. V ostatních případech tedy nemáme jednu přímku, ale jednu úsečku a dvě polopřímky. Úsečka svírá s osou x úhel $\frac{\pi}{2}$ rad = 45° nebo $\frac{3\pi}{4}$ rad = 135° a polopřímky svírají s osou x úhel buď π rad = 90° nebo 2π rad = 180° . Pro každé dva body vytvoříme pomyslný obdélník (tečkované křivky), uvnitř tohoto obdélníku bude úsečka s normálou tvořenou spojnicí počátku $(0, 0)$ a $(1, 1)$ nebo tvořenou spojnicí počátku $(0, 0)$ a $(-1, 1)$, podle toho jaký případ nastane a tam kde se tato úsečka protne s tečkovaným obdélníkem sestrojíme polopřímky, které budou kolmé na tu stranu tečkovaného obdélníku, kde se úsečka s obdélníkem protla viz obr. 4.1.

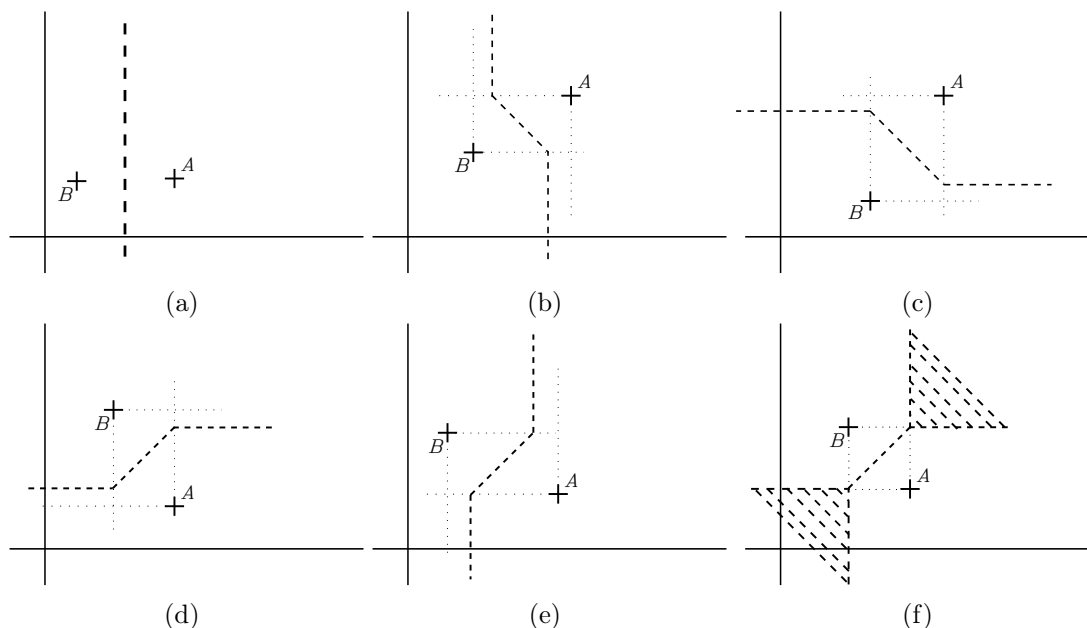
```
void nejblizeeuclid (vector<vector<double>>body, double x, double y, double *nejx,
double *nejy)
```

```
void nejblizemanhattan (vector<vector<double>>body, double x, double y, double *
nejx, double *nejy)
```

Funkce najde pro zadaný bod, který je zadán pomocí x -ových a y -ových souřadnic, bod z nějaké vstupní množiny bodů (`vector<vector<double>>body`), který je k tomuto zadanému bodu nejbliže vzhledem k euklidovské metrice.

Zvolil jsem přístup, že jsem „procházel“ vstupní vektor bodů a kontroloval jestli neměřím vzdálenost mezi dvěma totožnými body (kdyby se původní bod nacházel v zadané množině bodů), pokud ano, tak jsem do pole vzdáleností načetl vysoké číslo (10000) (předpokládal jsem že 10000 nebude nejmenší vzdálenost dvou prvků v zadané množině), pokud body nebyly totožné, tak jsem do pole vzdáleností načetl jejich euklidovskou/manhattanskou vzdálenost. Poté jsem použil funkci na vyhledávání na najetí pozice nejmenšího elementu v poli a našel hodnotu tohoto prvku v původní množině bodů.

Alternativní myšlenka by byla použít libovolný řadící algoritmus např: *Bubblesort*, *Mergesort*, *Insertsort*, *Quicksort*,... a po najetí nejmenšího prvku ho zpětně dohledat v původním vektoru bodů, více o řadících algoritmech v [21].



Obrázek 4.1: Všechny případy při konstrukci množiny stejně vzdálených bodů od dvou generujících bodů v manhattanské metrice. Tečkované čáry slouží jako pomocné, A a B jsou zadané body a čárkovaná křivka je znázornění množiny stejně vzdálených bodů od bodů A a B .

```
double overenipruseciku (double x1, double y1, double x2, double y2, double prusx,
double prusy, double xx1, double yy1, double xx2, double yy2)
```

Funkce ověří jestli zadaný bod (průsečík) leží na dvou úsečkách zároveň. Úsečky jsou zadány počátečními a koncovými body.

Funkce předpokládá, že uživatel zadal opravdový průsečík tedy bod, který leží někde na průsečíku přímek, které leží na zadaných úsečkách. Funkce se v rámci každé úsečky, průsečíku „ptá“, jestli je jeho x -ová souřadnice větší než jeden okrajový bod a menší než druhý okrajový bod, to samé provádí i pro y -ovou složku a celý tento postup je potřeba udělat pro všechny konfigurace bodů, protože nevíme, který z nich je větší nebo menší. Výstupem z programu je při kladné odpovědi jednička a při negativní nula.

```
vector<vector<double>> prunikprojedeneuklid (vector <vector <double>> body,
double kolikaty)
```

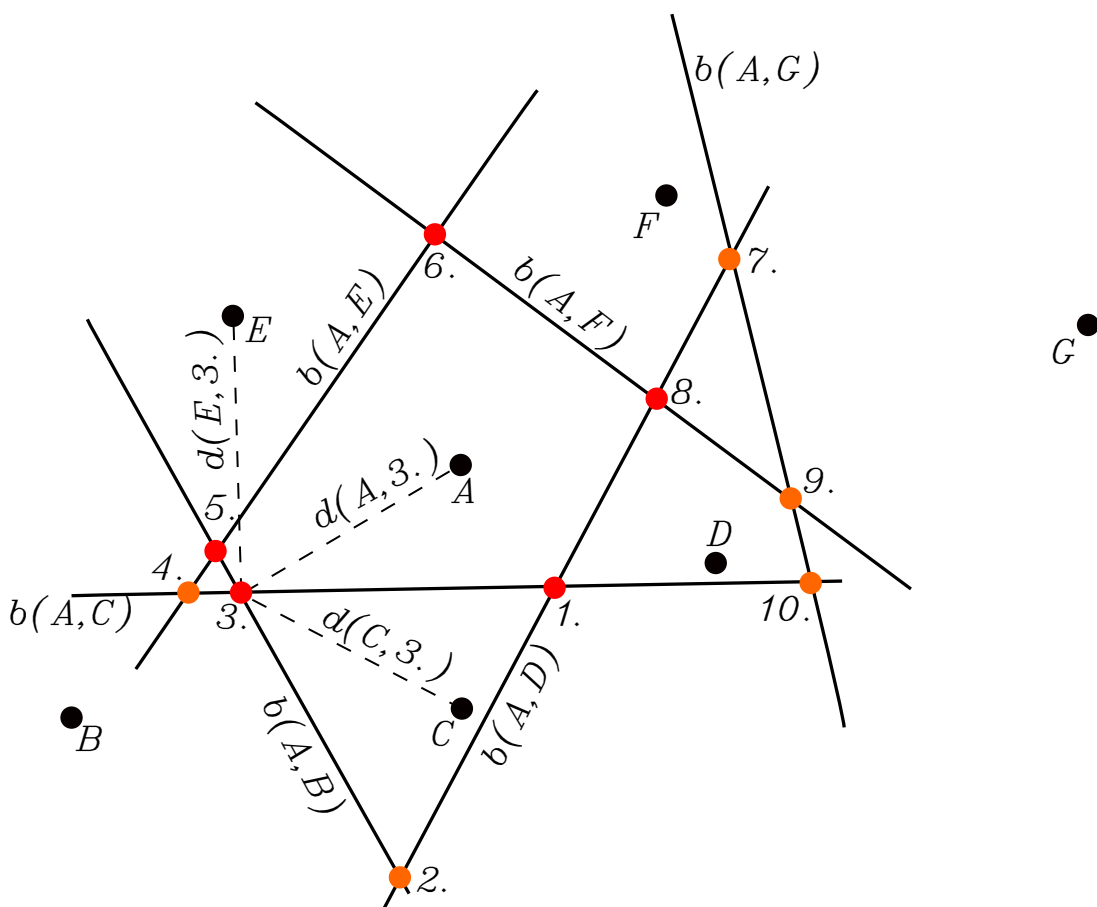
```
vector<vector<double>> prunikprojedenmanhattan (vector <vector <double>> body,
double kolikaty)
```

Funkce vytvoří kompletní Voroného region pro konkrétní bod ze zadané množiny generujících bodů vzhledem k euklidovské nebo manhattanské metrice.

Funkce tedy dostane množinu generujících bodů (`vector<vector<double>> body`) a pořadí jednoho (`double kolikaty`) bodu (*hlavního bodu*), pro který chceme napočítat Voroného region. Prvně si do proměnných načteme souřadnice hlavního bodu, pro který hledáme Voroného region.

Poté „projdeme“ množinu generujících bodů a pro všechny body které se neshodují v obou souřadnicích s naším hlavním bodem, najdeme množinu bodů z \mathbb{R}^2 , které jsou stejně daleko jak od aktuálního, tak od hlavního bodu vzhledem k dané metrice, jinými slovy najdeme všechny bisektory našeho hlavního bodu s ostatními body z generující množiny. Pro tuto akci vytvoříme více rozměrný vektor všech bisektorů hlavního bodu s ostatními body z generující množiny.

V dalším kroku najdeme všechny průsečíky všech těchto množin (u euklidovské metriky to jsou průsečíky obyčejných přímek u manhattanské průsečíky úseček a dvou polopřímek), pro tyto průsečíky vytvoříme další vektor do kterého je budeme načítat. Zároveň do tohoto vektoru načteme umělá kladná a záporná „nekonečna“, protože jak již víme, tak pokud by náš hlavní bod ležel na konvexním obalu generující množiny tak jeho Voroného region nebude uzavřený viz věta 2.1.15 a dvě jeho Voroného hrany „půjdou“ do nekonečna.



Obrázek 4.2: Znárodnění nalezení Voroného regionu k hlavnímu bodu (bod A). Porovnáváním vzdálenosti $d(A, 1.)$ jednotlivých průsečíků (čísla 1, 2, 3, ...) s hlavním bodem A a ostatními body z generující množiny všechny písmena až na A . $b(A, B)$ znázorňuje bisektor příslušný k generujícím bodům A a B . Červené průsečíky se podílí na tvorbě Voroného regionu který je generován bodem A , zatímco oranžové průsečíky se na tvorbě Voroného regionu nepodílí. Pokud je vzdálenost $d(C, 3.)$ stejná jako $d(A, 3.)$, pak průsečík 3. je vrchol Voroného regionu, který je generován hlavním bodem A .

Nyní projdeme náš vektor s průsečíky a zjistíme, které z nich tvoří Voroného region příslušný k hlavnímu bodu viz obr. 4.2. Najdeme vzdálenost daného průsečíku s hlavním bodem a poté najdeme k tomuto průsečíku nejbližší bod z generující množiny a zjistíme vzdálenost tohoto bodu s průsečíkem a ptáme se jestli je rozdíl těchto vzdáleností nulový, pokud ano, tak je jasné že tento bod tvoří Voroného region příslušný k hlavnímu bodu. Tento postup aplikujeme na všechny prvky z vektoru průsečíků. Průsečíky, které „projdou“ touto procedurou načteme do dalšího vektoru okrajových bodů.

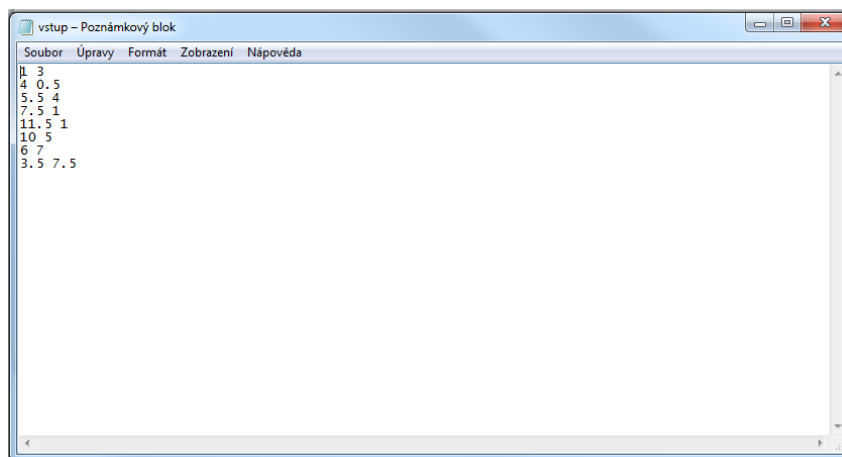
Teď najdeme dvojice z okrajových bodů a zjistíme, které z nich leží na stejné přímce z vektoru příemek a poskládáme je do dvojic a výstup z této funkce bude vektor dvojic těchto bodů, které budou tvořit úsečky daného Voroného regionu.

Stejný postup aplikujeme i na manhattanskou metriku, kde se bude jenom lehce lišit provedení nějakých operací, ale myšlenka zůstává stejná.

Nyní máme všechny funkce, které budeme potřebovat ke konstrukci kompletního Voroného diagramu, v `mainu` načteme zadané hodnoty z `.txt` souboru a vytvoříme z nich vektor, který budeme pomocí cyklů procházet a pro každý bod z tohoto vektoru vytvoříme Voroného region. Pro každý bod tedy dostaneme okrajové body úseček, které tvoří příslušný Voroného region, x -ové a y -ové souřadnice budeme skládat do posledního výstupního vektoru, některé hrany se nám tam načtou víckrát, to ovšem ničemu nevádí ev. je můžeme najít a vymazat z vektoru. Tento výsledný vektor poté načteme do předem připravené šablony, kterou potom vypíšeme do výstupního `.svg` souboru, který můžeme otevřít buď v prohlížeči nebo v aplikaci k tomu určené, já zvolil *Inkscape*.

4.2 Ukázka algoritmu

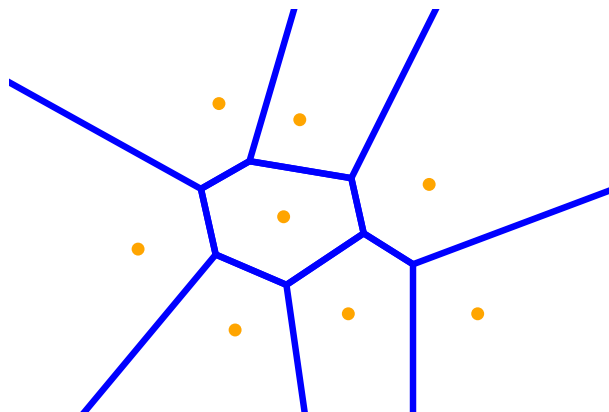
Jako názorný příklad na funkčnost jsem zvolil libovolnou jednoduchou množinu bodů viz obr. 4.3. Podíváme se jak bude vypadat výstup z programu pro obě metriky se stejným vstupním souborem bodů.



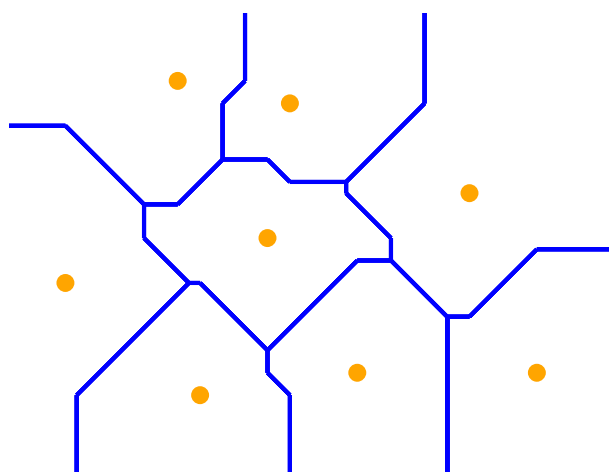
```
1 3
4 0.5
5 5 4
7.5 1
11.5 1
10 5
6 7
3.5 7.5
```

Obrázek 4.3: Vstup do programu ve formě `.txt` souboru.

Jako výstup programu jsem, jak již bylo zmíněno, zvolil `.svg` soubor. Výstup z programu pro euklidovskou a manhattanskou metriku viz obr. 4.4 a 4.5.



Obrázek 4.4: Výstup z programu ve formě `.svg` souboru pro euklidovskou metriku.



Obrázek 4.5: Výstup z programu ve formě `.svg` souboru pro manhattanskou metriku.

Kapitola 5

Delonovské množiny, teorie čísel a algebra

5.1 Úvod do teorie čísel a algebry

5.1.1 Algebraická čísla

Definice 5.1.1. *Monickým polynomem* nazveme polynom s jednou proměnnou, jehož koeficient u členu s nejvyšší mocninou, je roven jedné.

Definice 5.1.2. Řekněme, že číslo $x \in \mathbb{C}$ je *algebraickým číslem*, pokud existuje monický polynom $f \in \mathbb{Q}[X]$ tak, že $f(x) = 0$. Množinu všech algebraických čísel značíme \mathbb{A} .

Definice 5.1.3. Necht' $x \in \mathbb{A}$. Monický polynom $f \in \mathbb{Q}[X]$ nejnižšího stupně, pro který platí, že $f(x) = 0$, nazýváme *minimální polynom* algebraického čísla x . Ostatní kořeny nazýváme *algebraické sdružené*. *Stupeň* algebraického čísla definujeme jako stupeň polynomu f .

Definice 5.1.4. Číslo nazveme *algebraické celé číslo*, pokud existuje monický polynom $g \in \mathbb{Z}[X]$ tak, že $g(x) = 0$. Množinu všech algebraických celých čísel značíme \mathbb{B} .

5.1.2 Pisotova čísla

Speciálními případy algebraických celých čísel jsou tzv. *Pisotova čísla*.

Definice 5.1.5. Necht' α je reálné algebraické celé číslo takové, že $\alpha > 1$. Pokud jsou všechny algebraicky sdružené kořeny v absolutní hodnotě menší než jedna, pak číslo α nazveme *Pisotovo číslo*.

Definice 5.1.6. Necht' $\alpha \in \mathbb{C} \setminus \mathbb{R}$ algebraické celé číslo takové, že $|\alpha| > 1$. Pokud všechny algebraicky sdružené kořeny algebraického čísla α až na komplexně sdružené číslo $\bar{\alpha}$, jsou v absolutní hodnotě menší než jedna, pak číslo α nazveme *komplexní Pisotovo číslo*.

5.1.3 Řešení rovnice $X^3 + X^2 + X - 1 = 0$

V rámci této práce se budeme zabývat pouze rovnicí:

$$X^3 + X^2 + X - 1 = 0. \quad (5.1)$$

Rovnice 5.1 je kubická a bude mít tři kořeny z toho jeden bude reálný a dva komplexní:

$$X_1 \in \mathbb{R}, \quad X_1 \approx 0,54369, \quad (5.2)$$

$$X_2 \in \mathbb{C}, \quad X_2 \approx -0,77184 - 1,115i, \quad (5.3)$$

$$X_3 \in \mathbb{C}, \quad X_3 \approx -0,77184 + 1,115i. \quad (5.4)$$

Vidíme, že kořeny X_2 a X_3 jsou komplexní a kořen X_1 je reálný. Z definice komplexních Pisotových čísel (definice 5.1.6), vidíme, že kořeny X_2 a X_3 jsou komplexní Pisotova čísla. Nám bude stačit pokud bude dále pracovat pouze s jedním z nich, volme BÚNO X_3 jako náš *výchozí kořen*, který se nám bude za chvíli hodit.

5.2 Úvod do Delonovských množin

V této kapitole si zadefinujeme základní pojmy z diskrétní geometrie včetně toho co je Delonovská množina.

5.2.1 Základní pojmy a definice

Definice 5.2.1. Množinu bodů $A \subset \mathbb{R}^n$ nazveme *diskrétní*, pokud $\forall x \in A$ existuje otevřené okolí $U_x \subset \mathbb{R}^n$ tak, že platí $U_x \cap A = \{x\}$.

To znamená, že pro každý bod $x \in A$ existuje $r > 0$ tak, že $B_r(x) \cap A$ obsahuje pouze jediný bod x .

Definice 5.2.2. Diskrétní množinu $A \subset \mathbb{R}^n$ nazýváme *stejněměrně diskrétní*, pokud existuje poloměr $r > 0$ takový, že pro všechna $x, y \in A$ platí, že $B_r(x) \cap B_r(y) = \emptyset$.

Definice 5.2.3. Diskrétní množinu $A \subset \mathbb{R}^n$ nazveme *relativně hustou*, pokud $(\forall x \in \mathbb{R}^n)(\exists r > 0)(\forall y \in \mathbb{R}^n)(B_r^*(x) \cap A \neq \emptyset)$.

Definice 5.2.4. Necht' $A \subset \mathbb{R}^n$ je diskrétní množina bodů. Tuto množinu A nazýváme *Delonova množina* pokud je relativně hustá a zároveň rovnoměrně diskrétní.

Ukázka jednoduchých Delonovských množin viz obr. 5.1.

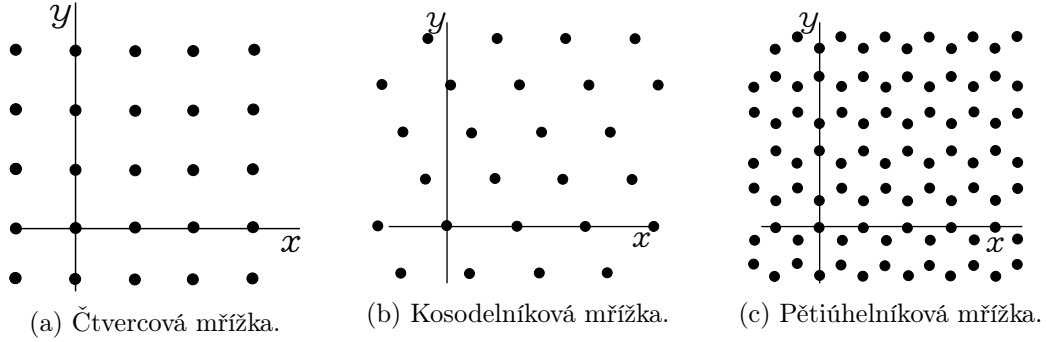
5.2.2 Množina $X^m(\beta)$

Nyní si definujeme množinu, se kterou budeme dále pracovat a budeme zkoumat její dláždění pomocí Voroného diagramu.

Definice 5.2.5. Necht' $\beta \in \mathbb{C} \setminus \mathbb{R}$, $\beta > 1$ a dále $n, m \in \mathbb{N}$, pak množinu $X^m(\beta)$ definujeme:

$$X^m(\beta) := \left\{ \sum_{j=0}^n a_j \beta^j \mid \forall k \in \hat{n}, a_k \in \{0, 1, \dots, m\} \right\}.$$

Kde takovým to zápisem máme na myli, že pro dané β splňující předpoklady a pro dané n, m a a_k z dané rodiny $\{0, 1, \dots, m\}$, bereme za a_j v sumě všechny kombinace z rodiny $\{0, 1, \dots, m\}$.



Obrázek 5.1: Ukázka jednoduchých Delonovských množin-mřížek.

Definice 5.2.6. Necht $\beta, x \in \mathbb{R}$, $x \geq 0$, $|\beta| > 1$ Necht $m \in \mathbb{N} \mid m > 0$ tak, že $\beta^{-m-1}x \in [0, 1)$. Pak můžeme číslo x vyjádřit jako:

$$x = x_m \beta^m + \dots + x_1 \beta + x_0 + \frac{x_{-1}}{\beta} + \frac{x_{-2}}{\beta^2} + \dots$$

Toto vyjádření nazýváme β -rozvojem. Navíc pokud existuje $k \in \mathbb{N}$ tak, že $x_i = 0$ pro $i < -k$, pak říkáme, že takovýto β -rozvoj je *konečný*.

Poznámka. Jak již bylo zmíněno v celé práci budeme uvažovat pouze rovnici 5.1, která má dva komplexní a jeden reálný kořen. Pojmem reálný doplněk v odstavci za definicí 5.2.8 máme na mysli právě tento jeden reálný kořen z rovnice 5.1.

Definice 5.2.7. Necht \mathbb{L} je neprázdna množina s operacemi *sčítání*(\oplus) a *násobení*(\otimes), která splňuje následující podmínky pro každé $a, b, c \in \mathbb{L}$:

1. $a \oplus b \in \mathbb{L}$.
2. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$.
3. $a \oplus b = b \oplus a$.
4. Existuje prvek $0_L \in \mathbb{L}$ tak, že $a \oplus 0_L = a$.
5. Rovnice $a \oplus x = 0_L$ má řešení v \mathbb{L} .
6. $a \otimes b \in \mathbb{L}$.
7. $a \otimes (b \otimes c) = (a \otimes b) \otimes c$.
8. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.

Pak množinu \mathbb{L} nazveme *okruhem*.

Definice 5.2.8. Necht $\text{Fin}(\beta)$ je množina všech nezáporných komplexních čísel s konečným β -rozvojem. Označme $\mathbb{Z}[\beta]$ nejmenší okruh obsahující \mathbb{Z} a β . Dále označme $\mathbb{Z}[\beta]_{\geq 0}$ nezáporné prvky $\mathbb{Z}[\beta]$. Říkáme, že číslo β splňuje tzv. *podmínku konečnosti* nebo také *Vlastnost (F)* pokud platí:

$$\text{Fin}(\beta) = \mathbb{Z}[1/\beta]_{\geq 0}.$$

Nás bude zajímat případ, kdy bude β v definici 5.2.5 právě komplexní Pisotovým číslem, které splňuje předpoklady definice. Dále budeme chtít aby pro reálný doplněk našeho komplexního Pisotova čísla platilo, že $1/\tilde{\beta}$ splňuje *Vlastnost (F)* (definice 5.2.8), kde $\tilde{\beta}$ je reálný doplněk komplexního Pisotova čísla.

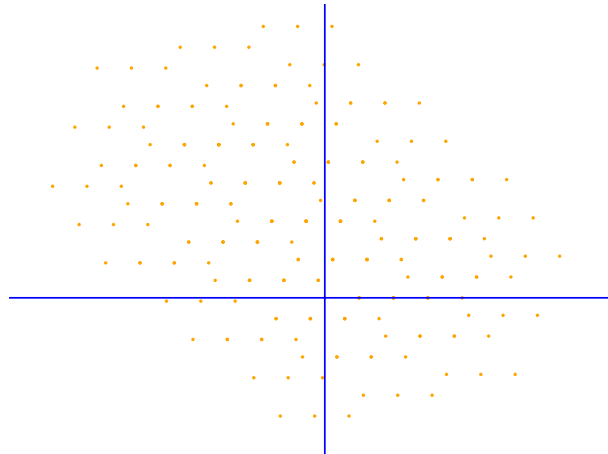
Dále budeme chtít, aby platila nerovnost $m > |\beta|^2 - 1$. Poté z [41] víme, že množina $X^m(\beta)$ je stejnoměrně diskrétní a relativně řídká tzn. jedná se od Delonovskou množinu.

5.2.3 Konstrukce naší množiny

Nyní sestavíme konkrétní množinu podle definice 5.2.5, kde za β budeme „brát“ kořen X_3 z rovnice 5.1. Dále budeme uvažovat $m = 2$ a $n = 3$. Jednotlivé výsledky v sumě z definice 5.2.5 budou komplexní čísla, které můžeme reprezentovat jako body v Gaussově rovině. Abychom mohli sestavit Voroného diagram pro takovou množinu, použijeme transformaci $f : \mathbb{C} \rightarrow \mathbb{R}^2$. Již víme, že \mathbb{R}^2 je metrický prostor a můžeme na něm sestavit Voroného diagram. Budeme tedy hledat množinu $X^2(X_3)$ definovanou jako:

$$X^2(X_3) := \left\{ \sum_{j=0}^4 a_j X_3^j \mid \forall k \in \hat{\mathbb{Z}}, a_k \in \{0, 1, 2\} \right\}.$$

Tuto množinu nazveme jako *naše množina*. K nalezení naší množiny si napíšeme jednoduchý program, který vypočítá všechny kombinace koeficientů a každou takovou kombinaci přenásobí příslušnou mocninou našeho β a následně sečte. Tím vznikne pro $m = 2$ (tedy abeceda koeficientů bude mít 3 prvky) a $n = 4$ (tedy suma provede 5 iterací pro každou kombinaci koeficientů) množina, která bude mít $3^5 = 243$ prvků. Pro náš konkrétní příklad víme že tato množina bude Delonovská a její znázornění je na obrázku 5.2.



Obrázek 5.2: Výstup z programu pro napočítání Deloneho množiny pro $m = 2$, $n = 4$ a kořen x_3 .

Kapitola 6

Voroného diagram pro naši množinu

V této kapitole si vyzkoušíme náš program na napočítání Voroného diagramu v euklidovské a manhattanské metrice na naší množině, kterou jsme zkonstruovali v předchozí kapitole.

Z [41] víme, že jedna z vlastností naší množiny $X^2(X_3)$ je, že pro každý bod x z této množiny existuje konečný poloměr koule, který nám říká, jaké další body z této množiny ovlivňují tvar Voroného regionu, který je generován bodem x . Na otázku jak tento poloměr vypočítat nám odpoví následující věta.

Věta 6.0.1. *Nechť $\Omega = [0; c)$ je interval. A Nechť p je první kladné celé číslo takové, že $\Im(\beta^p)$ a $\Im(\beta)$ mají opačné znaménko a dále nechť k je nejmenší celé číslo, pro které platí $(\beta)^k < c/2$. Pak*

$$L := |\beta|^k \max_{i,j \in \{0,p-1,p\}, i < j} \left| \frac{\beta^{i+j}(\beta^i - \beta^j)}{\Im(\beta^i \beta^{-j})} \right|, \quad (6.1)$$

kde $c = \frac{m}{1-\beta^l}$, $\beta^l = \Re(\beta)$, $m = 2$ a zároveň platí, že $\Delta(\text{VR}(y)) \leq L, \forall y \in X^m(\beta)$, kde $\Delta(\text{VR}(y))$ je nejmenší poloměr $D > 0$ takový, že $\text{VR}(y) \subset B(y, D/2)$.

Z této věty jsme schopni napočítat poloměr L a tedy zjistit, které body z naší množiny budou mít vliv na Voroného region jednoho konkrétního bodu. V [41] je tato věta dokázána ve vší obecnosti pro euklidovskou metriku, musíme okomentovat jak a jestli vůbec tato věta platí pro manhattanskou metriku. Důkaz prvně ukazuje platnost pro bod $y = 0$ tzn. $[0; 0]$.

Vzhledem k tomu, že čísla p, k ve větě jsou volena tak, aby projekce bodů $x_1 := \beta^k, x_2 := \beta^{k+p-1}, x_3 := \beta^{k+p}$ do \mathbb{R}^2 zajistila, že bod $[0; 0]$ bude uvnitř trojúhelníku, který vznikne spojením těchto bodů a že pokud budeme generovat Voroného region kolem bodu $[0; 0]$, tak že tento region v euklidovské metrice bude omezená množina. Bohužel to stejné se nedá říci o manhattanské metrice v naší množině, kde bychom vhodnou konfiguraci tří bodů možná ani nenašli. Naštěstí nemusíme platnost věty ukazovat obecně, ale pouze pro naši množinu. Proto nevolíme tři body, ale čtyři a to tak, že $x_1 := \beta^1, x_2 := \beta^2, x_3 := \beta^3, x_4 := \beta^4$ viz obr. 6.1. Tato konfigurace bodů nám zajistí, že daný Voroného region kolem bodu $[0; 0]$ bude omezený.

Nechť ρ je nejmenší poloměr koule se středem v $[0, 0]$ takový, že $\text{VR}(y)$ bude podmnožinou této koule. Z toho jak jsou definované $\Delta(\text{VR}(y))$ a $\text{VR}(y)$ vidíme, že platí $\Delta(\text{VR}(0)) < 2\rho$. Pokusíme se najít nejvzdálenější vrchol našeho Voroného regionu tak, že manhattanská vzdálenost tohoto vrcholu od $y = 0$ bude rovna $\rho = L/2$. Vzhledem k tomu, že víme jak vypadá naše množina a jsme schopni napočítat i konkrétní Voroného region generovaný bodem $[0; 0]$ a tvořen body

x_1, x_2, x_3, x_4 , jsme schopni sestavit rovnice ve kterých budeme znát znaménka jednotlivých členů viz obr. 6.1. Dostaneme soustavu rovnic:

$$|x - x_4| = |x - x_2| = |x - 0|, \quad (6.2)$$

po rozepsání do jednotlivých souřadnic dostáváme

$$\begin{aligned} |x_x - x_{3x}| + |x_y - x_{3y}| &= |x_x| + |x_y| \\ |x_x - x_{2x}| + |x_y - x_{2y}| &= |x_x| + |x_y|. \end{aligned}$$

Z obrázku 6.1 vidíme, kde (v jaké části roviny (kvadrantu)) se nachází náš bod $[x_x; x_y]$ je to nejvzdálenější bod Voroného regionu $VR(A)$ od bodu A v manhattanské metrice. Díky této informaci znázorněné na obr. 6.1 známe znaménka a budeme schopni soustavu vyřešit a najít tento bod přesně.

$$\begin{aligned} x_x - x_{3x} - x_y + x_{3y} &= -x_x - x_y \\ -x_x + x_{2x} + x_y - x_{2y} &= -x_x - x_y. \end{aligned}$$

Po vyjádření x_x a x_y dostáváme

$$\begin{aligned} x_x &= \frac{x_{3x} - x_{3y}}{2}, \\ x_y &= \frac{x_{2y} - x_{2x}}{2}. \end{aligned}$$

Po dosazení konkrétních hodnot jednotlivých bodů dostáváme, že $x_x = -2,3869$ a $x_y = -0,53681$ a je to bod, který je průsečík $VD(A)$ a k na obr. 6.1. Manhattanská vzdálenost tohoto bodu od A bude tedy $L/2 = \rho = 2,92371 \rightarrow L = 5,84742$. Z [41] víme, že pro euklidovskou metriku bude $L = 2,546$. Zároveň z [41] víme, že toto L platí i pro všechny body a nejenom pro $[0; 0]$.

Nyní tedy známe konkrétní hodnotu L pro obě metriky naší množiny. Abychom mohli určit kolik typů Voroného regionů se vyskytuje v naší množině s danou metriku, najdeme všechny možné L -plochy resp. lokální konfigurace kolem bodů z naší množiny do vzdálenosti L .

Definice 6.0.1. L -plochu bodu $x \in X^2(X_3)$ definujeme jako množinu

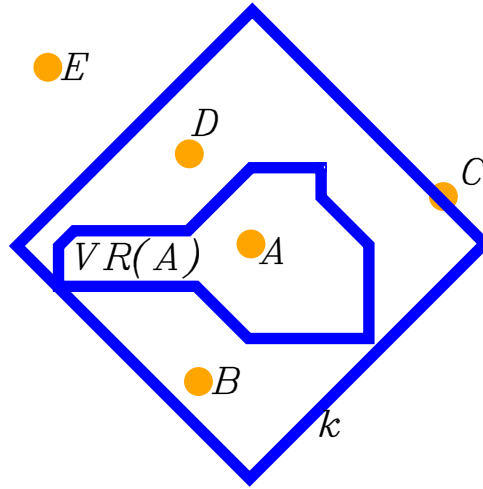
$$\mathcal{P}_L(x) := (X^2(X_3) \cap B(x, L)) - x. \quad (6.3)$$

Dále jako důsledek věty 5.2 z [41] máme, že pokud mají dva body z naší množiny stejnou L -plochu, pak jsou stejné i jejich Voroného regiony. Nyní vyslovíme větu, která nám nepřimo odpoví na otázku, kolik typů Voroného regionů se vyskytuje v naší množině pro každou metriku.

Věta 6.0.2. Necht $x, y \in X^2(X_3)$ a $L > 0$. Pokud $\mathcal{P}_L(x) \neq \mathcal{P}_L(y)$, pak existuje ξ z konečné podmnožiny $\Xi \subset [0; c]$:

$$\Xi := \{z' : z \in \mathcal{P}_L(0)\} \cup \{c - z' : z \in \mathcal{P}_L(0)\}, \quad (6.4)$$

tak, že ξ se nachází mezi x' a y' resp. $\min\{x'; y'\} < \xi \leq \max\{x'; y'\}$.



Obrázek 6.1: Důkaz věty 6.0.1. Kde $A = y = [0; 0]$, $B = x_2 = \beta^2 = [-0, 6478; -1, 7214]$, $C = x_3 = \beta^3 = [2, 4196; 0, 6062]$, $D = x_1 = \beta = [-0, 7718; 1, 1151]$, $E = x_4 = \beta^4 = [-2, 543; 2, 230]$. A k je koule s poloměrem ρ v manhattanské metrice.

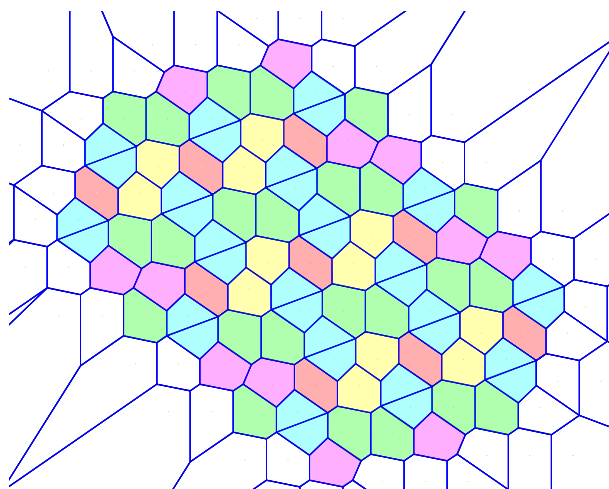
Důsledek věty 6.0.2 a definice 6.0.1 je ten, že najdeme množinu, ve které jsou všechny body z naší množiny a zároveň body, které jsou uvnitř koule s poloměrem L posunuté k počátku (bod $[0; 0]$). Poté najdeme množinu Ξ , která je sjednocení bodů tvořené reálnou částí kořene rovnice 5.2 z množiny $\mathcal{P}_L(0)$ a těchto bodů posunutých o c . Tím rozdělíme interval $[0; c)$ na pod intervaly jejichž hranice budou jednotlivé body ξ . Maximální počet typů Voroného regionů bude $|\Xi| + 1$. Díky implikaci ve větě 6.0.2 nemáme zaručeno, že v každém podintervalu $[\xi_i; \xi_{i+1}) \subset \Xi$ bude jiný typ Voroného regionu (mohou se nacházet dva stejné v sousedních podintervalech). Proto každý podinterval překontrolujeme a odstraníme ty regiony, které se vyskytují víckrát.

6.1 Voroného diagram naší množiny v euklidovské metrice

Při generování naší množiny jsem použil dva výstupy z programu, první z nich byl `.txt` soubor se dvěma sloupečky, kde první z nich obsahoval x -ové souřadnice a druhý y -ové souřadnice. Jako další výstup jsem použil `.svg` soubor, který mi rovnou vykreslil rozložení bodů v rovině (\mathbb{R}^2) . První z těchto výstupů (`.txt` soubor) jsem použil jako vstup do programu, který napočítává Voroného diagram vzhledem k euklidovské metrice pro zadanou množinu bodů v rovině (\mathbb{R}^2) . Vzniklé dláždění prostoru můžeme vidět na obrázku 6.2.

Z obrázku 6.2 si můžeme všimnout, že se v dláždění stále opakují 4 různé typy tvarů Voroného regionů, které jsou pouze pootočené (barvy, modrá, zelená, žlutá a červená). Mohlo by se zdát, že se zde opakuje ještě jeden tvar (fialová barva), tento tvar se vyskytuje pouze na „okrajích“ protože zde ještě nejsou vygenerovány všechny body, kdybychom generovali větší část naší množiny tyto fialové regiony by se „proměnily“ na regiony modré barvy.

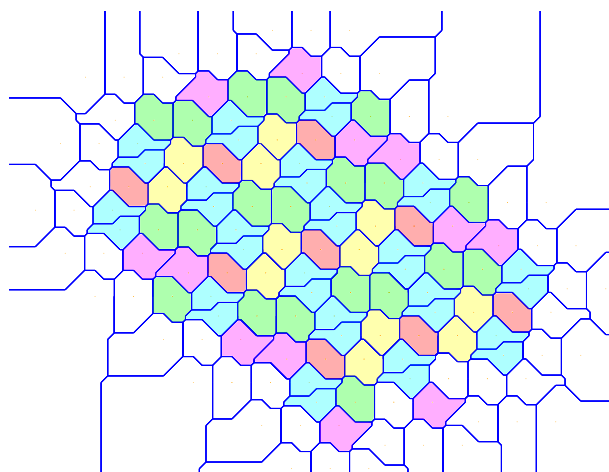
Ke stejnému závěru jsme došli po zkonstruování množiny Ξ a následném sjednocení podintervalů, které zde byly navíc. Mohutnost množiny $|\Xi| = 3$ tzn. celkový počet typů Voroného regionů v naší množině s euklidovskou metrikou je $3 + 1 = 4$.



Obrázek 6.2: Část Voroného diagramu Deloneho množiny $X^2(X_3)$. Čtyři opakující se pootočené Voroného regiony, které jsou rozlišeny různými barvami vzhledem k euklidovské metrice.

6.2 Voroného diagram naší množiny v manhattanské metrice

Při konstrukci Voroného diagramu v manhattanské metrice pro naši množinu, byl použit stejný postup jako v předchozím případě (pro euklidovskou metricku). Opět si můžeme všimnout, že se v dláždění opakuje 5 stejných pouze pootočených Voroného regionů.



Obrázek 6.3: Část Voroného diagramu Deloneho množiny $X^2(X_3)$. Čtyři opakující se pootočené Voroného regiony, které jsou rozlišeny různými barvami vzhledem manhattanské metrice.

Z obrázku 6.3 si můžeme všimnout, že se v dláždění stále opakují 4 různé typy tvarů Voroného regionů, které jsou pouze pootočené (barvy, modrá, zelená, žlutá a červená). Mohlo by se zdát, že se zde opakuje ještě jeden tvar (fialová barva), tento tvar se vyskytuje pouze na „okrajích“ protože zde ještě nejsou vygenerovány všechny body, kdybychom generovali větší část naší množiny tyto fialové regiony by se „proměnily“ na regiony modré barvy.

Ke stejnému závěru jsme došli po zkonstruování množiny Ξ a následném sjednocení podintervalů, které zde byly navíc. Mohutnost množiny $|\Xi| = 3$ tzn. celkový počet typů Voroného regionů v naší množině s manhattanskou metrikou je $3 + 1 = 4$.

Závěr

Cílem této práce bylo nastudovat teorii k Voroného diagramům v euklidovské a manhattanské metrice. Provést analýzu složitosti k různým algoritmům pomocí kterých je možné Voroného diagramy konstruovat. Implementovat vlastní algoritmus, pomocí kterého by bylo možné hledat Voroného diagramy pro libovolnou množinu zadaných bodů. A nastudovat teorii potřebnou k Delonovským množinám a Pisotovým číslům a vykonstruovat Delonovskou množinu jejíž generátor je komplexní Pisotovo číslo příslušné rovnici $X^3 + X^2 + X - 1 = 0$ a následně sestrojít Voroného diagram v manhattanské a euklidovské metrice příslušný k této množině.

Úvodní kapitola stručně vysvětluje „co je Voroného diagram“, stručnou historii a jeho aplikace napříč vědními obory jako jsou informatika, biologie, krystalografie, molekulární chemie a geometrické problémy.

Další kapitola je věnována základní teorii a vlastnostem Voroného diagramů. Dále se zde věnujeme různým zobecněním Voroného diagramů např. Voroného diagram v jiných metrikách, kružnic, úseček, . . .

V třetí kapitole se věnujeme úvodu do složitosti algoritmů a popisu principu některých algoritmů vhodných ke konstrukci Voroného diagramů jako jsou průnik polorovin, inkrementální konstrukce, rozděl a panuj a Fortunův algoritmus.

V následující kapitole je popsána vlastní implementace algoritmu, který slouží ke konstrukci Voroného diagramu v euklidovské a manhattanské metrice.

Pátá kapitola obsahuje teorii k algebraickým a Pisotovým číslům spolu s řešením rovnice $X^3 + X^2 + X - 1 = 0$. Dále se zde věnujeme Delonovským množinám a napočítáním konkrétní Delonovské množiny s generátorem jako Pisotovým číslem rovnice $X^3 + X^2 + X - 1 = 0$.

V poslední kapitole je ukázka implementovaného algoritmu na Delonovskou množinu, kterou jsme zkonstruovali v předchozí kapitole. Je zde vykreslení Voroného diagramů v obou metrikách pro naši množinu spolu s jednoduchou analýzou vlastností Voroného diagramů na Delonovských množinách.

Literatura

- [1] Birne Binegar. *Definition and Examples of Rings*. Oklahoma State University, Lecture 14. <<https://math.okstate.edu/people/binegar/3613/3613-114.pdf>>.
- [2] Camille Wormser. *Generalized Voronoi Diagrams and Applications*. PhD thesis, Computer Sciences. Université Nice Sophia Antipolis, 2008.
- [3] David Hodorkovsky. *2-Point Site Voronoi Diagrams*. M.Sc. Thesis, The Technion—Israel Inst. of Technology, Haifa, Israel, 2005. <<http://www.cs.technion.ac.il/~barequet/theses/hodorkovsky-msc-thesis.pdf>>.
- [4] Eric Weisstein. *Voronoi Diagram*. <<https://mathworld.wolfram.com/VoronoiDiagram.html>>. 2021. DOI:10.1137/S0036144599352836.
- [5] Francois Anton, Darka Mioc, Christopher M. Gold. *The Voronoi Diagram of Circles and Its Application to the Visualization of the Growth of Particles*. Transactions on Computational Science. 3. 20-54. DOI:10.1007/978-3-642-00212-0_2.
- [6] Franz Aurenhammer. *Power Diagrams: Properties, algorithms and applications*. SIAM J. Computing, Vol.16, No.1, February 1987.
- [7] Franz Aurenhammer, Rolf Klein. *Chapter 5 – Voronoi Diagrams*. In 'Handbook of Computational Geometry', J.-R. Sack and J. Urrutia (eds.), North-Holland 2000, 201–290. DOI: 10.1016/B978-044482537-7/50006-1.
- [8] Frederick Crabbe. *Voronoi Diagrams*. Teaching texts-Robotics and Machine Vision, United States Naval Academy, 2006. <<https://www.usna.edu/Users/cs/crabbe/SI475/2006/model/voronoi.pdf>>.
- [9] Frido Verweij. *Voronoi Tessellation*. <<https://library.fridoverweij.com/docs/voronoi.html>>.
- [10] J. J. O'Connor, E. F. Robertson. *Georgy Fedoseevich Voronoy*. <<https://mathshistory.st-andrews.ac.uk/Biographies/Voronoy/>>.2007.
- [11] Jakub Vallo. *Detekce asymetrických tunelů v proteinech pomocí Voroného diagramu*. Brno, 2017 [cit. 2021-04-07]. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce RNDr. Adam Jurčík, Ph.D.

- [12] Jana Homrová. *Zobecněné Voroného diagramy*. Plzeň, 2012 [cit. 2021-04-07]. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd. Vedoucí práce RNDr. Světlana Tomiczková, Ph.D.
- [13] Jared Burns. *Centroidal Voronoi Tessellations*. Teaching texts. 2009.
- [14] Jiří Sedláček, *Úvod do teorie grafů*. Academia., 510-21-826, Praha 1977.
- [15] Jiří Veselý. *Metrické prostory*. Přednáška - Funkcionální analýza, kapitola 12, 2015. <<https://www2.karlin.mff.cuni.cz/~jvesely/ma14-15/TUL/AN3E/kapi12.pdf>>.
- [16] Jiří Vyskočil, Marko Genyk-Berezovskyj. *Algorithms, complexity of recursive algorithms*. Presentation CTU FEL, 2010-2014. <https://cw.fel.cvut.cz/old/_media/courses/ae4b33alg/alg02e.pdf>.
- [17] Josef Škrášek, Zdeněk Tichý. *Základy aplikované matematiky*. SNTL – Nakladatelství technické literatury., 80-03-00150-1, Praha 1983.
- [18] Laura Toma. *Voronoi Diagrams*. Computational Geometry-lectures-spring 2017, Bowdoin College, 2017. <<https://www.bowdoin.edu/~ltoma/teaching/cs3250-CompGeom/spring17/Lectures/voronoi-6up.pdf>>.
- [19] Lili Ju, Todd Ringler, Max Gunzbruger. *Voronoi Tessellations and Their Application to Climate and Global Modeling*. United States, 2011. <<https://www.osti.gov/biblio/1090872>>.
- [20] Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars. *Computational Geometry, Algorithms and Applications*. Springer Third Edition. 978-3-540-77973-5. DOI 10.1007/978-3-540-77974-2. e-ISBN 978-3-540-77974-2. 2008.
- [21] Martin Mareš, Tomáš Valla. *Průvodce labirintem algoritmů*. CZ.NIC, z. s. p. o, 978-80-88168-22-5, 1. vydání, Praha, 2017.
- [22] Michael I. Shamos. *Computational Geometry*. A Dissertation, Yale University, The Faculty of the Graduate School, May, 1978.
- [23] Mohammad Farshi. *The Algebraic Computation Tree Model*. Presentation, Yazd University, 2015. <<http://cs.yazd.ac.ir/farshi/Teaching/Spanner-3941/Slides/ADT-Model.pdf>>.
- [24] Nodari Sitchinava. *Divide & Conquer and Analysis of Recurrences*. Teaching notes, University of Hawaii at Manoa, 2017. <<https://www2.hawaii.edu/~nodari/teaching/s17/Notes/Topic-07.html>>.
- [25] Ophir Setter. *Constructing Two-Dimensional Voronoi Diagrams via Divide-and-Conquer of Envelopes in Space*. Thesis Exam-presentation, Tel Aviv University, June 2009. <http://acg.cs.tau.ac.il/projects/in-house-projects/vd-via-dc-of-envelopes/exam_presentation.pdf>.
- [26] P. J. Green, R. Sibson. *Computing Dirichlet Tessellations in the Plane*. The Computer Journal, Volume 21, Issue 2, May 1978, 168–173

- [27] Paul E. Black. *Manhattan distance*. <<https://xlinux.nist.gov/dads/HTML/manhattanDistance.html>>. 2019.
- [28] Petr Beneš. *Voroného diagramy v molekulární chemii*. Brno, 2006 [cit. 2021-04-07]. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce prof. Ing. Jiří Sochor, CSc.
- [29] Petr Felker. *Computational Geometry Introduction*. Presentacion, FEL CTU PRAGUE, 2014. <https://cw.fel.cvut.cz/wiki/_media/misc/projects/oppa_oi_english/courses/ae4m39vg/lectures/01-intro.pdf>.
- [30] Petr Felkel. *Voronoi Diagram*. Presentacion, FEL CTU PRAGUE, 2018. <https://cw.fel.cvut.cz/b181/_media/courses/cg/lectures/06-voronoi-split.pdf>.
- [31] Petr Felkel. *Voronoi Diagram*. Presentacion, FEL CTU PRAGUE, 2017. <https://cw.fel.cvut.cz/b181/_media/courses/cg/lectures/07-voronoi-ii.pdf>.
- [32] Petr Švec. *Plánování trasy robota pomocí Voroného diagramů a dalších prostředků výpočetní geometrie*. <<https://theses.cz/id/jq4os6/>>.
- [33] Quiang Du, EVance Faber, Max Gunzburger. *Centroidal Voronoi Tessellations: Applications and Algorithms*. SIAM Review 41(4), 1999, 637-676.
- [34] Raymond Hill. *Javascript implementation of Steven J. Fortune's algorithm to compute Voronoi diagrams*. <<http://www.raymondhill.net/voronoi/rhill-voronoi.html>>.
- [35] Roberto Tamassia. *Introduction to Voronoi Diagrams*. Teaching texts, Computational geometry, Lecture 13, 1993. <<http://cs.brown.edu/courses/cs252/misc/resources/lectures/pdf/notes09.pdf>>.
- [36] Sarah Greer. *Interactive Voronoi Diagram*. <<http://www.sygreer.com/projects/voronoi/>>. 2017.
- [37] S.Akiyama, H.Rao, W.Steiner. *A certain finiteness property of Pisot number systems*. Journal of Number Theory, Elsevier, 2004, 107, pp.135-160. fhal-00023233f.
- [38] Swami Sarvottamananda. *Voronoi Diagrams*. Teaching texts, Ramakrishna Mission Vivekananda University, 2010. <<http://cs.rkmv.ac.in/~shreesh/media/voronoi.pdf>>.
- [39] Thomas M. Liebling, Lionel Pournin. *Voronoi Diagrams and Delaunay Triangulations: Ubiquitous Siamese Twins*. Optimization Stories, Documenta Mathematica, Extra Volume, 2012, 419-431.
- [40] Tomáš Bayer. *Voronoi diagram. Vlastnosti, použití, konstrukce. Zobecněné Voronoi diagramy..* Prezentace-přednáška, Přírodovědecká fakulta UK, 2016. <<https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adc6.pdf>>.
- [41] Tomáš Hejda, Edita Pelantová. *Spectral properties of cubic complex Pisot units*. Math Comp 85 (2016), 401-421.